Digital Object Identifier

Library Management System

JAMIE GROVES , COREA KITTII ,GEORGE OBIANWU ,REMON SHENOUDA , and DOMINIC SWADER.

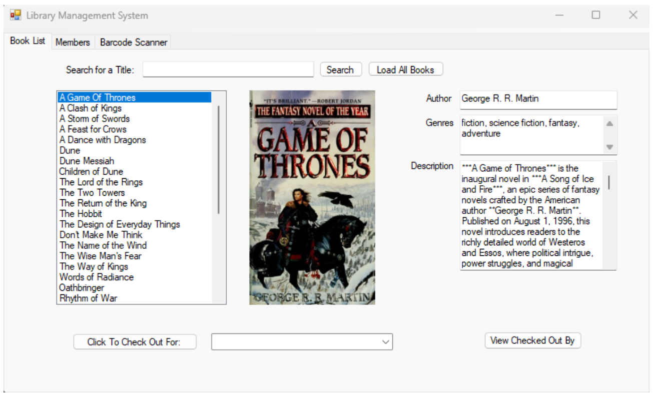
ABSTRACT The library management system is a Windows Forms application designed to streamline the management of library resources, including books, users, and checkout operations. The application was built using C# for the front-end and MySQL for database management on the back-end enabling efficient tracking of data across the application regarding users and book transactions. Some key features include book searching and browsing, user management, overdue book tracking, and bar code scanning integration. This system also connects to the OpenLibrary API to dynamically retrieve book metadata based on scanned ISBNs or titles. This application offers a reliable foundation for managing a small sized library with the potential for future enhancements.

I. INTRODUCTION

DUring the course of the semester, the team has worked to create an appealing Library Management application. The main objective of this application is to simplify the management of books, users, and transactions through a user-friendly and efficient interface. The project was planned, implemented, and finished within the constraints of the course timeline, focusing mainly on getting a demo product ready that could be improved in the future.

II. SYSTEM OVERVIEW

THe application is a Windows Form application that was developed in C# with a MySQL database back-end. The system was designed to manage a library system with ease via an easy-to-use graphical user interface.



A Game Of Thrones
A Clash of Kings
A Storm of Swords
A Feast for Crows
A Dance with Dragons
Dune
Dune Messiah
Children of Dune
The Lord of the Rings
The Two Towers
The Return of the King
The Hobbit
The Design of Everyday Things
Don't Make Me Think
The Name of the Wind
The Wise Man's Fear
The Way of Kings
Words of Radiance
Oathbringer
Rhythm of War

Author: George R. R. Martin

Genres: fiction, science fiction, fantasy, adventure

Description: ***A Game of Thrones*** is the inaugural novel in ***A Song of Ice and Fire***, an epic series of fantasy novels crafted by the American author **George R. R. Martin**. Published on August 1, 1996, this novel introduces readers to the richly detailed world of Westeros and Essos, where political intrigue, power struggles, and magical

The application consists of several features:

Book Management: Displays available books, retrieves metadata from the Open Library API, supports searching by the title, and allows the users to view detailed information including the cover images, descriptions, and genres of each book.

User Management: Allows administrators to view, select, and manage library members, including their checked-out books and contact information.

Check-out and Check-In: Allows users to check out and return books with data recorded in the database. This system also has a built-in overdue books tracker by tracking the checkout dates via the database.

Reporting: Provides the ability to view overdue books and export these overdue books and their checkout dates to a CSV file.

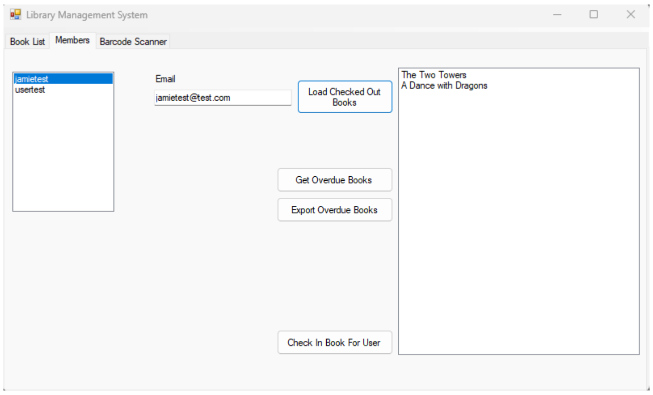
III. SYSTEM DESIGN AND IMPLEMENTATION

THe database was developed using MySQL to provide an easy-to-management database system for the application. The schema consists of three tables: 'books', 'users', and 'checked_out'. The primary keys for each table are: 'id' for the books table, 'user_id' for the users table, and then a transaction 'id' for the 'checked_out' table. The 'checked_out' table has 2 foreign keys which are the 'book_id' and the 'user_id', respectively matching the ids from the other two tables.

The library management system was designed using an approach to ensure the separation of concerns between the user interface, database access, and external API integration. The application features a graphical user interface for interacting with the system. The GUI is designed with the ability to accomplish all the users' needs and requirements. It includes multiple tabs for managing books, users, and barcode scanning.

On the first tab, users can view all the available books in the database. This tab allows the user to navigate the books via a list box of titles. Once the user selects a title from the list box, the application will call the OpenLibrary API and fill in the book's information including the author, genre, cover image, and a description of the book. From this tab, the application also has the capability of checking books out to library users via the drop down bar and the check out button.

On the second tab, the application will display all of the users of the library. From there, it is possible to load all the books checked out by the user and then check books back into the library from the same page. It is also possible to load all of the overdue books, books that have been checked out for over fourteen days, via the load overdue books button. Lastly, on this tab, there is an export overdue books button which will allow the application to export all of the overdue books to a csv and save it to a location that is chosen.



