Simple Library Management System

(Java)

K S S Perera

22IT0515

1.0 Introduction

This report explains the implementation of a simple Library Management System using Java. The program enables users to manage a collection of books by adding, borrowing, returning, and viewing available and borrowed books through a console-based menu system.

2.0 features

The Java program offers the following functionalities:

- **Add a New Book:** Users can add books by providing the title, author, and ISBN.
- **Borrow a Book:** Books can be borrowed if available.
- **View Available Books:** Displays a list of available books.
- **View Borrowed Books:** Displays the borrowed books along with the borrowing timestamp.
- **Return a Book:** Users can return borrowed books using their ISBN.
- **Exit:** Quits the program.

3.0 Main structure

C: > Users > HP > Desktop > J LibraryManagementSystem.java

```
21  static ArrayList<Book> bookList = new ArrayList<>();
22
23  public static void main(String[] args) {
24  Scanner scanner = new Scanner(System.in);
25
26  // Adding an initial book to the library
27  bookList.add(new Book("Sherlock Holmes", "Sir Arthur Conan Doyle", "001", "available", null));
28
29  while (true) {
30  System.out.println("\n------------------- Library Management System -------------------");
31  System.out.println("1. Add new book");
32  System.out.println("2. Borrow book");
33  System.out.println("3. View available books");
34  System.out.println("4. View borrowed books");
35  System.out.println("5. Return book");
36  System.out.println("6. Quit the system");
37  System.out.println("-----------------------------------------------------------------");
38
39  System.out.print("Enter your command: ");
40  String command = scanner.nextLine();
41
42  switch (command) {
43  case "1":
44  addBook(scanner);
45  break;
46  case "2":
47  borrowBook(scanner);
48  break;
49  case "3":
50  viewAvailableBooks();
51  break;
52  case "4":
53  viewBorrowedBooks();
54  break;
55  case "5":
56  returnBook(scanner);
57  break;
58  case "6":
59  System.out.println("Leaving the system. Have a great day!");
60  scanner.close();
61  return;
62  default:
63  System.out.println("Invalid command! Please try again.");
64  }
65  }
66  }
```

4.0 Adding Books

```
67
68    public static void addBook(Scanner scanner) {
69    System.out.print("Enter book name: ");
70    String title = scanner.nextLine();
71    System.out.print("Enter author name: ");
72    String author = scanner.nextLine();
73    System.out.print("Enter book ISBN: ");
74    String isbn = scanner.nextLine();
75
76    bookList.add(new Book(title, author, isbn, "available", null));
77    System.out.println("\"" + title + "\" has been added to the library.");
78    }
79
```

```
------------------- Library Management System -------------------
1. Add new book
2. Borrow book
3. View available books
4. View borrowed books
5. Return book
6. Quit the system
-----------------------------------------------------------------
Enter your command: 1
Enter book name: Treasure Island
Enter author name: Robert Louis
Enter book ISBN: B346
"Treasure Island" has been added to the library.
```

4.1 Borrow Books

```
80    public static void borrowBook(Scanner scanner) {
81    System.out.print("Enter book ISBN No: ");
82    String isbn = scanner.nextLine();
83
84    for (Book book : bookList) {
85    if (book.isbn.equals(isbn)) {
86    if (book.status.equals("available")) {
87    book.status = "borrowed";
88    book.time = java.time.LocalDateTime.now().toString();
89    System.out.println("'" + book.title + "' has been successfully borrowed. Enjoy your reading!");
90    return;
91    }
92    System.out.println("The book is already borrowed.");
93    return;
94    }
95    }
96    System.out.println("Book not found.");
97    }
98
```

```
------------------- Library Management System --------------------
1. Add new book
2. Borrow book
3. View available books
4. View borrowed books
5. Return book
6. Quit the system
----------------------------------------------------------------
Enter your command: 2
Enter book ISBN No: B0001
'Sherlock Holmes' has been successfully borrowed. Enjoy your reading!
```

4.2 View available books

```
 98
 99    public static void viewAvailableBooks() {
100    System.out.println("\nAvailable Books:");
101    boolean found = false;
102    for (Book book : bookList) {
103    if (book.status.equals("available")) {
104    found = true;
105    System.out.println("Title - " + book.title);
106    System.out.println("Author - " + book.author);
107    System.out.println("ISBN - " + book.isbn);
108    System.out.println("-------------------------------------------------");
109    }
110    }
111    if (!found) {
112    System.out.println("No books available.");
113    }
114    }
115
```

```
------------------- Library Management System --------------------
1. Add new book
2. Borrow book
3. View available books
4. View borrowed books
5. Return book
6. Quit the system
----------------------------------------------------------------
Enter your command: 3

Available Books:
Title - Sherlock Holmes
Author - Sir Arthur Conan Doyle
ISBN - 001
-------------------------------------------------
Title - Treasure Island
Author - Robert Louis
ISBN - B346
-------------------------------------------------
```

4.3 View borrowed books

```
115
116   public static void viewBorrowedBooks() {
117   System.out.println("\nBorrowed Books:");
118   boolean found = false;
119   for (Book book : bookList) {
120   if (book.status.equals("borrowed")) {
121   found = true;
122   System.out.println("Title - " + book.title);
123   System.out.println("Author - " + book.author);
124   System.out.println("ISBN - " + book.isbn);
125   System.out.println("Borrowed at: " + book.time);
126   System.out.println("--------------------------------------------------");
127   }
128   }
129   if (!found) {
130   System.out.println("No books are currently borrowed.");
131   }
132   }
133
```

