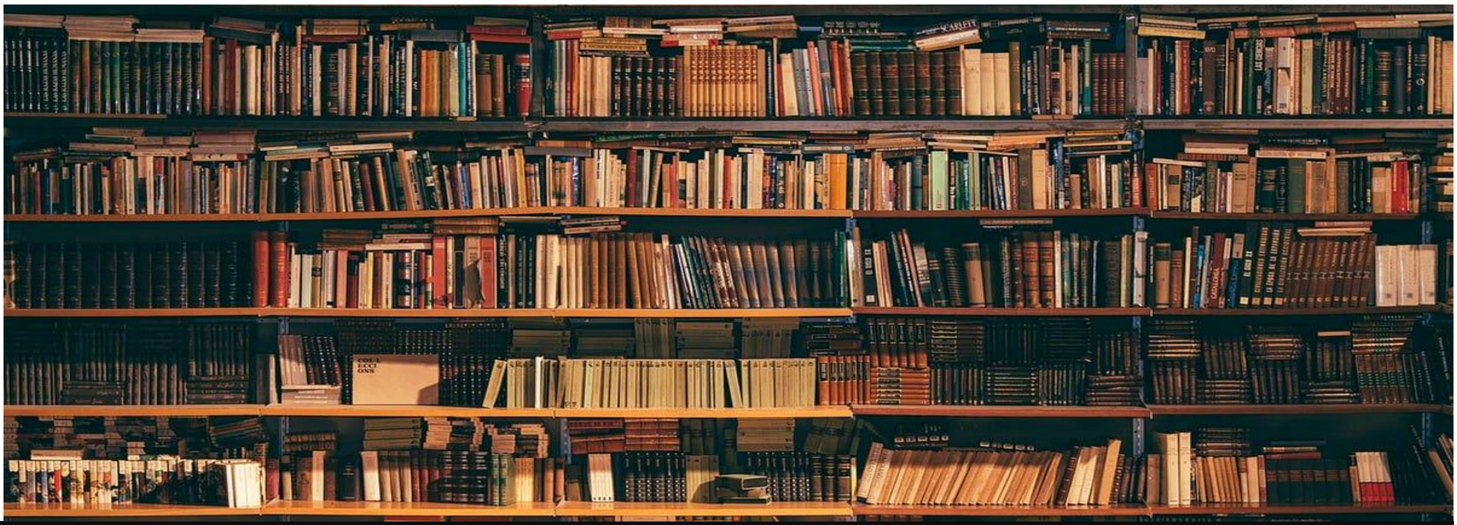

Bookworm Buddy

Library Management System

User Manual

Version: 1.0.0

Prepared by: Malerie Earle
Date: February 28, 2024

Welcome to Bookworm Buddy Library Management System (LMS)!

Welcome to the Bookworm Buddy User Manual! This guide provides an overview of the functionalities and usage guidelines for our software designed to efficiently manage books, authors, and patrons within a library setting. Below is an introduction to the product along with essential information for users.

Introduction

Welcome to Bookworm Buddy, the ultimate Library Management System (LMS) designed to revolutionize library operations and enhance user experience. Our meticulously crafted system empowers librarians to effortlessly add, edit, and delete books, authors, and patrons, while providing patrons with seamless access to borrow and return books. Each book entry is comprehensive, including essential details such as ISBN, title, author, publisher, available copies, and Book Status. Authors are intricately associated with their works and personal information, enriching the library experience for patrons. Patrons can easily search for books by title, author, or ISBN, and smoothly borrow desired copies, with the system seamlessly tracking borrowed books for each patron. With intuitive functionalities tailored for both librarians and patrons, Bookworm Buddy streamlines library operations, enhancing accessibility and convenience for all users. Welcome to the future of library management with Bookworm Buddy, where efficiency meets excellence in serving library needs.

User Documentation

Benefits for Users:

The Library Management System offers several benefits:

- Efficient management of library resources, including books, authors, and patrons.
- Streamlined borrowing and returning processes for patrons.
- Easy access to book information through search functionalities.

