# Midterm Sprint - Java

Library Management System

Victoria Breen, Beth-Ann Penney and Angelia Romanchuk

**October 31, 2024**

## User Documentation:

1. **Application Explanation**

The Library Management system program was designed for use by the librarians in order to manage different library items (books and periodicals) and actions (borrowing and returning of items). The system was built to focus on the efficiency of the storing items while maintaining the ease of the librarians workload. This code is operated through a menu-based interface that allows the user to select which options they preferred while displaying step by step instructions. This reduces the manual work while making the library system more efficient and reliable.

## Application Classes:

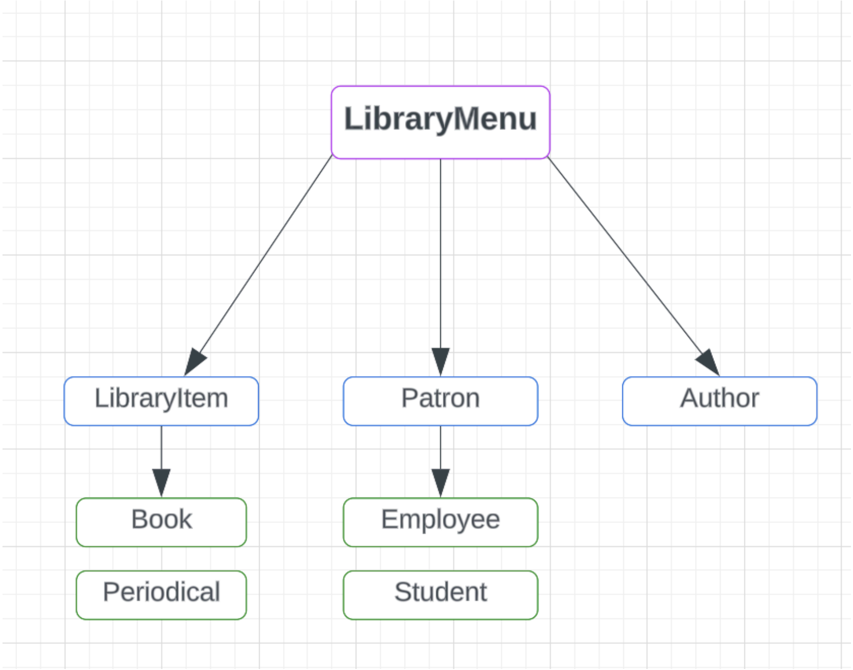
1. **LibraryItem (Abstract Class)**
   - Purpose: This is an abstract class representing a general library item, such as a book or periodical.
   - Key Properties:
     - The LibraryItem class is an abstract class that defines the common properties and behaviors of library items, such as books and periodicals. It includes properties like id, title, author, isbn, publisher, and copiesAvailable, along with methods for borrowing and returning items. Specific item types, such as Book and Periodical, extend this class to implement their unique features.
2. **Patron (Abstract Class)**
   - Purpose: This is an abstract class that represents a library patron (either a student or an employee).
   - Key Properties:
     - The Patron class represents individuals who borrow items from the library, including properties for their name, address, phone number, and a map of borrowed items. It provides methods for borrowing and returning items and displaying borrowed items. The Student and Employee subclasses extend this class to represent different patron types, allowing for specialized behaviors or attributes as needed.

3. **Author**
   - Purpose: Represents an author of library items.
   - Key Properties:
     - The Author class represents an author with properties for their name, date of birth, and a list of items they have written. It provides methods to add written items to the author's list and retrieve this list, allowing the library to track and display the works of each author.
4. **Library**
   - Purpose: This class represents the library itself, holding collections of authors, items (books and periodicals), and patrons.
   - Key Methods:
     - The Library class manages a collection of library items, authors, and patrons. It includes methods for adding and removing items, searching for items by title or ISBN, and facilitating the borrowing and returning process. This class serves as the central hub for managing the library's operations and maintaining relationships between items, authors, and patrons.
5. **LibraryMenu**
   - Purpose: This is the main class that provides a command-line interface for users to interact with the library system.
   - Key Methods:
     - The LibraryMenu class serves as the entry point for the library management system. It provides a user interface through a command-line menu, allowing users to interact with the library's functionalities. This class initializes the library with predefined records, handles user input to add items, authors, and patrons, and facilitates borrowing and returning items. Each option in the menu corresponds to a method that performs specific tasks, ensuring smooth operation and user engagement with the library system.

## How to Start the Application

1. **Environment Setup:**
    - Ensure you have a Java Development Kit (JDK) installed.
    - Use an Integrated Development Environment (IDE) like IntelliJ IDEA, Eclipse, or you can run from the command line.
2. **Running the Application:**
    - If using an IDE, import all the Java files into a project.
    - Locate the LibraryMenu class and run the main method.
    - If using the command line:
        - Compile the Java files using javac *.java.
        - Run the application with java LibraryMenu.
3. **Interacting with the Application:**
    - Follow the prompts in the console to choose options from the menu (add items, borrow/return items, etc.).
    - Provide the required input as requested.

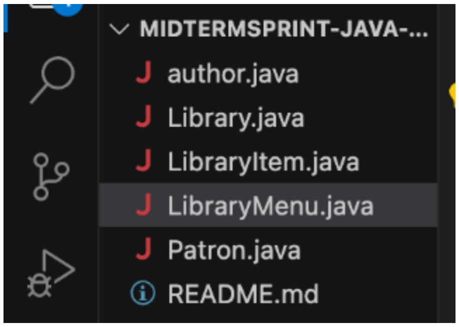
## Class Diagram



## Development Documentation:

## 1. Source Code Description:

The source code for the Library Management Application System is separated into five different files. Library, author, LibraryItems and Patron all contain their own class, where LibraryMenu is what would be the demo or the start of the application for the user to run the library menu system.