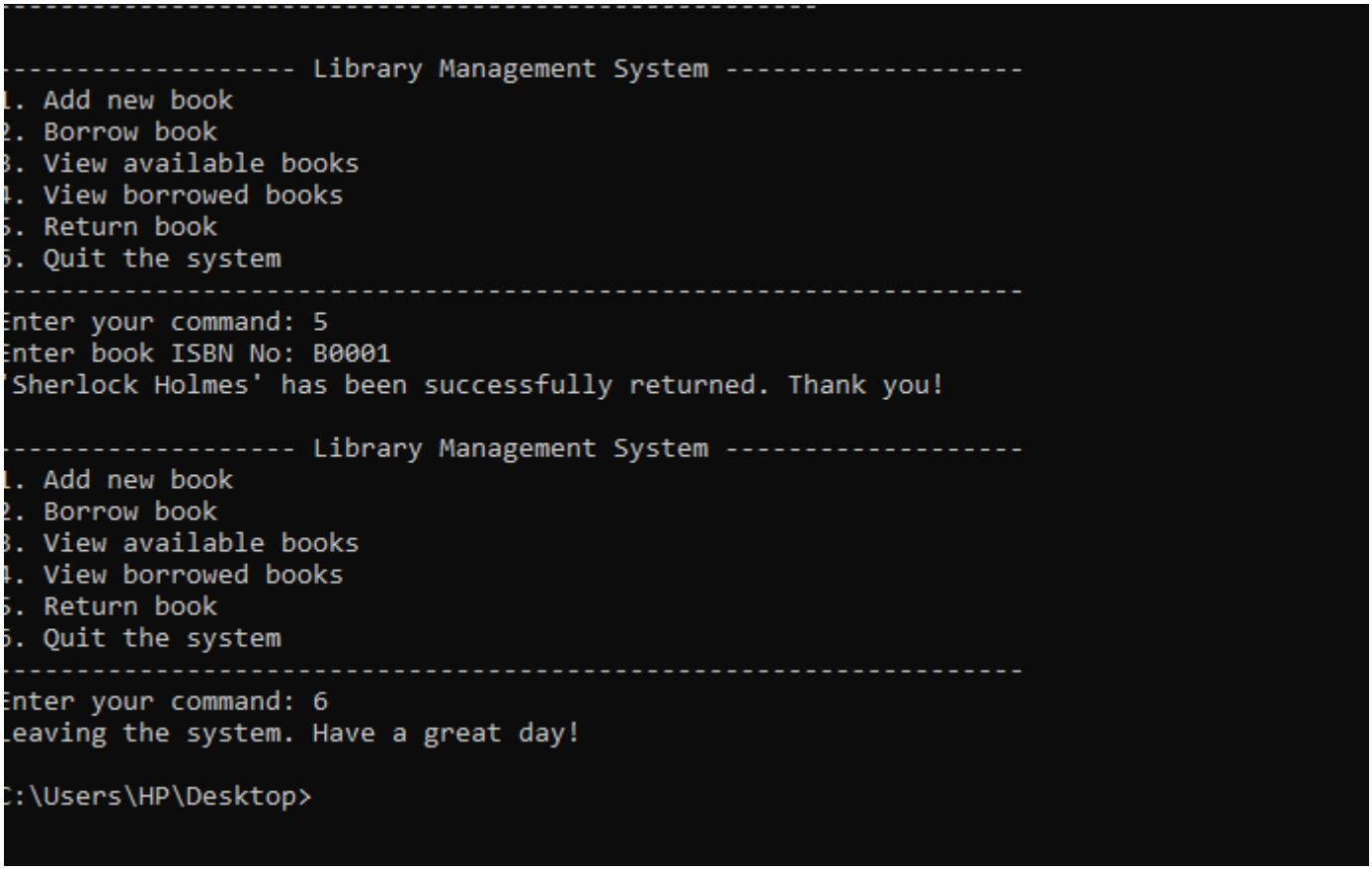
```
------------------- Library Management System -------------------
1. Add new book
2. Borrow book
3. View available books
4. View borrowed books
5. Return book
6. Quit the system
----------------------------------------------------------------
Enter your command: 4

Borrowed Books:
Title - Sherlock Holmes
Author - author conen doyle
ISBN - B0001
Borrowed at: 2025-01-13T02:26:40.733583100
--------------------------------------------------
```

4.4 Return books

```
public static void returnBook(Scanner scanner) {
System.out.print("Enter book ISBN No: ");
String isbn = scanner.nextLine();

for (Book book : bookList) {
if (book.isbn.equals(isbn)) {
if (book.status.equals("borrowed")) {
book.status = "available";
book.time = java.time.LocalDateTime.now().toString();
System.out.println("'" + book.title + "' has been successfully returned. Thank you!");
return;
}
System.out.println("This book is not borrowed.");
return;
}
}
System.out.println("Book not found.");
}
}
```



5.0 conclusion

The Java Library Management System effectively demonstrates how to create a basic console-based application using Java. It covers fundamental programming concepts such as classes, objects, collections, loops, and user input handling. The program provides a functional menu-driven system for managing a small library, allowing users to add, borrow, return, and view books