Purpose of the Manual:

This manual serves as a guide for users to effectively utilize the *Library Management System*. It provides instructions on installing, configuring, and using the software. Additionally, it includes documentation on the code structure, development process, and deployment procedures.

User Interface

Bookworm Buddy provides a simple and intuitive user interface for seamless interaction. Users can navigate through various menus, starting with the Main Menu options displayed on the screen. From here, there is a choice of Book Menu, Author Menu, Patron Menu, and Library Menu. The system prompts users for input when necessary and provides feedback on the actions performed.

Purpose & Key Features

The Library Management System (LMS) is a comprehensive software solution that allows librarians to manage various aspects of their library, including book inventory, author details, patron information, book borrowing, and returning. The key functionalities include:

Book Management:

- The system allows librarians to add new books, edit existing books, and delete books from the library. Each book has an ISBN, title, author, publisher, number of copies and status.
- It should implement an interface <<***Borrowable>>**, which defines methods to borrow and return books.

Author Management:

- The system allows librarians to add new authors, edit existing authors, and delete authors from the library. They can manage authors including their author ID, name, date of birth, and a list of books they have written.

Patron Management:

- The system allows librarians to add new patrons, edit existing patrons, and delete patrons from the library. Each patron should have a library card number, name, Patron Address, phone number, and a list of books they have borrowed.

- ***Patron Address*** – This class represents the address of a patron in the library management system. It covers the essential components of a patron's address, including the street address, city, province, and postal code. This class provides a convenient way to store and manage address information for patrons within the system.

Book Borrowing:

- Patrons can search for books by title, author, or ISBN, and borrow specific copies of the book. This reduces the number of copies of each book, if there are no copies left, the book’s status changes to borrowed for 14 days. If the book is borrowed for more than 14 days, then a late fee will be charged for every day they are late in returning it. If the book is gone more than 45 days, it is considered lost and will be given an additional 10 dollars along with the regular late fee. A patrons library card number, book ISBN, borrowed date is recorded.

Book Returning:

- Patrons can return books they have borrowed, making them available for other patrons to borrow. The number of copies will increase, any late fees incurred will be charged and payment made.

Development Documentation

Class Diagrams

Book Class
+ ISBN: String + bookTitle: String + bookAuthor: Author author + bookPublisher: String + numCopies: Integer + bookStatus: enum BookStatus() + borrowedDate: Date + returnDate: Date

+ Book()
+ Book(String ISBN, String bookTitle, Author bookAuthor, String bookPublisher, int copies, BookStatus status, Date borrowedDate, Date returnDate)
+ Book(null)
+ printBookInfo(void)

<<interface>> ***Borrowable***
+ borrowBook(Patron): BookStatus status + returnBook(): void

```
+ BookStatus(): BookStatus status
```

Author Class

```
+ authorID: Integer
+ authorName: String
+ dateOfBirth: Date
+ authorBookList: ArrayList<Book>
```

```
+ Author()
+ Author(Integer authorID, String authorName, String
dateOfBirth, ArrayList<Book> authorBookList)
+ Author(null)
+ printAuthorInfo(): void
```

Patron Class

```
+ libraryCard: Integer
+ patronName: String
+ dateOfBirth: String
+ PatronAddress(): PatronAddress patronAddress
+ phNum: String
+ email: String
+ booksBorrowed: ArrayList<Book> booksBorrowed
+ lateFee: double
+ lostFee: double
```

```
+ Patron()
+ Patron(Integer libraryCard, String patronName, String
dateOfBirth, PatronAddress patronAddress, String phNum,
String email, ArrayList<Book> booksBorrowed)
+ Patron(null)
+ getNewLibraryCard(Patron patron): Integer
+ addPatron()
+ deletePatron()
+ editPatron()
```

PatronAddress

- streetAddress: String
- city: String
- prov: String
- postalCode: String

```
+ patronAddress()
+ patronAddress(streetAddress: String, city: String, prov:
String, postalCode: String)
```

