**1.1 Synopsis Project:**

The Library Management System project is designed to manage students, books, and their interactions within a library. It encompasses an Array-based list for student management, a Book Management System to handle book checkout and return operations, and a Linked List for student storage. The project provides a menu-driven interface for users to perform various operations such as adding students, searching for students, displaying students, sorting students, deleting students, checking out books, returning books, and displaying relevant information.

**1.2 Objective of the Project:**

The primary objectives of the Library Management System project are:

1. **Student Management:**
   * Add, search, display, sort, and delete students using an Array-based list.
   * Implement a Linked List for student storage.
2. **Book Management:**
   * Manage books through a Book Management System.
   * Perform book checkout and return operations.
   * Display information about checked-out books and return queue.
3. **User Interface:**
   * Provide a user-friendly menu-driven interface for easy interaction.

**2. Use Cases and Brief Description:**

1. **Add Student:**
   * **Purpose:** Allows users to add a new student to the Array-based list.
   * **Usage:** User inputs student details (name, ID, age) through the interface.
2. **Search Student:**
   * **Purpose:** Enables users to search for a student by name.
   * **Usage:** User inputs the name of the student to be searched.
3. **Display Students:**
   * **Purpose:** Displays all students stored in the Array-based list.
   * **Usage:** Provides an overview of the current student list.
4. **Sort Students:**
   * **Purpose:** Utilizes Merge Sort to sort students alphabetically.
   * **Usage:** Enhances the organization of the student list.
5. **Delete Student:**
   * **Purpose:** Allows users to remove a student from the Array-based list.
   * **Usage:** User inputs the name of the student to be deleted.
6. **Checkout Book:**
   * **Purpose:** Facilitates the process of checking out a book.
   * **Usage:** User inputs the title of the book to be checked out.
7. **Return Book:**
   * **Purpose:** Handles the return of a checked-out book.
   * **Usage:** User inputs the title of the book to be returned.
8. **Display Checked Out Books:**
   * **Purpose:** Displays the stack of checked-out books in Last-In-First-Out order.
   * **Usage:** Provides information about books currently checked out.
9. **Display Return Queue:**
   * **Purpose:** Shows the queue of books to be returned in Last-In-First-Out order.
   * **Usage:** Offers insights into the order of book returns.
10. **Display Available Books:**
    * **Purpose:** Displays the list of books available in the library.
    * **Usage:** Provides information about the current availability of books.
11. **Exit:**
    * **Purpose:** Exits the Library Management System.
12. **Display Hardcoded Students and Books:**
    * **Purpose:** Displays hardcoded students, available books, and checked-out books.
    * **Usage:** Provides a snapshot of the initial state of the system.

**3.1 Used Data Structures:**

The project uses the following data structures:

1. **Array-based List:**
   * **Purpose:** Used for storing and managing student information.
   * **Justification:** Allows for dynamic addition, searching, deletion, and sorting of students.
2. **Linked List:**
   * **Purpose:** Used for an alternative implementation of student storage.
   * **Justification:** Supports efficient insertion and deletion of students.
3. **Stack:**
   * **Purpose:** Used for managing the Last-In-First-Out order of checked-out books.
   * **Justification:** Ensures a simple and efficient mechanism for tracking book checkouts.
4. **Queue:**
   * **Purpose:** Used for managing the Last-In-First-Out order of books in the return queue.
   * **Justification:** Ensures a systematic order for book returns.

**4.1 Conclusion:**

In conclusion, the Library Management System project successfully achieves its objectives of managing students and books in a library setting. The implementation of Array-based lists, Linked Lists, Stacks, and Queues demonstrates versatility in handling different aspects of the system. The use of Merge Sort enhances the efficiency of sorting student records. The menu-driven interface provides a user-friendly experience for interacting with the system. The project effectively combines data structures and algorithms to create a functional and organized library management solution.