LIBRARY MANAGEMENT SYSTEM

Project submitted to the
SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology/Master of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

S.Meghana-AP23110011415 S.Ruhi-AP23110011443

P.Devi-AP23110011419 M.Jahnavi-AP23110011445

Under the Guidance of

(Kavitha Rani Karnena)

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

[November, 2024]

CERTIFICATE

Date: 20-Nov-24

This is to certify that the work present in this Project entitled "**LIBRARY MANAGEMENT SYSTEM**" has been carried out by our team under our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in **School of Engineering and Sciences**.

Supervisor

Dr. Kavitha Rani Karnena

Lecturer.

ACKNOWLEDGEMENTS

Our greatest appreciation goes to everybody who contributed to the completion of this Library Management System project.

First and foremost, We would like to thank Dr.Kavitha Rani Karnena for her valuable guidance, constructive comments, and huge support during this project. Her expertise and encouragement really shaped this system.

We thank the institution, SRM University – AP, for providing the necessary resources and platform to conduct this project. We extend our gratitude to the faculty members of the Computer Science and Engineering department for their thoughtful inputs.

We acknowledge also the role of peers and friends for their moral support and technical suggestions, which enriched this work.

Finally, We would like to thank our families for their patience, support, and encouragement; these are what motivated us to complete this project successfully.

This project, implemented in C++, portrays the effort of designing an efficient library management system using Object-Oriented Programming principles; We eagerly look towards the scope of improvements and applications of this system in real-world scenarios.

Table of Contents

ABSTRACT

The Library Management System is an application that provides students with the facility of registration and library book management. It is implemented in C++ using Object-Oriented Programming (OOP) principles to enable automation of key operations of the library like student registration, book management, book issuance, and returns.

The application uses a hash table for efficient storage and retrieval of the records and dynamically allocated arrays for the data of books. The features implemented are:

Register students and prevent duplicate entries.

Add and display books with details such as ID, title, and availability.

Issue books to students while updating inventory and tracking issued books.

Process book returns and update system records in real-time.

This implementation demonstrates modular programming with clear abstraction of functionality, ensuring scalability and maintainability. The system enhances accuracy, reduces manual errors, and offers real-time updates on book availability and student transactions.

The project is meant to be a practically useful management tool for libraries and further provide a basis for future extensions: graphical user interfaces, database integration, or remote access.

Abbreviations

OOP	Object-Oriented Programming
ID	Identifier
Hash Table	Key-Value Storage
Roll ID	Roll Number Identifier
Book ID	Book Identifier
RAM	Random Access Memory
LIB	Library
DB	Database
FIFO	First In, First Out
LIFO	Last In, First Out
OS	Operating System

1.INTORDUCTION

The Library Management System is a software application that helps in simplifying the processes involved in maintaining student record books and inventory within a library. The usual problems that occur in the libraries relate to proper record-keeping of the available books, tracing borrowing and returning events, and recording user details. This system overcomes these challenges by automating the processes, reducing the need for tedious human intervention, and eliminating errors.

This system, implemented using C++, utilized Object-Oriented Programming (OOP) for providing modular and reusable software code with features to

Allow students to register and view account information using a hash table to seek access efficiently.

Store and update book records dynamically while updating the availability of books and the student's borrowing status.

Allow book lending and returns with automatic up-to-date listings of books in store and records of student transactions.

This project demonstrates effective use of data structures like hash tables and dynamic arrays, thereby demonstrating real-world application usage. Further, the system is scalable in nature, which makes it applicable to libraries of different capacities.

By providing the foundation of further enhancements through graphical interfaces, database integration, and remote access, this Library Management System is also structured and user-friendly. It is one step forward in modernizing the operation of libraries and improving the users' experience.

METHODOLOGY

Data Structures and Classes

Hash Table Records of students are stored, and access is quick with no duplication.

Dynamic Array Flexible dynamic allocation for managing book inventory of a library.

Student Class Records roll number, name, and number of books issued to a student.

Book Class Keeps book record, like book id, name, and available copies.

Library Class Controls the operations of student registration, library management, issues, and returns.

Major Functions

Student Enrollment Adds students by roll numbers and doesn't allow any duplicate entry in a hash table.

Library Management Allows books to be added dynamically along with displaying all the inventory details available.

Issue Book Issues books to enrolled students if copies available and adjusts the inventory accordingly.

Book Return Manages the return of books, updates the inventory, and reduces the count of books issued for any student.

