**DOCUMENTATION**

**1. User Documentation**

**Introduction:**This user documentation provides an overview of the library management system, explains its features and functionality, provides instructions for using the software, addresses common questions and issues, and offers contact information for support.

**Overview:**

The library management application allows users to manage books, authors, and patrons within a library system. It provides functionality for adding, searching, borrowing, and returning books, as well as managing information about authors and library patrons.

**Class Explanation:**

**Author:**

The Author class represents an author in the library system. It contains attributes such as name, date of birth, and a list of books written by the author. Methods are provided for adding/removing books, editing author details, and displaying author information.

**Book:**

The Book class represents a book in the library system. It includes attributes such as title, author, ISBN, publisher, number of copies, and status. Methods are provided for managing book details, borrowing, returning, and displaying book information.

**Borrowable:**

The Borrowable interface defines methods for borrowing and returning books. This interface is implemented by the Book class to enable borrowing and returning functionality.

**Patron:**

The Patron class represents a patron (library member) in the system. It includes attributes such as name, address, phone number, and a list of borrowed books. Methods are provided for managing patron details, borrowing/returning books, and displaying patron information.

**Library:**

The Library class serves as the central component for managing books, authors, and patrons within the library system. It includes methods for adding books, authors, and patrons, listing available books/authors/patrons, searching for books, borrowing/returning books, and managing book copies.

**Status:**

The Status enum defines the status of a book, including AVAILABLE, CHECKED\_OUT, and OVERDUE. This enum is used to track the availability of books in the library.

**Main:**

The Main class serves as the entry point for the application. It contains test cases to demonstrate the functionality of the library management system.

**Features and Functionality:**

**General Features**:

- **Adding** **Books**: Librarians can add new books to the library by providing details such as title, author, ISBN, publisher, and number of copies.

- **Searching Books:**Users can search for books by title, author, or ISBN to find specific items in the library's collection.

- **Borrowing Books:** Patrons can borrow books(more than one copy) from the library, and the system tracks the availability of copies and book status.

- **Returning Books:** Patrons can return borrowed books, and the system updates the status of returned items and adjusts the book inventory accordingly.

- **Managing Authors:** Librarians can list all authors, add information about authors, including their names, dates of birth, lists of books they have written and edit their details as needed.

-**Managing Patrons:** Librarians can create profiles for library patrons, list all patrons, including their names, addresses, phone numbers, and lists of borrowed books.

**Current Features:**

**-** List all library books with the number of available copies

- List all Patrons and their borrowed books

- List All authors and their authored books

- Searching for books by different authors

- Searching for a book by title  
- Searching for a book by ISBN

- Checkout available books to Patrons for borrowing and updates book and patron details

- Tracks borrowed books and updates a book status upon return by a patron

**User** **Interface**:

- The application can be run using a CLI/ Terminal and an IDE (e.g vs code)

**Instructions and Tutorials:**

**Instantiate a new library object**

* Library library = new Library();

**Load data into the library using the loadData method**

* loadData(library);

**Display the library data: books, authors, and patrons**

* library.listBooks();
* library.listAuthors();
* library.listPatrons();

**search for a book by title, author and ISBN**

* library.searchBookByAuthor("J.K. Rowling");
* library.searchBookByISBN("ISBN1");

**search for books to create book objects**

* Book book = library.searchBookByTitle("Harry Potter");
* Book bookTwo = library.searchBookByISBN("ISBN1");

**display book info**

* book.displayBookInfo();

**get the first patron**

* Patron patron = library.patrons.get(0);

**display patron info**

* patron.displayPatronInfo();

**borrow a book by a patron with and without specifying the number of copies**

* library.borrowBook(book.title, patron);
* library.borrowBookPlusCopies(bookThree.title, patron,5);

**return a book by a patron with and without specifying the number of copies**

* library.returnBook("Harry Potter", patron);
* library.returnBookPlusCopies("The Stand", patron,copiesToReturn);

**FAQs and Troubleshooting:**

- Q: How many books can a patron borrow at once?

A: The maximum number of books that a patron can borrow is determined by the number of copies available

Q: Can a patron borrow the same many copies of the same book?

A: Yes so far as there are copies available

- Q: What should I do if I encounter an error while using the system?

A: If you encounter any errors or issues while using the system, please send an email to [valentine.ampah@keyin.com](mailto:valentine.ampah@keyin.com) for support and assistance.

