ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Acadlib Tracker: Development of a Python-Based Student Gradebook, GPA Tracking, and Library Management System using Object-Oriented Programming

Almiñe, Jonathan B., Balmaceda, Aristeo C., Domanico, Jan Jade C., Limbo, John Kenneth F., Mamaspas, Aldrin Breyl., Angel Jessica Pastor., Loraine San Andres., Meshelle N. Fabro.

Department of Computer Engineering Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) Nagtahan, Manila, Philippines

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500264

Received: 08 June 2026; Accepted: 13 June 2026; Published: 24 June 2026

ABSTRACT

Managing student grades, GPA calculations, and library records in academic institutions is often done manually, leading to errors, inefficiencies, and delayed reporting. This study presents **AcadLib Tracker**, a Python-based system that integrates student gradebook management, GPA computation, and library management into a unified platform using object-oriented programming (OOP).

The system allows teachers and administrators to input and update student grades, automatically compute GPA, monitor library book availability, and track borrowing and return activities. Developed using Python and OOP principles, with SQLite for database management and a graphical interface for ease of use, AcadLib Tracker ensures accurate record-keeping, reduces manual workload, and provides quick access to academic and library data.

Functional testing demonstrated that the system successfully performs its intended operations, including grade entry, GPA calculation, library transaction management, and report generation. Overall, the system provides a reliable and user-friendly solution for academic record management and library monitoring in educational settings.

Keywords: AcadLib Tracker, Student Gradebook, GPA Tracking, Library Management System, Python, Object-Oriented Programming, SQLite

INTRODUCTION

Background of the Study

Academic institutions require efficient methods to manage student performance and library resources. Teachers and administrators often maintain gradebooks manually, record student performance in spreadsheets, and track library books using logbooks or separate systems. These methods can lead to errors in grade calculation, delayed GPA computation, and difficulty in monitoring library activities. In addition, manual systems consume significant time and effort, limiting the ability of staff to focus on instructional and support tasks.

A computerized system can address these challenges by integrating academic and library management into a single platform. Automated gradebooks and GPA calculators reduce human errors, ensure accurate academic tracking, and generate reports efficiently. Similarly, a library management module ensures that book inventory, borrowing, and return activities are monitored in real-time, providing reliable access to library resources.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Problem Statement

This study aims to develop a unified system, **AcadLib Tracker**, that manages student grades, computes GPA, and monitors library records using object-oriented programming. Specifically, the system seeks to answer the following questions:

1. How can student grades and GPA calculations be automated to ensure accuracy and efficiency?
2. How can the system integrate library management functions with student academic records?
3. How can a graphical user interface improve usability for teachers, administrators, and library staff?
4. How can object-oriented programming and database integration improve system maintainability and reliability?

Objectives of the Study

General Objective

The general objective of this study is to develop **AcadLib Tracker**, a Python-based OOP system that combines student gradebook management, GPA tracking, and library management for academic institutions.

Specific Objectives

1. Design a system workflow integrating student grade entry, GPA calculation, and library management.
2. Develop the system using Python and object-oriented programming concepts for modularity and maintainability.
3. Implement a database system using SQLite to store student grades, GPA records, and library transactions.
4. Create a user-friendly graphical interface for easy navigation and interaction.
5. Automate GPA computation and generate academic performance reports.
6. Manage library operations including book availability, borrowing, and returns.
7. Test system functionality to ensure accurate record-keeping and reliable operation.

Significance of the Study

The development of AcadLib Tracker provides multiple benefits:

- **Teachers and Academic Staff:** Can efficiently manage student grades, calculate GPA, and access reports without manual errors.
- **Library Staff:** Can track borrowed and returned books, monitor inventory, and maintain accurate library records.
- **Students:** Benefit from timely updates on grades, GPA, and library availability.
- **Educational Institutions:** Improve record-keeping, reduce administrative workload, and ensure reliable academic and library management.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter discusses the theoretical foundations, related studies, and literature relevant to the development of **AcadLib Tracker**, a Python-based system integrating student gradebooks, GPA tracking, and library management. It includes concepts on student information systems, library management systems, database integration, object-oriented programming, and GUI-based applications.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Student Gradebook and GPA Management

Figure 1. Student Gradebook and GPA Management

Gradebooks are essential tools for recording student academic performance. Traditionally, grades were maintained manually using notebooks or spreadsheets, which may lead to miscalculations, missing records, or delayed reporting. A computerized gradebook automates calculations, tracks cumulative GPA, and allows educators to access academic history easily.