Display Features Lists all the books available and issues books for each student separately.

Implementation Details:

Hash Function

Maps roll numbers to hash table indices for efficient retrieval of student data.

Dynamic Memory Allocation

Used for book record management to accommodate the ever increasing number of books.

Modular Approach:

Functionalities spread across classes to ensure reusability along with easy maintenance capabilities.

Scalability and Extendibility

The system's modular design allows additional features, such as:

Graphical User Interface (GUI) improved user interaction.

Database integration for persistent storage of student and book records.

Remote accessibility via web or mobile interface.

DISCUSSION

Student Enrollment

A hash table is used to efficiently and rapidly enrol students and retrieve their information. Given that the hash function is designed to minimize collisions, the access to student records remains fast.

Library Book Administration

A dynamically allocated book inventory allows flexibility in managing the library's inventory of books, and can accommodate additions without a limit.

Issue and Return

The system updates availability and issue counts for books in real-time thus reducing the opportunity for error from human intervention.

Inventory Display

The available books are presented clearly, thus enabling the effective management of a library by administrators.

Efficiency and Scalability

The record keeping of students through hash is done in constant time complexity data retrieval.

Memory allocation of dynamic for books has the effects of making a system scale with the growth of the library, supporting extra entries without restructuring.

Challenges and Limitations

Limited Data Persistence: The current design does not store data in a database; all the records are stored in runtime memory and lost after a program terminates.

Fixed Hash Table Size

The hash table size will be fixed, leading sometimes to inefficiency due to excess space for students beyond that allocated.

User Interface

The system uses a command-line interface. A command-line interface may not be intuitive for users who are not familiar with machines.

Practical Uses

This system can be used in small to medium size libraries to automate some of the regular tasks like book borrowing and returns.

Educational institutions can use this solution for proper management of library records.

FUTURE ENHANCEMENTS

Database

Inclusion of a database will help in creating a persistent record

Graphical User Interface (GUI)

Creation of an even friendlier GUI will further enhance the user interface.

Advanced Features

Addition of features like reservation for books, reminders and transaction details.

CONCLUDING REMARKS

The Library Management System is a robust and efficient solution for the automation of library operations. Using C++ and applying Object-Oriented Programming (OOP) principles, the system effectively integrates functionalities for student registration, book inventory management, and book issuance and return processes.

The use of hash tables and dynamic memory allocation ensures efficient storage and retrieval of data, while the modular design allows for scalability and easy maintenance. The system effectively covers common issues present in library management, including data organization, inventory tracking, and transaction monitoring.

Although the system provides a good deal of structure for managing small to medium-sized libraries, there are some limitations. These include the fact that data is not stored persistently, the size of the hash table is fixed, and the system relies on a command-line interface, which may be inconvenient for non-technical users.

Future improvements include its integration into a database, developing a GUI, and implementation of features like booking reservations and overdue notifications, thereby giving the application a more improved functionality and user experience.

The project gives an implementation of programming concepts into real-world problems and provides a scalable framework for further development and adaptation to larger, more complex systems.

FUTURE WORK

Database Integration:

Store data permanently by using a database instead of in-memory storage.

Allow the system to support massive libraries with voluminous records on students and books.

Graphical User Interface (GUI)

Implement an easy-to-use, user-friendly GUI that replaces the command-line interface.Make the system user-friendly for non-technical users; an easy menu and buttons system

Online Accessibility

Make the system a web-based application that can be accessed from anywhere.

The user should be able to search for a book, check its availability, and request issues/returns online.

Mobile Application

Develop a mobile application that could expand usability and deliver library services from the palm of an individual's hand.

PREMIUM FEATURES

Book Reservation

Reserve books at least before hand.

Overdue Notice

Send reminders about overdue books through email or SMS.

Detailed Transaction Log

That kept track of all library transactions for auditing purposes.

REFERENCES

B. Stroustrup, 2013. The C++ Programming Language.

Addison-Wesley. 4th Edition. Herbert Schildt, 2017.

C++: The Complete Reference. McGraw-Hill Education.

Cormen, Thomas H .,Charles E. Leiserson , Ronald L. Rivest, Clifford Stein, 2009. Introduction to Algorithms. MIT Press. 3rd Edition.

Mark Allen Weiss, 2013. Data Structures and Algorithm Analysis in C++. Pearson.
Scott Meyers, 2005.

Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Addison-Wesley.