**THE 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**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Candidate Name**

**AP23110011520**

**AP23110011495**

**AP23110011500**

**AP23110011504**

**A picture containing text

Description automatically generated**

Under the Guidance of

**LAKSHMI NARAYANA KOLLA**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[ NOVEMBER, 2024 ]**

# Certificate

Date: 13-Nov-24

This is to certify that the work present in this Project entitled “**THE LIBRARY MANAGEMENT SYSTEM**” has been carried out by **Balaji.M, Mehadeesh.P, Koushik.P, Dileep.B** under my/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**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

**Co-supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

# Acknowledgements

I would like to express my heartfelt gratitude to everyone who contributed to the successful completion of this project, *Library Management System*.

First and foremost, I would like to thank my supervisor, **LAKSHMI NARAYANA KOLLA**, for their invaluable guidance, support, and encouragement throughout this project. Their insights and expertise have greatly enriched my learning experience.

I am also grateful to **SRM University – AP** for providing me with the resources and environment needed to work on this project effectively. The knowledge I gained from my courses and faculty members was instrumental in the development of this project.

Furthermore, I would like to extend my thanks to my friends and colleagues who provided encouragement and feedback, which motivated me to improve and refine my work continuously.

Finally, I would like to acknowledge my family for their unwavering support and understanding throughout my academic journey. Their encouragement has been a source of strength for me.

Thank you all for your contributions, which made this project possible.

# Table of Contents

1. Abstract
2. List Of Tables
3. Introduction
4. Methodology
5. Concluding Remarks
6. Future Work
7. Code Link
8. References

# Abstract

The Library Management System is designed to streamline the management of books and records within a library setting, enhancing efficiency and accuracy. This project implements a digital solution for managing book inventory, issuing and returning books, and maintaining student records. Through this system, library staff can easily add new books, register students, issue books to students, track book status, and view overdue books.

Developed in C++, the system uses an object-oriented approach, with classes representing books, students, and core library operations. The application features functions for adding and issuing books, returning books, and listing available and overdue books. The system limits the number of books each student can borrow, preventing misuse and ensuring fair access to resources.

This project addresses common library management challenges, such as record maintenance and tracking, by offering a simple, user-friendly interface and effective backend management. By digitizing these processes, the system provides a reliable and efficient solution for library management, making it suitable for use in small to medium-sized libraries.

# List Of Tables

**Table 1: The Library Management System Functionalities Overview**

| **Feature** | **Description** |
| --- | --- |
| Add Book | Allows library staff to add new books with details like ID, title, and author. |
| Add Student | Registers students with unique IDs and names. |
| Issue Book | Issues a book to a student, sets the due date, and marks it as issued. |
| Return Book | Allows students to return books, updating the status and availability. |
| Check Book Status | Displays the current status of a specific book, showing if it’s issued or available. |
| List Available Books | Shows all books that are currently available in the library. |
| List Overdue Books | Displays books that are overdue based on their due dates. |

**Table 2: Class Descriptions and Attributes**

| **Class Name** | **Attribute** | **Description** |
| --- | --- | --- |
| Library | books | Vector to store all books in the library. |
|  | students | Vector to store all registered students. |
| Book | bookId | Unique identifier for each book. |
|  | title | Title of the book. |
|  | author | Author of the book. |
|  | isIssued | Boolean flag to check if the book is issued. |
| Student | studentId | Unique identifier for each student. |
|  | name | Name of the student. |
|  | booksIssued | Counter for the number of books issued to the student. |
| Date | day, month, year | Stores the issue or return date. |

# Introduction

In educational institutions and public libraries, efficient management of resources is essential to ensure that library services run smoothly and that users have easy access to books and other resources. Traditional library management methods, often involving manual record-keeping, are time-consuming and prone to human error. As a result, many libraries are adopting digital systems to enhance efficiency, reduce errors, and provide seamless services to users.

This project, Library Management System (LMS), is a digital solution designed to streamline the management of library resources. Developed in C++, this system helps librarians and library staff keep accurate records of books and student users, track book availability, and manage book issuance and returns. By automating these processes, the system minimizes errors and provides a simple, user-friendly interface for managing library operations.

**The Library Management System focuses on key functionalities:**

1. **Book Management**: Adding new books with details such as title, author, and ID.
2. **Student Management**: Registering students with unique IDs to enable borrowing.
3. **Issuing and Returning Books**: Tracking which students have borrowed specific books, setting due dates, and updating availability.
4. **Availability and Overdue Tracking**: Displaying available books and identifying overdue books based on their due dates.

By integrating these features, the Library Management System provides a comprehensive tool for maintaining a well-organized library. This report outlines the system’s structure, key functionalities, implementation approach, and potential future enhancements.

# Methodoloy

**System Design**

The Library Management System is built using an object-oriented programming approach in C++. The system’s structure is designed around four primary classes: Library, Book, Student, and Date. Each class plays a specific role in managing the functionalities of the system, as described below.