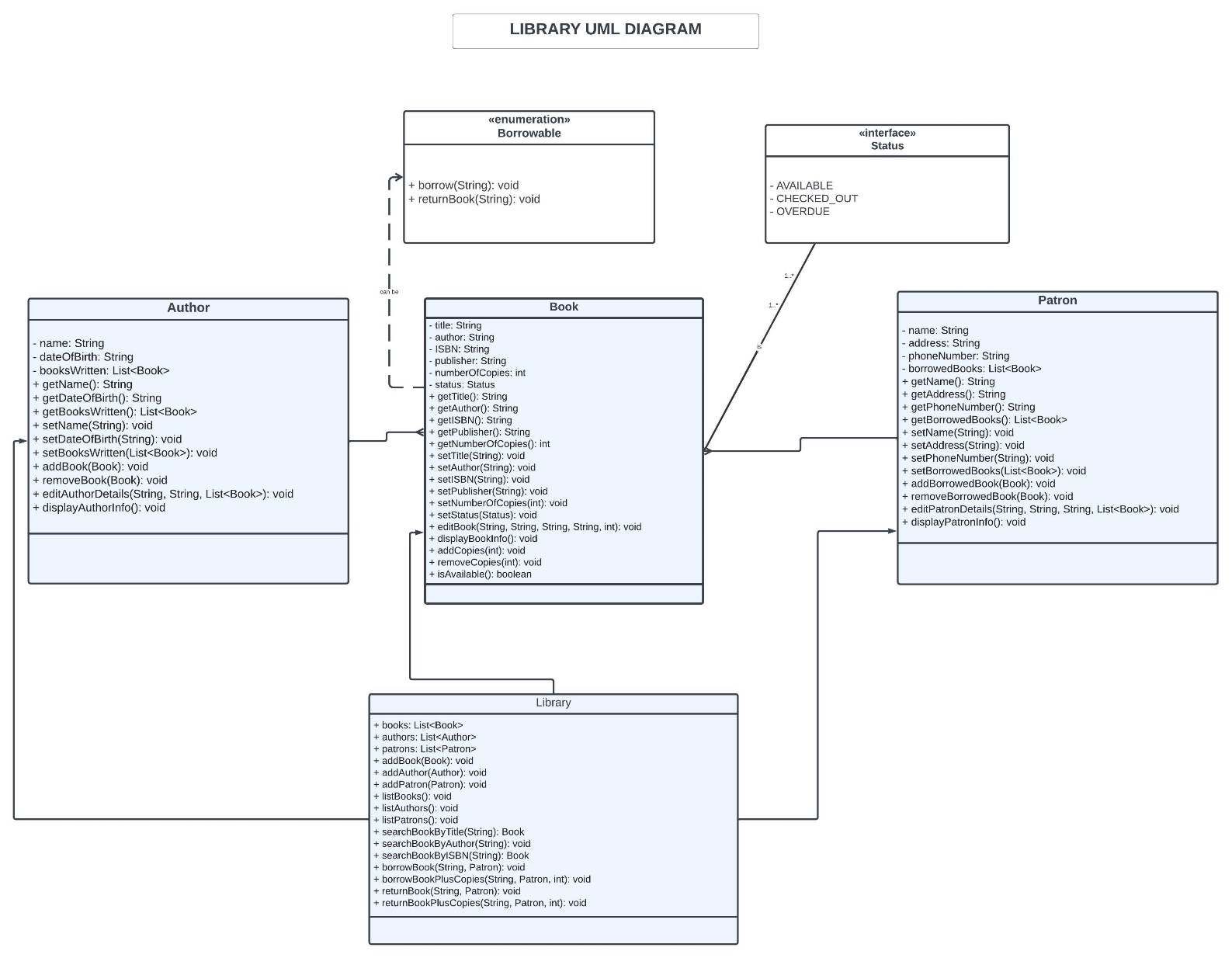
Contact Information and Support:

For further assistance or support, please contact:

- Email: [valentine.ampah@keyin.com](mailto:valentine.ampah@keyin.com) fo

- Phone: 123-567-7908

**Class Diagram:**



**2. Development Documentation**

**Source Code Structure:**

- javasprint/

- Author.java

- Book.java

- Borrowable.java

- Patron.java

- Library.java

- Status.java

- Main.java

**Build Process:**

To compile the project, navigate to the **javasprint**/ directory and run the following command:

* javac javasprint /\*.java

**Compiler Time Dependencies:**

* None

**Database Setup:**

The library management application does not require a database for development.

3. **Deployment** **Documentation**

**Installation Manual:**

1. Unzip the source code files from the zip file.

2. Compile the project using the provided build process.

3. Run the Main class to execute the library management application.

**Configuration Settings:**

No configuration settings are required for deployment.

**Deployment Environment:**

The application can be deployed on any environment with Java Runtime Environment (JRE) installed.

**Troubleshooting Guide:**

If you encounter any issues during deployment, please refer to the following troubleshooting tips:

- Ensure that the Java Development Kit (JDK) is properly installed and configured.

- Check for any compilation errors and resolve them before running the application.

- Verify that the source code repository is accessible.