This is a well-structured plan for a Library Management System. Here are the details of the classes and a roadmap for development.

**Library Management System Class Details**

Here are the proposed classes, their inheritance, and their primary responsibilities and fields.

**1. Base Class: User**

This is an abstract base class that provides common attributes for people interacting with the system.

| Field | Description |
| --- | --- |
| user\_id | Unique identifier (e.g., integer, string). |
| name | Full name of the user. |
| address | Contact address. |
| phone\_number | Contact phone number. |
| email | Contact email address. |

**2. Derived Class: Member (Inherits from User)**

Represents a library member who can borrow books.

| Field | Description |
| --- | --- |
| member\_id | Inherited from user\_id, specific to members. |
| borrowed\_books | A list of Book objects or IDs currently issued to the member. |
| borrowing\_history | A list of past Transaction objects or IDs. |
| total\_fines | Total accumulated fines (e.g., currency value). |

| Method | Description |
| --- | --- |
| display\_details() | Shows the member's profile and current borrowed books. |
| get\_borrowing\_history() | Retrieves and displays the member's past transactions. |

**3. Derived Class: Librarian (Inherits from User)**

Represents a staff member who manages the library operations.

| Field | Description |  |
| --- | --- | --- |
| staff\_id | Inherited from user\_id, specific to librarians. |  |
| employment\_date | Date the librarian started working. |  |

| Method | Description (These methods typically act on the Library class) |
| --- | --- |
| add\_book() | Creates and registers a new book in the system. |
| update\_book\_details() | Modifies the details of an existing book. |
| remove\_book() | Deletes a book from the inventory. |
| register\_member() | Adds a new member to the system. |
| remove\_member() | Deletes a member from the system. |
| issue\_book() | Records a book being borrowed (creates a Transaction). |
| return\_book() | Records a book being returned (updates a Transaction). |
| calculate\_fine() | Determines and records any fines for an overdue return. |

**4. Class: Book**

Represents a single book in the library's inventory.

| Field | Description |
| --- | --- |
| book\_id | Unique identifier for the book. |
| title | The title of the book. |
| author | The author's name. |
| isbn | International Standard Book Number (unique). |
| publication\_year | The year the book was published. |
| total\_copies | The total number of copies the library owns. |
| available\_copies | The number of copies currently not issued. |
| is\_reference\_only | Boolean indicating if the book cannot be borrowed. |

| Method | Description |
| --- | --- |
| display\_details() | Shows the book's title, author, and availability. |
| update\_copy\_count() | Internal method to adjust total\_copies and available\_copies. |

**5. Class: Transaction**

Records the details of a single book borrowing/return event.

| Field | Description |
| --- | --- |
| transaction\_id | Unique identifier for the transaction. |
| book\_id | ID of the book involved in the transaction. |
| member\_id | ID of the member involved. |
| issue\_date | Date the book was borrowed. |
| due\_date | Date the book is expected to be returned. |
| return\_date | Date the book was actually returned (null/None if not yet returned). |
| fine\_amount | Fine charged for this specific transaction. |
| status | E.g., 'Issued', 'Returned', 'Overdue'. |

| Method | Description |
| --- | --- |
| calculate\_fine(return\_date) | Computes the fine based on due\_date and return\_date. |
| complete\_transaction(return\_date, fine\_amount) | Sets the return\_date, fine\_amount, and updates status. |

**6. Class: Library (The Central Hub)**

The main class that encapsulates all the data and provides the interface for management.

| Field | Description |
| --- | --- |
| book\_catalog | A dictionary/list to store all Book objects, keyed by book\_id. |
| member\_list | A dictionary/list to store all Member objects, keyed by member\_id. |
| librarian\_staff | A dictionary/list to store all Librarian objects. |
| transaction\_history | A dictionary/list to store all Transaction objects. |
| fine\_policy | Rules for calculating fines (e.g., amount per day). |

| Method | Description (These act as wrappers/orchestrators for other class methods) |
| --- | --- |
| get\_book(book\_id) | Retrieves a specific book object. |
| get\_member(member\_id) | Retrieves a specific member object. |
| display\_all\_books() | Iterates through book\_catalog and displays details. |
| display\_all\_members() | Iterates through member\_list and displays details. |
| display\_issued\_books() | Finds and displays all transactions with 'Issued' status. |
| process\_issue(librarian, member\_id, book\_id) | Uses the librarian's method to handle the issue, updating book, member, and creating a transaction. |
| process\_return(librarian, transaction\_id) | Uses the librarian's method to handle the return, updating transaction, book, and member. |

**Roadmap to Building the System**

This roadmap breaks down the development process into logical, manageable phases.

**Phase 1: Foundation and Data Structures (The Core)**

| Step | Goal | Key Classes/Concepts | Deliverable |
| --- | --- | --- | --- |
| **1.1** | Define the base User class. | User | A working User class with properties and a basic \_\_init\_\_ (constructor). |
| **1.2** | Implement Member and Librarian. | Member, Librarian | Derived classes inheriting from User with their unique fields (e.g., borrowed\_books list for Member). |
| **1.3** | Implement the Book class. | Book | A complete Book class with all fields and the display\_details method. |
| **1.4** | Implement the Transaction class. | Transaction | A class to record the details of a single borrowing event. |

**Phase 2: Central Management and Persistence**

| Step | Goal | Key Classes/Concepts | Deliverable |
| --- | --- | --- | --- |
| **2.1** | Implement the Library class. | Library | Class initialized with empty dictionaries/lists for book\_catalog, member\_list, etc. |
| **2.2** | Implement basic book management. | Librarian, Library | Implement add\_book, update\_book\_details, and remove\_book functions within the Librarian or as methods of Library called by Librarian. |
| **2.3** | Implement basic member management. | Librarian, Library | Implement register\_member and remove\_member methods. |
| **2.4** | Implement data storage (simple version). | *Data Structure* | Save and load the data (e.g., all Library dictionaries) to a simple file format (like JSON or CSV) to test persistence. |

**Phase 3: Transaction and Fine Logic**

| Step | Goal | Key Classes/Concepts | Deliverable |
| --- | --- | --- | --- |
| **3.1** | Implement the issue\_book process. | Librarian, Library, Book, Member, Transaction | A function that creates a new Transaction, updates the Book's availability, and adds the book to the Member's borrowed\_books list. |
| **3.2** | Implement the fine calculation. | Transaction | Implement the calculate\_fine method in Transaction based on a fixed rule defined in Library.fine\_policy. |
| **3.3** | Implement the return\_book process. | Librarian, Library, Book, Member, Transaction | A function that uses calculate\_fine, completes the Transaction, updates the Book's availability, updates the Member's fines and borrowed\_books list. |

**Phase 4: Reporting and User Interface**

| Step | Goal | Key Classes/Concepts | Deliverable |
| --- | --- | --- | --- |
| **4.1** | Implement display functions. | Library, Member | Implement display\_all\_books, display\_all\_members, and display\_issued\_books in the Library class. Implement get\_borrowing\_history in the Member class. |
| **4.2** | Create a Command-Line Interface (CLI). | *Main Program* | A simple loop (e.g., a while loop) that presents a menu (1. Add Book, 2. Issue Book, 3. Display All Books, etc.) and calls the relevant methods. |
| **4.3** | Final Testing and Refinement. | *All Classes* | Test all possible scenarios (e.g., trying to issue an unavailable book, late return, removing a member with an outstanding fine). |