<<enumeration>>
BookStatus

```
AVAILABLE,
BORROWED,
OVERDUE,
LOST,
UNAVAILABLE,
```

REMOVED

Library Class
- List<Book> bookList - List<Author> authorList - List<Patron> patronList
+ viewAllBooks(): void + addToBookList(): void + editBook(): void + removeBook(): void + printBookInfo(): void + printBookList(): void + addCopies(): void + removeCopies(): void + searchBookByISBN(): void + searchBookByTitle(): void + searchBookByAuthor(): void + printChosenBookInfo(): void + viewAllAuthors(): void + addNewAuthor(): void + editAuthor(): void + removeAuthor(): void + addBookToAuthor(): void + removeBookFromAuthor(): void + printChosenAuthorInfo(): void + viewAllPatrons(): void + addNewPatron(): void + editPatron(): void + removePatron(): void + addPatronAddress(): void + editPatronAddress(): void + removePatronAddress(): void + printChosenPatronInfo(): void + borrowBook(): void + addToBorrowedBooks(Patron patron, Book book): void + removeFromBorrowedBooks(Patron patron, Book book): void + printLateFee(Patron patron, double lateFee): void + returnBook() : void + payLateFee() : void

Javadocs

Javadocs for the project were generated using the `javadoc` tool included with the JDK. Run the following command in the root directory of the project:

```
javadoc -d docs -sourcepath src -subpackages JavaSprint
```

This generated Javadocs for all classes in the `JavaSprint` package and put them in a folder named `docs`.

Source Code Directory Structure

The source code is organized in a package named `JavaSprint`. The main class, `Menu`, is in this package, along with other classes like `Library`, `Book`, `Author`, `Patron`, and `PatronAddress`. You will also find the enum, `BookStatus` and interface <<Borrowable>>.

```
JavaSprint
├── src
│   └── JavaSprint
│       ├── Library.java
│       ├── Patron.java
│       ├── Book.java
│       ├── Author.java
│       └── PatronAddress.java
├── docs
   └── (documentation files)
```

System Requirements

- Java Runtime Environment (JRE) installed on your system.

- Command-line interface (CLI) or Integrated Development Environment (IDE) to run Java programs.

- The system requires a computer or server with sufficient resources to run the software smoothly.

- Java Development Kit (JDK) version 8 or above.

- *System Compatibility:* The software is compatible with Windows, macOS, and Linux operating systems.

Build Process

To compile the project, navigate to the `src` directory in your terminal and run the following command:

Ex:

```
javac JavaSprint/*.java
```

This will compile all Java files in the `JavaSprint` package.

Compiler Time Dependencies
The project uses the `java.util` package for classes like `List` and `Scanner`. No external libraries or frameworks are used.

Development Standards
The project follows standard Java coding conventions. Class names are in PascalCase, method and variable names are in camelCase, and constants are in UPPER_SNAKE_CASE.

Database Setup
If I was to set up a database for this project, I would set the database up as per the class diagrams above.

Deployment Documentation

Getting Started

To start using Bookworm Buddy, follow these steps:

1. **Download:** Obtain the Bookworm Buddy software package from the designated source.

2. **Installation:** No installation is required as Bookworm Buddy is a standalone Java application.

3. **Compilation:** Compile the Java files using a Java compiler. You can do this by navigating to the directory containing the Java files and running the following command:
   javac Main.java

4. **Execution:** Once compiled successfully, execute the main class to launch the application:
   java Main

5. **Usage:** Follow the on-screen instructions to navigate through the application and utilize its features.

Support

For any assistance or queries regarding Bookworm Buddy, please contact our support team at info@newfienook.com. We are dedicated to providing prompt assistance and resolving any issues you may encounter while using our software.

Conclusion

Bookworm Buddy is your ultimate companion for efficient library management. With its robust features and user-friendly interface, managing books, authors, and patrons has never been easier. We hope you find Bookworm Buddy a valuable asset in your library operations.

Happy Reading!