Studies show that digital student information systems improve accuracy, reduce administrative workload, and provide real-time academic insights (Bhavani et al., 2025; Jaiswal & Pandey, 2024). GPA tracking ensures that students and teachers can monitor performance over multiple semesters, identify academic trends, and generate reports efficiently.

Library Management Systems

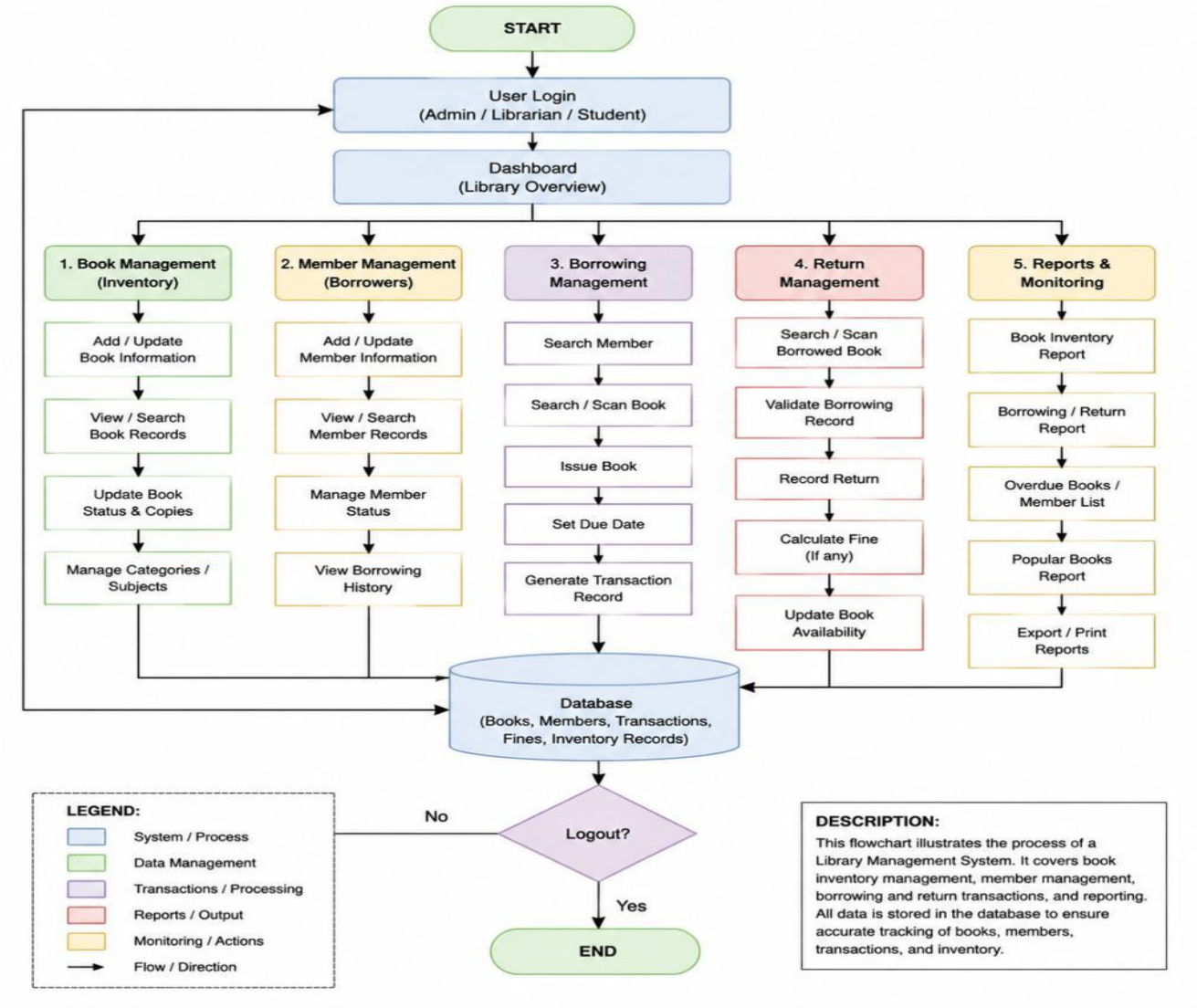

Figure 2. Library Management Systems

Library Management Systems (LMS) provide structured access to book records, borrowing activities, and inventory monitoring. Manual methods in schools often lead to misplaced books, delayed return tracking, and inaccurate inventory reporting. A computerized LMS integrates book records with borrower information and automates borrowing, return, and reporting functions (Dagogo, 2024; Mardiyati, Pauziah, & Sutrisno, 2025).

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Digital LMS solutions improve accessibility for administrators and students, reduce errors, and support reporting requirements. When combined with academic performance systems, LMS modules can provide integrated oversight of both academic and resource usage data.

Object-Oriented Programming in System Development

Figure 3. Object-Oriented Programming in System Development

Object-Oriented Programming (OOP) is a programming paradigm based on encapsulating data and functions into objects. OOP enhances modularity, code reusability, maintainability, and scalability (Silberschatz et al., 2019). In Python, OOP allows developers to create classes representing students, courses, books, or library users, with methods to manipulate grades, compute GPA, or update inventory records.

Previous studies highlight the effectiveness of Python OOP for academic applications. Bhavani et al. (2025) developed a Python-based student information system that demonstrated reliability and easy maintenance through class-based structures. Python OOP also supports integration with databases like SQLite, ensuring data persistence.

Database Management and Integration

Figure 4. **Database Management and Integration**

Database systems are essential for storing academic and library records. SQLite is a lightweight, file-based relational database system suitable for small-scale applications like AcadLib Tracker. It allows storage of student grades, GPA calculations, book inventory, and transaction logs.

