**Library Management System Documentation**

**1. User Documentation**

1.1 Application Overview

This is a console-based Library Management System (LMS) to help librarians manage library items like books and periodicals, keep track of authors, and handle patrons who borrow items. It has features for adding, updating, deleting, and viewing details for all these areas.

1.2 Explanation of Classes and Their Functionality

* **LibraryItem**: Abstract base class representing any item in the library. Attributes include id, title, author, ISBN, publisher, and numOfCopies. It has subclasses Book and Periodical with specific formats (e.g., printed or electronic).
* **Book**: Extends LibraryItem, represents books with additional formats like audio, electronic, and print.
* **Periodical**: Extends LibraryItem, represents periodicals that can be either printed or electronic.
* **Author**: Represents an author in the system, with attributes for name, dateOfBirth, and a list of library items they've contributed to.
* **Patron**: Abstract base class for individuals using the library. Attributes include name, address, phone, and a list of borrowed items. Two types of patrons include Student and Employee.
* **Student & Employee**: Subclasses of Patron, representing different patron types.
* **Library**: Manages all library items, authors, and patrons. Contains methods for item searches, borrowing, and returning items.
* **Status**: Enum representing the status of library items, with values AVAILABLE, CHECKED\_OUT, and OVERDUE.
* **LibraryDemo**: Entry point for the application. Contains a menu-driven interface allowing users to navigate the system functionalities.

**1.3 Access Instructions**

1. **Setup**: Clone or download the project files from the GitHub repository.
2. **Execution**:
   * Open the project in an IDE (like IntelliJ IDEA or Visual Studio Code).
   * Ensure JDK 8 or higher is installed.
   * Run the LibraryDemo class to start the application.
3. **Usage**:
   * The main menu will provide options to manage library items, authors, patrons, and borrowing/returning items.

Book

Periodical

Author

name, dob

itemList

Student

Employee

LibraryItem

id, title, author

ISBN, publisher

numOfCopies

LibraryMenu

addBook()

deleteItem()

Patron

name, address

phone, borrowed

**Class Diagram**

**2. Development Documentation**

**2.1 Source Code Structure**

* src/LibraryItem: Contains classes and enums related to library items (LibraryItem, Book, Periodical, Status).
* src/Author: Contains the Author class.
* src/Patron: Contains the Patron, Student, and Employee classes.
* src/Library: Contains the Library class with methods for managing items, patrons, and authors.
* src/LibraryDemo: The main entry point of the application with a menu-driven interface.

**Database Structure (Theoretical)**

The following is a suggested database schema for an LMS:

* **Tables**:
  + library\_items: Stores item details with columns for id, title, author, isbn, publisher, num\_of\_copies, type, status.
  + authors: Stores author details with columns for id, name, date\_of\_birth.
  + patrons: Stores patron details with columns for id, name, address, phone, type.
  + borrowed\_items: Stores borrowing records with item\_id, patron\_id, borrow\_date, return\_date.

**GitHub Repository Access**

1. **Cloning the Repository**:
   * URL: https://github.com/username/library-management

**3. Deployment Documentation**

**Installation Instructions**

1. **Install JDK**: Ensure JDK 8 or higher is installed.
2. **Download the Project**:
   * Clone or download the repository from GitHub.
3. **Compile and Run**:
   * Open the project in an IDE, build, and run the LibraryDemo class.
   * Alternatively, navigate to the project root and use the command

**Troubleshooting**

* **Common Issues**:
  + Ensure the Java version is compatible.
  + Confirm directory paths are correctly set if using command line.
* **Database**:
  + The theoretical database setup is optional but provided for potential future expansion into a full-stack application.