**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**

**J.JAGADEESH(AP23110010773)**

**P. Siva Rama Krishna(AP23110010737)**

**N. Praneeth(AP23110010673)**

**J.Teja(AP23110010792)**

**A picture containing text

Description automatically generated**

Under the Guidance of

**MARY CHILAKALAPUDI**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[ NOVEMBER, 2024 ]**

# Certificate

Date: 23-Nov-24

This is to certify that the work present in this Project entitled “**THE LIBRARY MANAGEMENT SYSTEM**” has been carried out by **Jagadeesh.J, Siva rama krishna.P, Praneeth.P, Teja.J** 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. MARY CHILAKALAPUDI

# 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,**MARY CHILAKALAPUDI**, 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 a computerized system meant to make and streamline the process of book and record management in such a way as to ensure efficient and accurate handling. This system provides easy handling for managing the inventory, book issuance, returns, and records of students. They can go ahead to keep the simple records of adding books, student registration, book issuance, tracking status, and picking up overdue ones.

The system, built using C++, is object oriented, where classes are representing books, students, and library operations. The main functionalities included in it are adding and issuing books, processing returns, and generating lists of available and overdue books. Further, the borrowing limits of students are implemented to avoid unbalanced usage and to prevent students from misusing the resources.

This project effectively takes care of the issues relating to record maintenance and tracking, carrying on with a user-friendly interface and substantial backend functionality. By digitizing library processes, it offers a practical and reliable solution suitable for small to medium-sized libraries, ensuring stream-lined operations and improved resource management.

# List Of Tables

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

| **Feature** | **Description** |
| --- | --- |
| Add Book | Enables library staff to add new books, including details such as ID, title, and author. |
| Add Student | Facilitates student registration with unique IDs and names. |
| Issue Book | Issues a book to a student, assigns a due date, and updates its status to issued. |
| Return Book | Allows students to return books, updating their availability and status. |
| Check Book Status | Provides the current status of a book, indicating whether it is available or issued |
| List Available Books | Displays all books currently available in the library for borrowing. |
| List Overdue Books | Shows a list of books that have exceeded their due dates and are overdue for return. |

**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 assigned to each book. |
|  | title | Title of the book. |
|  | author | Author of the book. |
|  | isIssued | Boolean flag indicating whether the book is issued. |
| Student | studentId | Unique identifier assigned to each student. |
|  | name | Name of the student. |
|  | booksIssued | Counter tracking the number of books issued to the student. |
| Date | day, month, year | Attributes to store the day, month, and year of an issue or return date. |

# Introduction

Efficient resource management is essential for educational institutions and public libraries to improve accessibility of books and resources and to operate smoothly. Typically, traditional methods of record-keeping involve manual procedures, which are time-consuming and easily prone to making errors. To cope with these problems, libraries have been increasingly adopting digital systems so as to increase the efficiency, reduce errors, and maintain smooth services.

The Library Management System (LMS) is a digital solution aimed at optimizing library resource management. Built using C++, this system enables librarians and staff to maintain accurate records of books and student users, monitor book availability, and handle book issuance and returns efficiently. By automating these tasks, the LMS reduces errors and provides a straightforward, user-friendly platform for managing library operations effectively.

**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 the mentioned features, the Library Management System offers an all-inclusive support mechanism for establishing a well-structured library. The report shows the system structure, key functionalities, approach towards implementation, and future enhancement.

# Methodoloy

**System Design**

The Library Management System uses an object-oriented programming approach in C++. Hence, the structure of the system is designed around four basic classes: Library, Book, Student, and Date. Each class is responsible for playing a specific role in managing the functionalities of the system, as mentioned 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.

# Concluding

# Remarks

The Library Management System project successfully demonstrates an efficient and well-organized approach in managing library resources. LMS reduces the need for manual record-keeping, minimizes errors, and increases the overall accessibility and convenience of library management with the digitized traditional processes of a library.

The system would generally ease the staff of the library out of some of their common problems through its core functionalities, such as including books and students, issuing and returning books, and tracking overdue items. Object-oriented principles of C++ enable a module-based and scalable codebase, easy to maintain, and the menu-driven interface makes it available for users of all technical background.

It not only meets the objectives of simplified book management and streamlined issuance and return processes but also makes room for future improvements. These potential enhancements might include advance searching capabilities, online catalog integration, and overdue book notice features that will, therefore, augment the library experience.

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

# Future Work

Though the Library Management System (LMS) facilitates its primary objectives of simplification in book management with tracking of issuance and returns, several enhancements would make it more functional and user-friendly. Some of the possible future improvements are:

**Advanced Search Functionality**

Adding search functionality that would enable users to find books by criteria such as title, author, genre, or year of publication would be of immense value for usability. Such functionality may utilize partial text matching in support of flexible and user-friendly searching.

**Notification System for Overdue Books**

Implementation of a system to alert students and the library about the overdue books would increase timely returns. This might be in the form of e-mail or text messages a period before and after the due date, thus reducing the rate of over dues and maintaining a good organized collection.

**Online Catalog Integration**

Integrating the LMS with online book databases (using APIs) would enable the system automatically to retrieve and store book details, thereby accelerating and improving the accuracy of data entry. Such a feature would be especially helpful to libraries with voluminous collections.

Each of these features could be a stepping stone toward an ever-strengthened, accessible, and scalable LMS in line with the shifting needs of libraries of the future. Development of these features would help to transform the LMS into a single, fully all-inclusive tool for the management of libraries.

# CODE LINK:

<https://github.com/SRK1202/LIBRARY-MANAGEMENT.git>

# 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.