Database integration ensures data consistency, prevents duplicate records, and supports queries for reporting and analysis. Studies by Urwin (2025) and GeeksforGeeks (2025) emphasize the importance of proper database design in education management systems to facilitate accurate data retrieval and manipulation.

Graphical User Interface (GUI) Design

Figure 5. **Graphical User Interface (GUI) Design**

A graphical user interface improves usability for non-technical users. GUI-based applications allow users to interact through forms, tables, buttons, and menus instead of command-line instructions. Tkinter or PyQt5 libraries in Python are commonly used to build interactive, desktop-based GUIs for education and library systems (Ariyanti & Supriyanto, 2023).

In educational systems, GUI interfaces increase efficiency, reduce errors, and provide a visually intuitive environment for entering grades, tracking GPA, and managing library resources.

Related Studies

1. **Lathifuddin et al. (2026)** developed a web-based LMS integrating academic record management with digital resource tracking, showing improved record accuracy and reporting efficiency.
2. **Singh (2024)** emphasized the importance of GUI and database integration for usability and accurate student record management.
3. **Nurfadillah & Rahman (2023)** applied usability testing in an OPAC library system, highlighting the need for intuitive interfaces in educational applications.
4. **Rijal (2024)** customized the KOHA library system to improve interface and accessibility, supporting user-centered design principles.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

These studies collectively show that integrating grade management and library systems with proper database design and GUI interfaces can significantly improve accuracy, usability, and efficiency.

Synthesis

Based on the reviewed literature:

- Manual gradebooks and library records are prone to errors and inefficiency.
- Python OOP provides a robust framework for modular and maintainable educational systems.
- SQLite databases allow persistent storage of student and library records.
- GUI interfaces enhance usability for administrators and students.
- Integration of academic and library modules in a single system improves oversight, reporting, and operational efficiency.

