A REPORT

ON

LIBRARY MANAGEMENT SYSTEM

(TASK 2)

Submitted By:
Sakshi Mittal

INTERN AT VERVEBRIDGE
POSITION- C++ Developer

LIBRARY MANAGEMENT SYSTEM

INTRODUCTION

This report details the design and functionality of a Library Management System implemented in C++. The system facilitates various library operations, including adding new books, issuing and returning books, searching for books, displaying available books, and data persistence through file handling. The application features a menu-driven interface for ease of use and maintains a record of all books in a vector structure.

SYSTEM DESIGN OVERVIEW

The Library Management System is designed using a modular approach, with each function addressing a specific feature of the system. The program employs a looped menu interface to guide users through available operations.

Key Components:

1. BOOK STRUCTURE
   - Contains the book's title, author, ISBN, and a flag indicating whether the book is currently issued.
   - The book class constructor initializes these properties.

Main Features:

1. Main Menu

- Displays options for adding a new book, issuing a book, returning a book, searching for a book, displaying available books, and exiting the program.

2. Adding Books

- Allows the user to input the title, author, and ISBN of a new book.
- Checks for existing books using the ISBN before adding.

3. Issuing Books

- Facilitates the issuance of books by updating the isIssued flag if the book is available.

4. Returning Books

- Updates the status of a book to indicate it has been returned.

5. Searching for Books

- Offers search options based on ISBN, author, or title, displaying relevant book details if found.

6. Displaying Available Books

- Lists all books in the library along with their status (issued or available).

7. Data Persistence

- Saves and loads library data using a text file, ensuring that book information persists between program executions.

Functional Details:

1. Adding Books

- When adding a new book, the user is prompted to enter the book's title, author, and ISBN.
- The system checks if a book with the same ISBN already exists to prevent duplicates.

2. Issuing Books

- The user enters the ISBN of the book they wish to issue.
- The system verifies if the book is available for issuance and updates its status accordingly.

3. Returning Books

- The user inputs the ISBN of the book they wish to return.
- The system checks for the book’s existence and resets its issued status.

4. Searching for Books

- The user can search for books based on different criteria (ISBN, author, title).
- The system displays relevant information if the book is found.

5. Displaying Books

- The system displays a list of all available books along with their details, including issuance status.

6. Data Persistence

- The library's data is saved to a file (library_data.txt) when the program exits and loaded at the start of the program.

ERROR HANDLING & EDGE CASES

The system handles common user errors and edge cases:

- **Duplicate ISBNs:** When adding a new book, the system checks for duplicates to avoid adding the same book multiple times.

- **Issuing Non-Existent Books:** If the user attempts to issue a book that isn't found, the system provides appropriate feedback.
- **Returning Non-Issued Books:** The system ensures that only books marked as issued can be returned.

CODE ANALYSIS

The system incorporates error handling to address common issues:

- Duplicate ISBNs: When adding a new book, the system checks for duplicates to avoid adding the same book multiple times.
- Issuing Non-Existent Books: If the user attempts to issue a book that isn't found, the system provides appropriate feedback.
- Returning Non-Issued Books: The system ensures that only books marked as issued can be returned.

CONCLUSION

The Library Management System effectively manages a range of library operations with a user-friendly, menu-driven interface. It serves as a valuable tool for managing library resources in small-scale applications. Future enhancements could include advanced search features, user authentication, and additional reporting capabilities.