**Overview**

The Library Management System (LMS) is designed to manage a collection of books efficiently. It utilizes a linked list data structure for in-memory management of book records, allowing for dynamic memory allocation and easy manipulation of book entries.

**Key Features**

1. **Book Management**:
   * **Add Book**: Users can add new books by providing the title, author, and ISBN. The system checks for duplicate ISBNs to ensure data integrity.
   * **Delete Book**: Users can delete books from the collection using the ISBN. The linked list is updated accordingly to remove the book node.
   * **Update Book**: Users can modify existing book details, including title, author, and ISBN, ensuring that the linked list reflects these changes.
2. **PDF Management**:
   * **Upload PDF**: Users can associate PDF files with books, allowing for easy access to digital copies of the books.
   * **View PDF**: The system enables users to open and view the associated PDF files directly from the application.
3. **Undo Functionality**:
   * The system maintains an undo stack to allow users to revert the last add or delete operation, enhancing user experience and data management.
4. **Data Persistence**:
   * While the linked list manages books in memory, the system also integrates with an SQLite database for persistent storage. This ensures that book records are saved and can be retrieved even after the application is closed.

**Data Structure**

**Linked List Implementation**

* **Node Structure**: Each book is represented as a node in the linked list, containing the following attributes:
  + **title**: The title of the book.
  + **author**: The author of the book.
  + **isbn**: The unique ISBN of the book.
  + **pdf\_path**: The file path of the associated PDF (if any).
  + **next**: A pointer to the next node in the linked list.
* **Head Pointer**: The linked list maintains a head pointer that points to the first book in the list. If the list is empty, the head pointer is **None**.

**Operations on Linked List**

1. **Adding a Book**:
   * A new book node is created and added to the end of the linked list. If the list is empty, the new node becomes the head.
2. **Deleting a Book**:
   * The system traverses the linked list to find the book with the specified ISBN. Once found, the node is removed, and the pointers are updated to maintain the list's integrity.
3. **Updating a Book**:
   * The system searches for the book by ISBN and updates its attributes directly within the linked list.
4. **Viewing Books**:
   * The system can traverse the linked list to retrieve and display all book records, providing a comprehensive view of the library's collection.

**User Interface**

The LMS features a graphical user interface (GUI) built with Tkinter, allowing users to interact with the system easily. Key components include:

* Buttons for adding, deleting, updating, and viewing books.
* Dialogs for user input, ensuring a smooth workflow.
* A search feature to filter books by ISBN.

**Conclusion**

The Library Management System using a linked list provides an efficient and flexible way to manage a library's book collection. The linked list structure allows for dynamic memory management, making it easy to add, delete, and update book records. Coupled with a user-friendly interface and persistent storage through SQLite, the LMS serves as a robust solution for library management.