Thus, the development of **AcadLib Tracker** follows these principles to provide a unified, reliable, and user-friendly platform for academic performance tracking and library management.

METHODOLOGY

This chapter presents the research design, system development process, database design, tools and technologies, system features, testing procedure, and evaluation methods used in developing **AcadLib Tracker**, a Python-based student gradebook, GPA tracker, and library management system.

Research Design

The study employed a **system development approach**, suitable for designing and implementing a desktop-based application that automates academic and library record-keeping. The system addresses challenges in manual gradebooks, GPA calculation, and library management by providing an integrated platform that reduces errors, improves record accuracy, and streamlines reporting.

The approach includes planning, system design, coding, database integration, testing, and implementation.

System Development Process

Planning

The planning phase involved identifying the key problems in managing student academic records and library resources, such as miscalculations in grades, delayed GPA computation, misplaced library records, and inefficient manual reporting. System requirements were established, including:

- Gradebook management
- GPA computation
- Student record tracking
- Library management (books, borrowing, returns)
- Report generation

System Design

System design included creating flowcharts, interface mockups, and class diagrams to guide development. Key modules were defined:

- **Gradebook Module** – for recording, updating, and calculating student grades and GPA.
- **Student Management Module** – for maintaining student details.
- **Library Module** – for managing books, borrowing, and returns.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

- **Report Module** – for generating academic and library summaries.
- **Exit Module** – for safely closing the system.

Development

The system was developed using **Python** and implemented with **Object-Oriented Programming (OOP)** principles. Python classes were used to represent students, courses, books, and transactions, allowing modular and maintainable code. Methods were implemented for adding, editing, deleting, and retrieving records.

Database Integration

An **SQLite database** was used to store academic records, GPA data, and library information. The database contains the following tables:

Table Name	Description
students	Stores student IDs, names, and course details
grades	Stores student grades for each subject or course
gpa	Stores calculated GPA values for each student
books	Stores book ID, title, author, category, year, and availability
transactions	Records borrow and return details for library books
activity_log	Tracks user actions, such as grade updates or book management

The database ensures data persistence and supports efficient querying for report generation.

Graphical User Interface

A **GUI** was created to improve usability. The GUI allows users to interact with the system through forms, buttons, tables, and menus rather than using command-line inputs. It supports key operations such as:

- Adding and editing grades
- Calculating GPA automatically
- Borrowing and returning books
- Searching student or book records
- Generating academic and library reports

Testing Procedure

Functional testing was performed to verify that each module works as intended. The following aspects were tested:

- Adding, editing, and deleting grades and student records
- Automatic GPA calculation accuracy
- Adding, editing, deleting, borrowing, and returning books
- Searching for students and books
- Generating academic and library reports
- Database integration and data persistence

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

- Input validation to prevent incorrect entries

Each test case was evaluated for expected versus actual results and marked as **PASS** if the system performed correctly.

Implementation

After testing, the system was implemented as a desktop application capable of managing student grades, GPA, and library operations. The interface allows administrators, teachers, or staff to access modules through a secure and intuitive GUI while the SQLite database stores all academic and library data permanently.

Tools and Technologies Used

- **Programming Language:** Python – Handles system logic and operations
- **GUI Framework:** PyQt5 – Creates the interactive user interface
- **Database:** SQLite3 – Stores academic and library records permanently
- **Development Tools:** VS Code – IDE for coding, testing, and debugging
- **UI/UX Design:** Figma – Used to design the interface layout and workflow

System Features

The developed system includes:

1. **Gradebook Management** – Add, edit, and delete student grades
2. **GPA Tracking** – Automatically calculates GPA for each student
3. **Student Management** – Record and maintain student details
4. **Library Management** – Add, edit, delete books; process borrow/return transactions
5. **Report Generation** – Generate summaries of grades, GPA, and library transactions
6. **Activity Logging** – Tracks administrative actions
7. **Exit Module** – Safely closes the system while maintaining database integrity

Summary

The methodology followed a system development approach incorporating Python, OOP, SQLite database, and GUI design. Functional testing confirmed the reliability and accuracy of all modules, ensuring that **AcadLib Tracker** effectively manages student academic and library records, automates GPA calculation, and supports report generation for educational institutions.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of **AcadLib Tracker**, a Python-based system integrating student gradebook management, GPA calculation, and library management. It highlights the system overview, main modules, GUI, testing results, and performance evaluation.