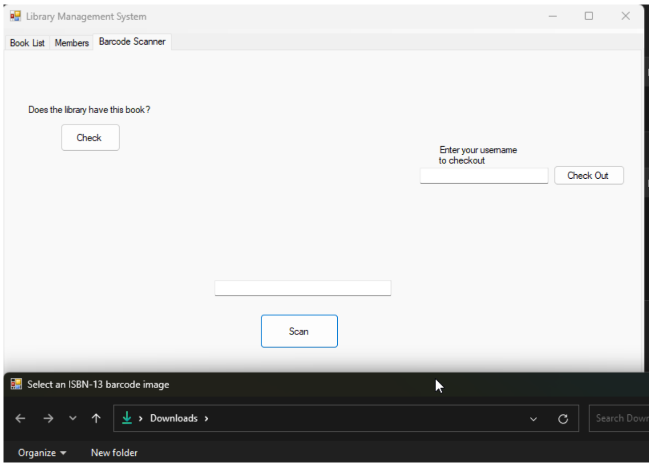
jamietest
usertest

jamietest@test.com

The Two Towers
A Dance with Dragons

On the third tab, the user can scan barcodes from images that are saved locally, or by entering in the book title. This works by searching for a matching ISBN number via the API and pulling the book title information. After a barcode is successfully scanned, the user can check if the book exists in the local library database. If the book does exist, the user can then check out that book.

A barcode scanning algorithm is implemented in the BarcodeScanner class, and utilizes image processing algorithms from the Emgu.CV library, which is a C binding of the OpenCV library with additional features. The scanning process begins with loading the image to be scanned and converting it to a grayscale image, as reducing the available color information makes the process easier. Next is preprocessing the image, which is done by first applying Gaussian blur to it in order to reduce noise. The image is then binarized: the colors of individual pixels are essentially rounded to the closest extreme, resulting in a black-and-white image. Finally, morphological operations are performed in order to clean up the image by removing small artifacts, as well as filling small holes that may be present in the barcode.


Library Management System
Book List
Members
Barcode Scanner
Does the library have this book?
Check
Enter your username to checkout
Check Out
Scan
Select an ISBN-13 barcode image
Downloads
Search Down
Organize
New folder

After preprocessing, the algorithm searches for each bar and space that forms the barcode in the image via contour detection. Of all the contours that are found, any that do not form a rectangle of a minimum height (currently hardcoded at 50 pixels) are filtered out, as those are most likely to be part of the human-readable ISBN number that is not a part of the actual barcode. The valid bars and spaces are then sorted from left to right in preparation for the next step.

Once all valid bars and spaces have been detected and sorted, a binary string representation of the barcode is constructed by iterating through all of them and appending a "1" for each bar, and a "0" for each space. The completed sequence of bits is then decoded and converted into an ISBN-13 barcode number. This process starts by finding the start and end guards (bit pattern "101") and the middle guard (bit pattern "01010"). The bit sequences are then split in two (with the exception of the guards), further split into 6-bit chunks, and then evaluated separately. The algorithm searches a dictionary of predefined bit patterns and matches those present in the barcode in order to find the corresponding number. All decoded digits are then concatenated into a single string and returned as the output, where it can be utilized by the rest of the program.

IV. RESULTS AND DISCUSSION

THe Library Management System was successfully implemented and demonstrated the core functionalities required for managing a small-scale library. All primary features,including book browsing, user selection, book check-in and checkout, overdue tracking, and barcode scanning, were tested during development and confirmed to be working as intended.

Book searching and metadata retrieval via the Open Library API consistently returned accurate information when valid titles or ISBNs were provided. Bar code scanning using EmguCV was tested on a variety of high-resolution bar code images and produced accurate ISBN results in most cases. However, performance was sensitive to image quality; lower contrast or skewed bar codes occasionally failed to decode correctly, highlighting a potential area for future robustness improvements.

User-related operations, such as loading checkout histories, verifying overdue books, and processing check-ins, functioned reliably with real-time updates to the MySQL backend. Exporting overdue records to CSV files was also tested successfully, with generated files preserving the correct data format and encoding.

Overall, the project met its functional requirements and demonstrated that a lightweight yet capable library management system can be built using Windows Forms, C, and MySQL in combination with external libraries and APIs.

V. CONCLUSION

THis project presented the design and implementation of a Library Management System using C, Windows Forms, and MySQL. The system was developed to streamline the management of books, users, and checkout transactions in a small-scale library environment. Key features, such as book browsing, user tracking, overdue monitoring, bar code scanning, and Open Library API integration, were successfully implemented and tested.

The results demonstrated that the application provides a functional and user-friendly platform for basic library operations. The bar code scanning module, in particular, showcased effective integration of EmguCV for real-world image processing. The use of external APIs for book metadata retrieval added value by automating data population and reducing manual entry.

Despite these successes, the system has several areas for future improvement. Including a feature where the application is able to add or delete users would be a primary goal to add to this application. Bar code decoding can be made more robust to accommodate poor image quality, and the addition of user authentication and access control would increase security. Furthermore, implementing features such as automatic due date reminders, advanced analytics, and support for multi-user environments would significantly enhance the application's utility.

Overall, the project achieved its primary objectives and serves as a strong foundation for more advanced library management solutions. It demonstrates how lightweight technologies can be used effectively to deliver scalable tools for administrative tasks in real-world contexts.

• • •