1. **Library Class**
   * **Purpose**: Manages the overall functionality of the system, including storing collections of books and students, handling book issuance, returns, and displaying information.
   * **Attributes**:
     + vector<Book> books: Stores all books in the library.
     + vector<Student> students: Stores all registered students.
   * **Key Methods**:
     + addBook(): Adds a new book to the library's collection.
     + addStudent(): Registers a new student in the system.
     + issueBook(): Issues a book to a student, updating the book’s status and setting a due date.
     + returnBook(): Processes book returns, marking the book as available and updating the student’s issued book count.
     + displayAvailableBooks(): Lists all books that are currently available for borrowing.
     + checkOverdueBooks(): Displays a list of books that are overdue based on their due dates.
2. **Book Class**
   * **Purpose**: Represents each book in the library with attributes to track its status and borrower information.
   * **Attributes**:
     + int bookId: Unique identifier for each book.
     + string title: Title of the book.
     + string author: Author of the book.
     + bool isIssued: Tracks if the book is currently issued or available.
     + int issuedToStudentId: Stores the ID of the student who borrowed the book (if issued).
     + Date issueDate and Date dueDate: Track the issue date and calculate the due date.
   * **Key Methods**:
     + issue(): Marks the book as issued and sets the issue date and due date.
     + returnBook(): Marks the book as available and clears borrower information.
     + displayInfo(): Displays book details and, if issued, borrower information.
3. **Student Class**
   * **Purpose**: Represents each student registered with the library, managing their ID, name, and the number of books they have issued.
   * **Attributes**:
     + int studentId: Unique identifier for each student.
     + string name: Name of the student.
     + int booksIssued: Number of books currently issued to the student.
   * **Key Methods**:
     + incrementBooksIssued(): Increases the count of books issued to the student.
     + decrementBooksIssued(): Decreases the count when a book is returned.
     + canIssueMoreBooks(): Checks if the student can borrow more books based on the system limit.
4. **Date Class**
   * **Purpose**: Manages date information, specifically for recording issue and due dates for borrowed books.
   * **Attributes**:
     + int day, int month, int year: Store the day, month, and year.
   * **Key Methods**:
     + addDays(int daysToAdd): Calculates a future date based on the current date (used for setting due dates).
     + isPastDue(): Compares the current date with the due date to check if a book is overdue.

# ConcludingRemarks

The Library Management System (LMS) project successfully demonstrates an efficient and organized approach to managing library resources. By digitizing traditional library operations, the LMS reduces the need for manual record-keeping, minimizes errors, and enhances the overall accessibility and convenience of library management.

Through its core functionalities—including adding books and students, issuing and returning books, and tracking overdue items—the system addresses common challenges faced by library staff. The integration of object-oriented principles in C++ allows for a modular, scalable, and easy-to-maintain codebase, while the menu-driven interface makes the system accessible to users of varying technical backgrounds.

The LMS not only meets the objectives of simplified book management and streamlined issuance and return processes but also provides a foundation for future improvements. Potential enhancements could include advanced search capabilities, online catalog integration, and notification features for overdue books, further enhancing the library experience.

In conclusion, this project has been a valuable learning experience, combining software development skills with real-world problem-solving to create a system that could effectively serve small to medium-sized libraries. The Library Management System stands as a practical tool that can be expanded and improved upon in the future to continue meeting the evolving needs of library management.

# Future Work

While the Library Management System (LMS) achieves its core objectives of simplifying book management and tracking issuance and returns, several enhancements could further improve its functionality and usability. Potential future improvements include:

**Advanced Search Functionality**

Adding a search feature that allows users to find books based on criteria such as title, author, genre, or publication year would greatly enhance usability. This feature could use partial text matching to support flexible and user-friendly searching.

**Notification System for Overdue Books**

Implementing a notification system to alert students and library staff about overdue books could improve timely returns. This feature might send email or SMS notifications to students before and after the due date, helping to reduce overdue rates and maintain a well-organized collection.

**Online Catalog Integration**

Integrating the LMS with online book databases (using APIs) would allow the system to automatically retrieve and store book details, making the data entry process faster and more accurate. This feature could be particularly beneficial for libraries with extensive collections.

Each of these enhancements represents an opportunity to make the Library Management System more robust, user-friendly, and adaptable to the evolving needs of modern libraries. Future development efforts can focus on incorporating these features to transform the LMS into a comprehensive tool for library management.

# CODE LINK:

<https://github.com/balaji-mallepalli/Library-Management-System/blob/main/Library_Management_System.cpp>

# References

1. Bjarne Stroustrup: The C++ Programming language (Third Edition). Addison-Wesley.1997.
2. Thinking in C++, Bruce Eckel, Pearson, Second edition, Volume 1, 2002.
3. Object-oriented programming in C++, Robert Lafore, Course Sams Publishing, Fourth edition, 2001.
4. Lischner, Ray. STL Pocket Reference: Containers, Iterators, and Algorithms. O'Reilly Media, Inc., 2003.