System Overview

AcadLib Tracker is a desktop application designed to streamline academic and library operations. The system automates:

- Recording student grades
- GPA calculation
- Student record management
- Library book tracking

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

- Borrowing and return transactions
- Report generation

It uses **Python** with OOP principles for modularity, **SQLite** for database storage, and **PyQt5** for the GUI. The SQLite database stores student information, grades, GPA, book records, and borrowing history permanently.

The system begins with a secure login. After successful authentication, the user can navigate to modules for grade entry, GPA computation, library operations, and reports.

System Features and Results

Figure 6. Screenshot of the AcadLib Tracker interface showing the Dashboard, Student Management, Gradebook & GPA, Library Management, Borrowing Transactions, and Reports modules.

The GUI demonstrates intuitive navigation, data entry, and real-time display of academic and library information.

AcadLib Tracker software interface showcasing multiple modules: Dashboard, Student Management, Gradebook & GPA, Library Management, Borrowing Transactions, and Reports. The GUI highlights real-time academic and library data management, intuitive navigation, and efficient user interaction

Figure 7. Screenshots of individual modules of the AcadLib Tracker system: (1) Gradebook & GPA Module, (2) Student Management Module, (3) Library Management Module, (4) Borrowing Transactions Module, (5) Reports Module, and (6) Settings Module.
Each module demonstrates intuitive GUI elements such as tables, buttons, and forms, facilitating efficient academic and library management.
AcadLib Tracker interface showcasing six core modules: Gradebook & GPA, Student Management, Library Management, Borrowing Transactions, Reports, and Settings. The GUI demonstrates organized tables, interactive forms, and buttons designed for efficient academic and library data management.

Gradebook and GPA Module

The Gradebook module allows teachers to add, edit, and delete student grades. The GPA module automatically calculates each student's GPA based on the entered grades.

Result:

- Grades were recorded correctly in the database.
- GPA calculation was accurate based on the stored grades.
- Input validation prevented invalid entries such as empty fields or invalid grade values.

Student Management Module

This module manages student personal information, such as name, student ID, and course details.

Result:

- Student records were added, edited, and deleted successfully.
- Search functionality returned accurate student information.
- Database updates reflected correctly in the GUI.

Library Management Module

This module allows tracking of books, borrowing, and return operations.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Result:

- Book records were added, edited, and removed correctly.
- Borrow and return transactions updated book availability status accurately.
- Overdue books and borrowed books were displayed correctly in the interface.

Report Generation

The system generates reports for grades, GPA, and library transactions.

Result:

- Reports displayed accurate records based on database entries.
- Export functionality worked as intended, providing summaries for student performance and library activity.

Graphical User Interface

The GUI, built with PyQt5, included:

- Forms for grade and student entry
- Tables for book inventory and transaction display
- Buttons for module navigation and data operations

Result:

- GUI was intuitive and responsive.
- Users could perform tasks efficiently without prior technical knowledge.

System Testing Results

Functional testing was conducted on all major modules to ensure proper system operation. The test cases included both valid and invalid inputs.

