**Java Sprint 1 Documentation**

**Alex Ewida**

**Feb.27,2024**

1. User Documentation:

Application Overview:

The Java project is a Library Management System that provides functionality to manage books, authors, and patrons in a library. It includes three main classes: Book, Author, and Patron. Each book has details such as title, author, ISBN, publisher, and number of copies. Authors have their birthdate info and a list of books they have written. Patrons have information such as name, address, and phone number. The Library class organizes and manages the interactions between books, authors, and patrons.

Class Descriptions:

Book: Represents a book with its properties and an interface for borrowable items.

Author: Represents an author with name, birthdate, and a list of written books.

Patron: Represents a library patron with details like name, address, and borrowed books.

Library: Manages books, authors, and patrons. Provides methods to add books, authors, and patrons, and includes helper methods to find authors and book IDs.

How to Start/Access:

To use the application, compile the LibraryManagementSystem.java file using the Java compiler (javac). Run the compiled program using the Java Virtual Machine (java). The library details, including books, authors, and patrons, will be displayed in the terminal.

**Class Diagram:**

**A diagram of a software program

Description automatically generated**

2. Development Documentation:

Javadocs:

Javadoc comments are provided for all classes, methods, and fields in the source code, ensuring comprehensive documentation. These comments describe the purpose and usage of each component.

Source Code Structure:

The source code is organized into a directory structure with separate files for each class (Book.java, Author.java, Patron.java, Library.java, BookID.java, BookCopy.java, Borrowable.java). The Text\_Files folder contains sample text files for loading initial data.

Build Process:

Compile the project using the Java compiler (javac) by executing the following command in the terminal:

Write javac LibraryManagementSystem.java in the terminal.

Run the compiled program with the Java Virtual Machine (java).

Write java LibraryManagementSystem in the terminal.

Dependencies:

The project has no external dependencies beyond the Java Standard Library. It can be compiled and run using the Java Development Kit (JDK).

Development Standards:

Coding standards adhere to Java best practices. Descriptive variable and method names, appropriate use of classes and interfaces, and comprehensive error handling contribute to clean and readable code.

Source Code Repository:

The source code is available on the repository. Clone the repository using the following command: git clone <repository-url>

3. Deployment Documentation:

Installation:

No installation is required. Simply compile and run the LibraryManagementSystem.java file using the Java compiler and Java Virtual Machine.

Steps to Run:

Open a terminal window.

Navigate to the project directory.

Compile the project using javac: javac LibraryManagementSystem.java

Run the compiled program with java: java LibraryManagementSystem

The library details will be displayed in the terminal.

Database Setup:

The application does not utilize a database for development or deployment. All data is managed within the application itself.

For my Java Sprint 1 project, I chose 10 Fantasy novels. I’ve always been a big fan of the fantasy genre. I had to make a library Management system for books, authors and patrons. I found this project very interesting because I enjoy reading books and listening to audiobooks. Building these java files on VS Code was a big challenge, but I feel confident about them.

The LibraryManagementSystem.java file serves as the entry point for a simple library management system. It utilizes the Library class, which encapsulates the core functionality of managing books, authors, and patrons. In the main method, a new instance of the Library is created, and the addBooksAndAuthors method is called to manually populate the library with 10 books, 10 authors, and 10 patrons. The books and authors are added to the library using the addBook and addAuthor methods, respectively, demonstrating the system's capability to handle book and author information.

For the addBooksAndAuthors method, each book, author, and patron is added with specific details such as title, author name, ISBN, publisher, birthdate, address, and phone number. Once the library is populated, the toString method of the Library class is invoked to print the details of the entire library to the terminal window. This file provides a clear demonstration of how the library management system organizes and displays information about books, authors, and patrons in a structured and readable format in the console.