MIDTERMSPRINT-JAVA-...
author.java
Library.java
LibraryItem.java
LibraryMenu.java
Patron.java
README.md

## 2. Development Standards:

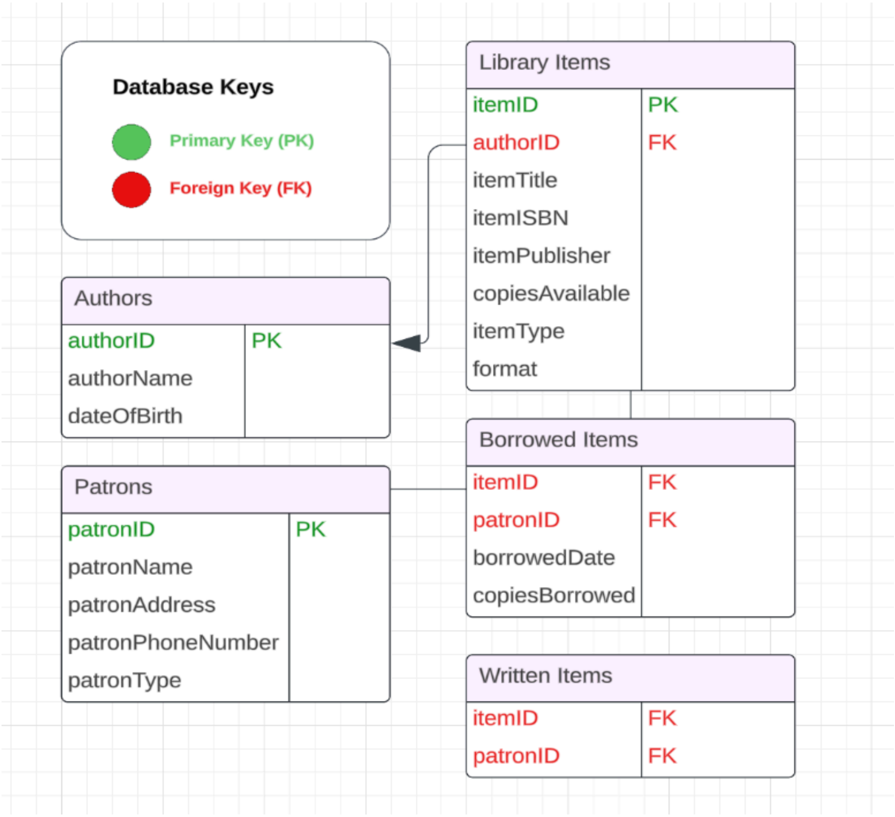
Here are some of the standards we ensured while we were developing this application.

- Made all of the Method names in the project follow camelCase (ex. borrowItem)
- Javadoc comments for most classes and methods
- Version control through Github with correct commit messages

## 3. Database Theory and Relationship:

In theory, it would be much easier to form this Library management system by using a database. The database can handle a larger volume of information about books, authors, and patrons efficiently instead of being stored in arrays. The data would be more effective every time someone uses the program, there would be no using it once and then having to reset the entire system. The database schema could include individual tables for entities like Authors, LibraryItems, Patrons, WrittenItems, and borrowedItems.

The database would be a relational database, it includes Libraryitem to Authors which would be a many-to-one relationship where each library can have one author but the author can have many items. The other would be Patron to LibraryItems where it is a many-to-many relationship through the table of borrowedItems. Here is the entity relationship for the database:



**Database Keys**

🟢 Primary Key (PK)

🔴 Foreign Key (FK)

| Library Items | |
|---|---|
| itemID | PK |
| authorID | FK |
| itemTitle | |
| itemISBN | |
| itemPublisher | |
| copiesAvailable | |
| itemType | |
| format | |

| Authors | |
|---|---|
| authorID | PK |
| authorName | |
| dateOfBirth | |

| Borrowed Items | |
|---|---|
| itemID | FK |
| patronID | FK |
| borrowedDate | |
| copiesBorrowed | |

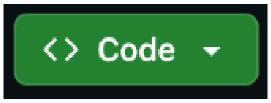
| Patrons | |
|---|---|
| patronID | PK |
| patronName | |
| patronAddress | |
| patronPhoneNumber | |
| patronType | |

| Written Items | |
|---|---|
| itemID | FK |
| patronID | FK |

## 4. Installation Manual

The source code for the application will be located in the repository on github at link https://github.com/victoriaa-b/MidtermSprint-Java-VB-BP-AR

In order to download the code for the application, you will need to select this button.



Once that button is clicked it will give you a few different options such as HTTPS, SSH and GitHub CLI. The one that needs to be accessed is the **HTTPS**, the web URL of that repository is what needs to be copied. After this is done, you will then enter your IDE of choice such as Visual Studio Code or IntelliJ. From there, the terminal will need to be an option and this will be added

**git clone *paste URL here***

This command will clone the repository to your local system and allow you to run the application and make changes to the code.

If you're having issues when it comes to cloning this repository into your IDE, take a look at this webpage for more information.

**GitHub Cloning**

After all of these steps are complete, you can access the code files in your IDE. To run the system, you must go to the LibraryMenu.java file and run it using the arrow button. This will display the system's menu in the terminal and prompt the user to make input options such as insert book or borrowed books.