Test Case	What Was Done	Expected Result	Actual Result	Status
Invalid Login Credentials	Entered wrong username or password	Show error message	Error message displayed	PASS
Empty Input Fields	Left required fields blank	Block saving and display warning	Warning displayed	PASS
Add Student Record	Entered valid student info	Record saved in database	Saved successfully	PASS
Edit Student Record	Modified student info	Changes reflected in database	Changes saved successfully	PASS
Delete Student Record	Removed a student	Record deleted from database	Deleted successfully	PASS
Add Book Record	Entered book details	Record saved in database	Saved successfully	PASS

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Edit Book Record	Modified book info	Changes reflected	Changes saved successfully	PASS
Delete Book Record	Removed a book	Record deleted	Deleted successfully	PASS
Borrow Book	Borrowed available book	Status updated, transaction recorded	Status updated, transaction saved	PASS
Borrow Unavailable Book	Attempted to borrow unavailable book	Block transaction	Transaction blocked with warning	PASS
Return Book	Returned borrowed book	Status updated	Status updated successfully	PASS
Search Records	Searched for student or book	Display matching records	Correct records displayed	PASS
Generate Report	Requested GPA or library report	Report generated accurately	Correct report displayed	PASS

All test cases passed successfully, demonstrating that AcadLib Tracker reliably performs its intended functions.

DISCUSSION OF FINDINGS

The results indicate that AcadLib Tracker successfully automates student academic and library operations:

- The **Gradebook and GPA module** reduces manual errors in recording grades and calculating GPAs.
- **Student management** ensures accurate personal records, reducing administrative workload.
- **Library management** supports real-time tracking of books and borrower activity, improving resource allocation.
- **Report generation** allows quick review of academic performance and library activity.
- The **GUI** enhances usability, making the system accessible for staff without programming experience.

Overall, the system improves record-keeping efficiency, minimizes human errors, and provides a unified platform for academic and library management in educational institutions.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of **AcadLib Tracker**, the researchers concluded that the system successfully met its objectives of providing a unified platform for managing student grades, GPA tracking, and library operations.

The system addressed the limitations of manual record-keeping by:

- Allowing accurate recording and modification of student grades.
- Automatically calculating and updating GPA for students based on their course grades.
- Maintaining organized student records, including personal information and academic performance.
- Managing library books, borrowing, and return transactions efficiently.
- Generating academic and library reports to support administrative decision-making.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Functional testing showed that all system modules—gradebook, GPA computation, student management, library management, borrowing/return processing, and report generation—performed correctly. Input validation prevented errors such as empty fields, duplicate IDs, and invalid entries, ensuring the integrity of records.

The graphical user interface provided an intuitive and user-friendly environment, allowing teachers and staff to navigate the system easily without requiring technical expertise. The SQLite database ensured permanent storage of academic and library data, providing reliable access even after the system was closed.

Overall, **AcadLib Tracker** proved to be a reliable, accurate, and efficient solution for managing student academic records and library operations in an educational setting.

Recommendations

Based on the study findings and observed limitations, the following recommendations are proposed for future improvement:

1. **User Authentication and Roles:** Implement role-based access for administrators, teachers, and library staff to enhance system security.
2. **Advanced Reporting:** Add detailed reporting features including GPA trends, borrowing history, overdue books, and performance analytics.
3. **Notification System:** Include notifications or reminders for overdue books or upcoming report submissions.
4. **Multi-User Access:** Enable networked access to allow multiple staff members to operate the system simultaneously.
5. **Cloud Integration:** Add cloud storage or online backup for data security and remote access.
6. **Mobile or Web Interface:** Develop mobile or web-based versions to provide flexible access for teachers and students.
7. **Barcode or QR Code Integration:** Integrate book barcodes or student ID QR codes for faster borrowing and return processing.
8. **Enhanced GUI Design:** Improve interface layout and add visual indicators for borrowed, overdue, or returned books.
9. **Audit Trails:** Include detailed logging of all modifications to grades, GPA, or library records to enhance accountability.
10. **Usability Testing:** Conduct further usability testing with actual users to refine the system's interface and workflow for real-world implementation.

In conclusion, **AcadLib Tracker** currently meets its core objectives but can be further enhanced with additional security, reporting, notification, multi-user support, and modern interface features. Implementing these recommendations will make the system more robust, scalable, and suitable for broader use in educational institutions.

