# Library Management System

***SDD** (Project detailed documentation):-

## Introduction

A Library Management System (LMS) is a software application that simplifies the operations of librarians. It is a complete system for manage member, authors, storing, publishers ,and categories. The primary objective of an LMS is to properly organize and manage the resources available in a library, making it easier for librarians to conduct everyday operations and create a user-friendly experience for users.

The objective of the Library Management System (LMS) project is to design and implement an efficient and user-friendly system that automates the various tasks associated with managing a library.

### The primary goals of the project include:

1. Effective book management: Simplify the process of managing and tracking books to ensure an organized and easily accessible collection.
2. User-Friendly Interface: Developing an intuitive and user-friendly interface for library staff to facilitate easy navigation, quick retrieval of information, and seamless interaction with the system.
3. Automation of Processes: Automate routine library tasks such as login, book management, and member management to improve operational efficiency and reduce manual workload.
4. Inventory Management: Implementing a robust inventory management system to monitor stock levels, identify popular titles, and facilitate timely reordering of books to maintain a well-stocked library.
5. Integration with Other Systems: Offering the flexibility for integration with other academic or administrative systems to create a cohesive and interconnected information ecosystem within the institution.
6. Scalability: Designing the system to be scalable, allowing for easy expansion and adaptation to the evolving needs of the library as it grows over time.

By achieving these goals, the library management system aims to enhance overall efficiency and accessibility and assist those responsible for library services, ultimately contributing to the enrichment of the learning and research environment within the institution.

### Functional Scope:

1. Book management: The system must allow the librarian to perform tasks related to books such as adding, displaying, updating and deleting books. Each book contains information such as title, author, category, publisher, and ISBN.
2. Member Management: The system must allow the librarian to perform tasks related member such as adding, displaying, updating and deleting a member. Each member contains information such as full name, mobile number and identification number
3. Author Management: The system must allow the librarian to perform tasks related to authors including adding, viewing, updating, and deleting author details.
4. Category Management: The system must allow the librarian to perform tasks related to book categories by performing operations such as adding, viewing, updating, and deleting categories.
5. Publisher Management: The system must allow the librarian to perform tasks related to publishers by providing options to add, view, update, and publisher information.

### Non-Functional Scope:

1. Usability: Ensuring a user-friendly interface that promotes ease of navigation and a positive user experience for library staff .
2. Scalability: Designing the system to accommodate growth in the library's collection and member base over time.
3. Performance: Meeting performance standards to ensure timely response and efficient processing of library transactions.
4. Reliability: Building a reliable system that minimizes downtime and ensures the continuous availability of library services.

5. Security: Incorporating robust security measures to protect against unauthorized access, data breaches, and other security threats.

**Summary:** This project is a Library Management System implemented using Spring Boot. It provides CRUD (Create, Read, Update, Delete) operations for managing books, authors, categories, and publishers.

### Methodologies | Project Synopsys for Library Management System:

In LMS we are using various technologies and new methodologies to solve our problems. Below are the detailed description about the technology used and methods we are applying in our project.

Technology Used:

- Spring Boot: A powerful framework for building Java applications quickly and with minimal configuration.
- MySQL: An open-source relational database management system used for storing application data.
- Maven: A build automation tool used primarily for Java projects to manage dependencies and build processes.

# Library Management System

***SRS**(Software Requirements Specification):-

UseCase Diagram:

