Library Management System

A **Python-based Library Management System** that allows users to manage books through a menu-driven interface. This system supports essential library operations such as adding books, borrowing, returning, searching, and sorting. It also includes advanced features like Depth First Search (DFS) for searching and sorting algorithms like Insertion Sort and Bubble Sort.

Features

Catalog Management

- Add new books to the library catalog.
- Display all books currently available in the catalog.

Borrowing and Returning Books

- Borrow books by specifying their unique ID.
- Return borrowed books to the library.
- Keep track of recently returned books.

Search Functionality

- Search for books by title or author (case-insensitive).
- Perform an advanced search using **Depth First Search (DFS)**.

Sorting

- Sort books alphabetically by title using:
    - **Insertion Sort**
    - **Bubble Sort**

User-Friendly Menu

- Intuitive console-based interface for managing the library.
- Interactive options for users to perform various operations.

How to Run the Project

1. Clone the repository:
2. git clone https://github.com/your-username/library-management-system.git
3. Navigate to the project directory:

4. cd library-management-system
5. Run the Python script:
6. python library.py
7. Follow the interactive menu to perform library operations.

Menu Options

1. **Add New Book**: Add books by providing their ID, title, and author.
2. **Display Catalog**: View a list of all available books in the catalog.
3. **Borrow Book**: Borrow books by entering their unique ID.
4. **Return Book**: Return borrowed books and update the catalog.
5. **Display Recently Returned Books**: View a list of books recently returned to the library.
6. **Search for a Book**: Find books by title or author.
7. **Sort Catalog (Insertion Sort)**: Sort books alphabetically by title using Insertion Sort.
8. **Sort Catalog (Bubble Sort)**: Sort books alphabetically by title using Bubble Sort.
9. **Search for a Book (DFS)**: Search books using Depth First Search (DFS).
10. **Exit**: Exit the system.

Code Structure

Book Class

Represents a book with attributes:

- book_id: Unique identifier for the book.
- title: Title of the book.
- author: Author of the book.

Library Class

Manages the library operations including:

- Book catalog and borrowed/returned books.
- Sorting and searching functions.
- Menu-driven interface for user interaction.

Technologies Used

- **Language**: Python 3.x
- **Data Structures**: Lists, Recursion
- **Algorithms**: Sorting (Insertion Sort, Bubble Sort), DFS (Depth First Search)

Contributing

Contributions are welcome! To contribute:

1. Fork the repository.
2. Create a new branch:
3. git checkout -b feature-name
4. Commit your changes:
5. git commit -m "Add feature-name"
6. Push to the branch:
7. git push origin feature-name
8. Open a pull request.

License

This project is licensed under the MIT License. See the LICENSE file for details.

Let me know if you'd like any adjustments or specific information added!

Use case diagram: Library Management System