Library Management System Development Documentation

Table of Contents

1. Introduction

Welcome to the Development Documentation for the Library Management System. This document provides insights into the development process, including code structure, dependencies, standards, database setup, and source code access.

2. Source Code Directory Structure

The source code for the Library Management System is organized as follows:

- **`src`**: The source code files.
    - **`Library.java`**: Manages books, authors, and patrons.
    - **`Book.java`**: Represents a book.
    - **`Author.java`**: Represents an author.
    - **`Patron.java`**: Represents a patron.

    - Status.java: Enum class for book status.
    - Other utility classes or interfaces, if needed.
- **docs**: Contains documentation files, such as this document.
- **data**: Data files or databases used for testing or storage, if applicable.

This directory structure ensures a clean separation of source code and documentation.

3. Build Process

The project can be compiled and executed using Java. Here's how to compile and run the application:

1. Open a command prompt or terminal.
2. Navigate to the directory containing the source code (the root directory of the project).
3. Compile the Java classes using the `javac` command:

   cssCopy code

   ```
   src
   ```

   This command will compile all `.java` files in the `src` directory.
4. Run the application by executing the main class (e.g., `Demo.java`) with the `java` command:

   Copy code

   ```
   ```

   Replace `Demo` with the name of your main class.

4. Compiler Time Dependencies

The Library Management System does not have external compiler time dependencies. It relies solely on the Java Standard Library.

5. Development Standards

In this project, we follow common Java development standards, which include:

- Meaningful class and method names.
- Proper indentation and formatting.
- Use of meaningful comments to explain code logic.
- Consistent variable naming conventions.

Additionally, we aim to maintain clean and modular code that adheres to Object-Oriented Programming principles.

6. Setting Up a Database for Development

The provided sample does not include a database setup as it uses in-memory data structures. If you wish to integrate a database into the system for development or production, you can do so by using database libraries like JDBC for SQL databases or libraries tailored to NoSQL databases, depending on your requirements. Ensure that you provide the necessary database connection details in your code.

7. Accessing the Source Code Repository

The source code for the Library Management System can be accessed from the project's repository. To clone the repository or obtain the source code, follow these steps:

1. Visit the project's repository URL (e.g., GitHub, GitLab, etc.).
2. Locate the "Clone" or "Download" button and copy the repository's URL.

3. Open a command prompt or terminal.
4. Navigate to the directory where you want to store the project's source code.
5. Use the **`git clone`** command to clone the repository:

   bashCopy code

   ```
   clone
   ```

   Replace **`<repository_url>`** with the actual URL of the repository.

Once the repository is cloned, you'll have access to the latest source code.