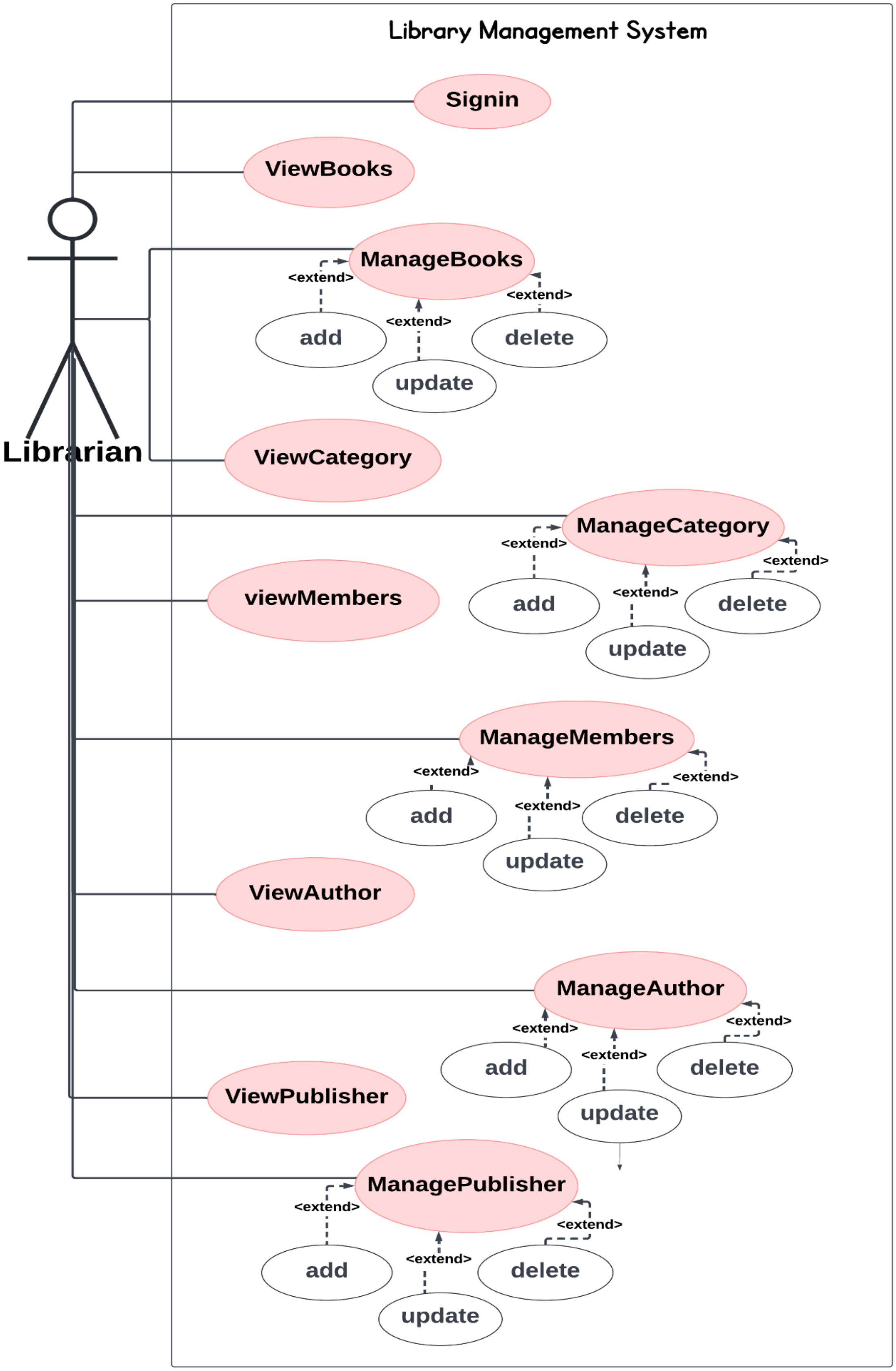
Actors

**librarian**

UseCases

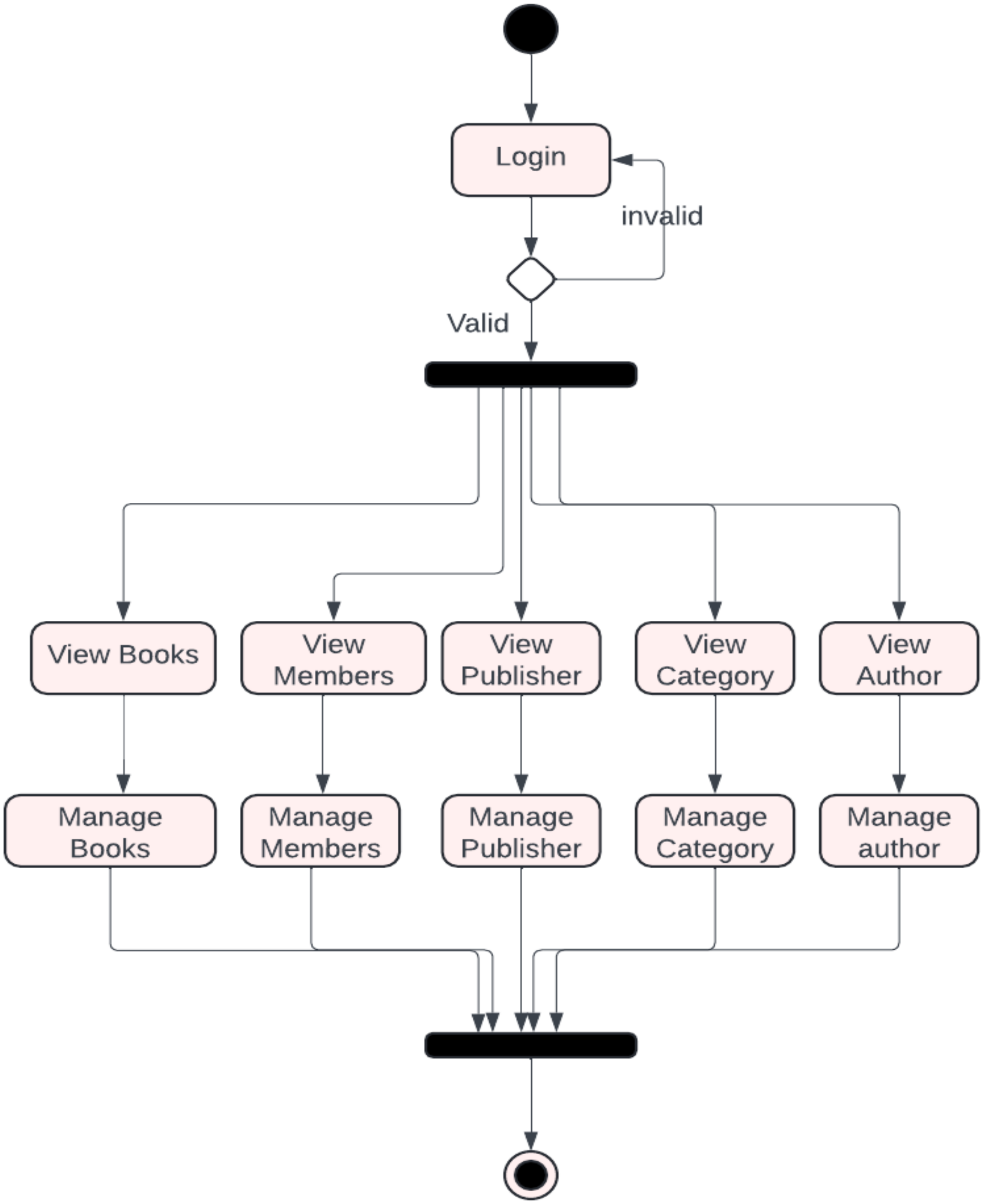
- **Login**
- **View books**
- **View Members**
- **View Category**
- **View Publisher**
- **View Author**
- **Manage books (CRUD)**
- **Manage Members (CRUD)**
- **Manage Category (CRUD)**
- **Manage Publisher (CRUD)**
- **Manage Author (CRUD)**

