

Library Management System Project

Abderrahim Elallam

December 26, 2023

Project Overview

The Library Management System is a C++ application designed to automate and streamline the management of a library. The system includes functionalities for adding books, managing user accounts, tracking borrowed and returned books, user authentication, and data persistence through file I/O.

Project Structure

Classes

Book Class:

```
1 #include <iostream>
2 #include <string>
3
4 class Book {
5 public:
6     std::string title;
7     std::string author;
8     std::string genre;
9     bool available;
10
11     // Constructor
12     Book(std::string t, std::string a, std::string g) : title(t), author(a), genre
        (g), available(true) {}
13
14     // Display book information
15     void display() {
16         std::cout << "Title: " << title << "\nAuthor: " << author << "\nGenre: "
            << genre;
17         if (available) {
18             std::cout << "\nStatus: Available\n\n";
19         } else {
20             std::cout << "\nStatus: Not Available\n\n";
21         }
22     }
23 };
```

User Class:

```
1 #include <iostream>
2 #include <string>
3
4 class User {
```

```

5 public:
6     std::string username;
7     std::string password;
8
9     // Constructor
10    User(std::string u, std::string p) : username(u), password(p) {}
11
12    // Display user information
13    void display() {
14        std::cout << "Username: " << username << "\nPassword: " << password << "\n\n";
15    }
16
17    // Authenticate user
18    bool authenticate(std::string enteredPassword) {
19        return enteredPassword == password;
20    }
21 };

```

Library Class:

```

1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <fstream>
5 #include <sstream>
6
7 class Library {
8 private:
9     std::vector<Book> books;
10    std::vector<User> users;
11
12 public:
13     // ... Existing code ...
14
15     // Borrow a book
16     void borrowBook(std::string bookTitle, std::string username) {
17         // Implementation details as discussed earlier
18     }
19
20     // Return a book
21     void returnBook(std::string bookTitle, std::string username) {
22         // Implementation details as discussed earlier
23     }
24
25     % Display borrowed books for a user
26     void displayBorrowedBooks(std::string username) {
27         // Implementation details as discussed earlier
28     }
29
30     % ... Other functionalities ...
31
32 private:
33     % Helper functions
34     Book* findBook(std::string title) {
35         // Implementation details as discussed earlier
36     }
37

```

```

38     User* findUser(std::string username) {
39         // Implementation details as discussed earlier
40     }
41
42     % Save library data to a file
43     void saveData() {
44         // Implementation details as discussed earlier
45     }
46
47     % Load library data from a file
48     void loadData() {
49         // Implementation details as discussed earlier
50     }
51 };

```

User Interface and Main Program

```

1  #include <iostream>
2
3  void displayMenu() {
4      % Implementation details as discussed earlier
5  }
6
7  void runLibrarySystem() {
8      % Implementation details as discussed earlier
9  }

```

Example Outputs

Expected outputs for the project include user authentication messages, book availability status, and successful operations such as borrowing and returning books. Below are some example outputs:

```
=== Library Management System ===
```

```
1. Login
```

```
2. Exit
```

```
Enter your choice: 1
```

```
Enter username: user1
```

```
Enter password: password1
```

```
Login successful! Welcome, user1!
```

```
=== Main Menu ===
```

```
1. Display All Books
```

```
2. Borrow a Book
```

```
3. Return a Book
```

```
4. Display Borrowed Books
```

```
5. Logout
```

```
Enter your choice: 1
```

```
=== All Books ===
```

```
Title: Book1
```

```
Author: Author1
```

```
Genre: Fiction
```

Status: Available

Title: Book2

Author: Author2

Genre: Non-Fiction

Status: Not Available

...

Enter your choice: 2

=== Borrow a Book ===

Enter the title of the book you want to borrow: Book2

Borrowing Book2...

Book successfully borrowed! Enjoy your reading.

=== Main Menu ===

1. Display All Books

2. Borrow a Book

3. Return a Book

4. Display Borrowed Books

5. Logout

Enter your choice: 4

=== Borrowed Books ===

Title: Book2

Author: Author2

Genre: Non-Fiction

Status: Not Available

...

Enter your choice: 3

=== Return a Book ===

Enter the title of the book you want to return: Book2

Returning Book2...

Book successfully returned. Thank you!

=== Main Menu ===

1. Display All Books

2. Borrow a Book

3. Return a Book

4. Display Borrowed Books

5. Logout

Enter your choice: 4

=== Borrowed Books ===

No books currently borrowed.

Enter your choice: 5

...