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.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

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. Bhavani, V. T., Chinthana, S., Aishwarya, K., Shweta, B., & Nirmala, S. (2025). Implementation of a student information system using Python. International Journal for Multidisciplinary Research.
3. Birajdar, R. S. (2025). Automated student and library record management: Advancing efficiency in educational institutions. IIP International Multidisciplinary Research Journal.
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. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database system concepts (7th ed.). McGraw-Hill Education.
7. Python Software Foundation. (n.d.). Python documentation. https://docs.python.org/3/
8. Python Software Foundation. (n.d.). Tkinter — Python interface to Tcl/Tk. https://docs.python.org/3/library/tk.html
9. 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
10. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: Strategies for effective human-computer interaction (6th ed.). Pearson.
11. Singh, P. (2024). Library management system. Gurukul Multidisciplinary Research Journal, 407–417. https://doi.org/10.69758/jubi7297
12. SQLite. (n.d.). SQLite documentation. https://www.sqlite.org/docs.html
13. Urwin, M. (2025, April 24). What is SQLite? https://builtin.com/learn/sqlite
14. 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

Jonathan B. Almiñe is a dedicated Computer Engineering student at the Eulogio "Amang" Rodriguez Institute of Science and Technology. He possesses a deep passion for the intersection of hardware integration and software development. Known for his strong work ethic, Jonathan is highly committed to completing his academic responsibilities with excellence. Driven by a commitment to technological advancement, he recently contributed to the research and implementation of a portable, solar-powered charging station designed for student use. Beyond his proficiency in programming, Jonathan is an avid problem solver who is always eager to learn and adapt to the ever-evolving technological landscape.

Aristeo C. Balmaceda is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). He is passionate about technology and dedicated to expanding his knowledge in programming, computer systems, and software development. As a hardworking and determined student, he continuously strives to improve his technical skills and academic performance through perseverance,

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

responsibility, and a strong commitment to learning, Aristeo aims to overcome challenges and achieve his goals in the field of computer engineering. He is eager to explore new technologies and gain valuable experience that will help him become a skilled and successful computer engineer in the future.

Jan Jade C. Domanico is a dedicated Computer Engineering student at Eulogio "Amang" Rodriguez Institute of Science and Technology. Passionate about achieving his dream, he is committed to finishing his studies as a way to give back to the people who are supporting him. While he is currently building foundational knowledge, he is highly motivated to improve his skills in the field of computer engineering. Through focus, determination, and a continuous desire to learn, he aspires to grow into a knowledgeable and successful professional in the technology industry.

John Kenneth F. Limbo is a Computer Engineering student at the Eulogio "Amang" Rodriguez Institute of Science and Technology. Passionate about learning and personal growth, he is eager to gain new knowledge and develop valuable skills. His goal is to successfully complete his studies, graduate, and build a successful career in the future. Through this research, he hopes to contribute meaningful insights while continuing to grow academically and professionally.

Aldrin Breyl A. Mamaspas is a Computer Engineering student at Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) who is passionate about technology and committed to expanding his knowledge in computing and engineering. He continuously works on developing his skills in programming, problem-solving, and system development while maintaining dedication to his academic responsibilities. With a strong desire to learn and adapt to new technologies, he embraces challenges as opportunities for personal and professional growth. Through hard work, perseverance, and continuous self-improvement, he aims to become a competent Computer Engineer capable of contributing innovative solutions and making a meaningful impact in the field of technology.

Angel Jessica L. Pastor is a Computer Engineering student at Eulogio "Amang" Rodriguez Institute of Science and Technology. She is passionate about technology and committed to developing her knowledge and skills in the field of computing and innovation. As a dedicated and responsible student, she continuously seeks opportunities to enhance her understanding of programming, computer systems, and emerging technologies. Through hard work, perseverance, and a strong eagerness to learn, she strives to achieve academic excellence and prepare for future professional challenges. She aims to contribute to the advancement of technology and become a competent and successful professional in the industry.

Loraine San Andres is a diligent student from Eulogio 'Amang' Rodriguez Institute of Science and Technology who aims to develop her skills and knowledge through education and experience. She is recognized for her perseverance, teamwork, and commitment to achieving academic goals.