# Library Management System



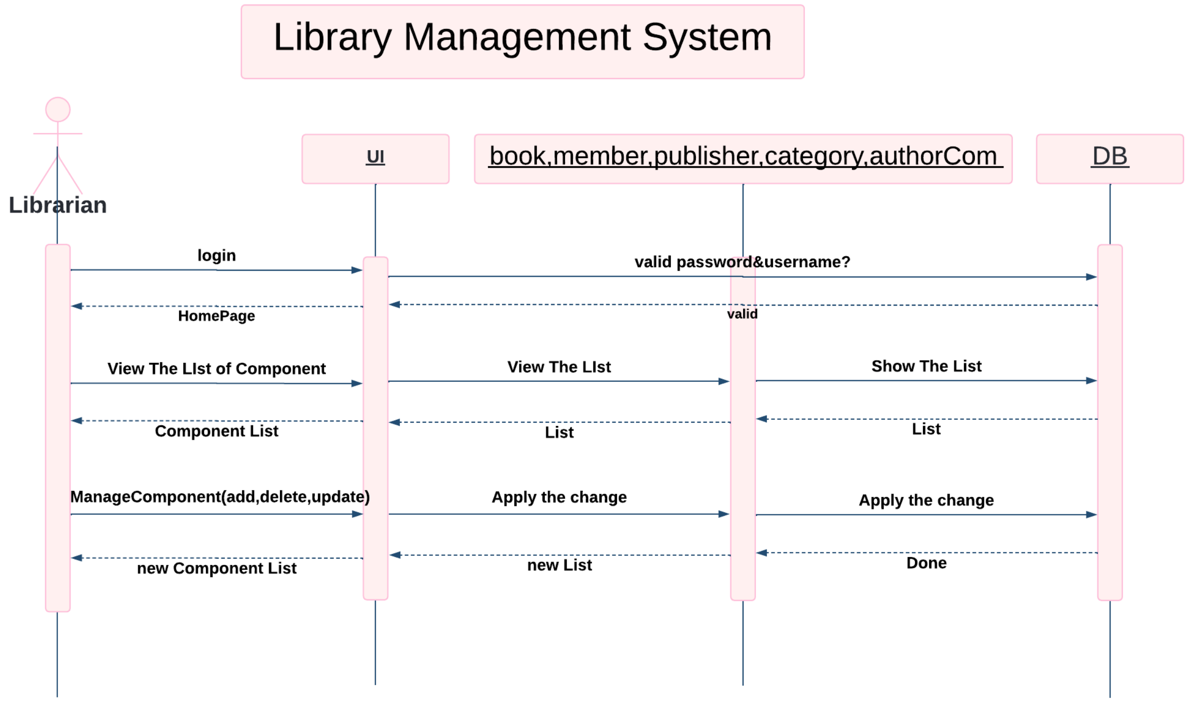
# Library Management System

Activity Diagram:



# Library Management System

## Sequence Diagram:



### the system components would typically include:

- User Interface (UI) Components:

  ***Login Interface:*** Allows Librarian to authenticate and access the system.

  ***View Interface for Books, Members, Category, Publisher, and Author:*** Provides interfaces for Librarian to view information about books, members, categories, publishers, and authors.

- Business Logic Components:

  ***Book Management Service:*** Handles CRUD (Create, Read, Update, Delete) operations related to books.

  ***Member Management Service:*** Handles CRUD operations related to library members.

***Category Management Service:*** Handles CRUD operations related to book categories.

***Publisher Management Service:*** Handles CRUD operations related to book publishers.

***Author Management Service:*** Handles CRUD operations related to authors.

- Data Access Components:

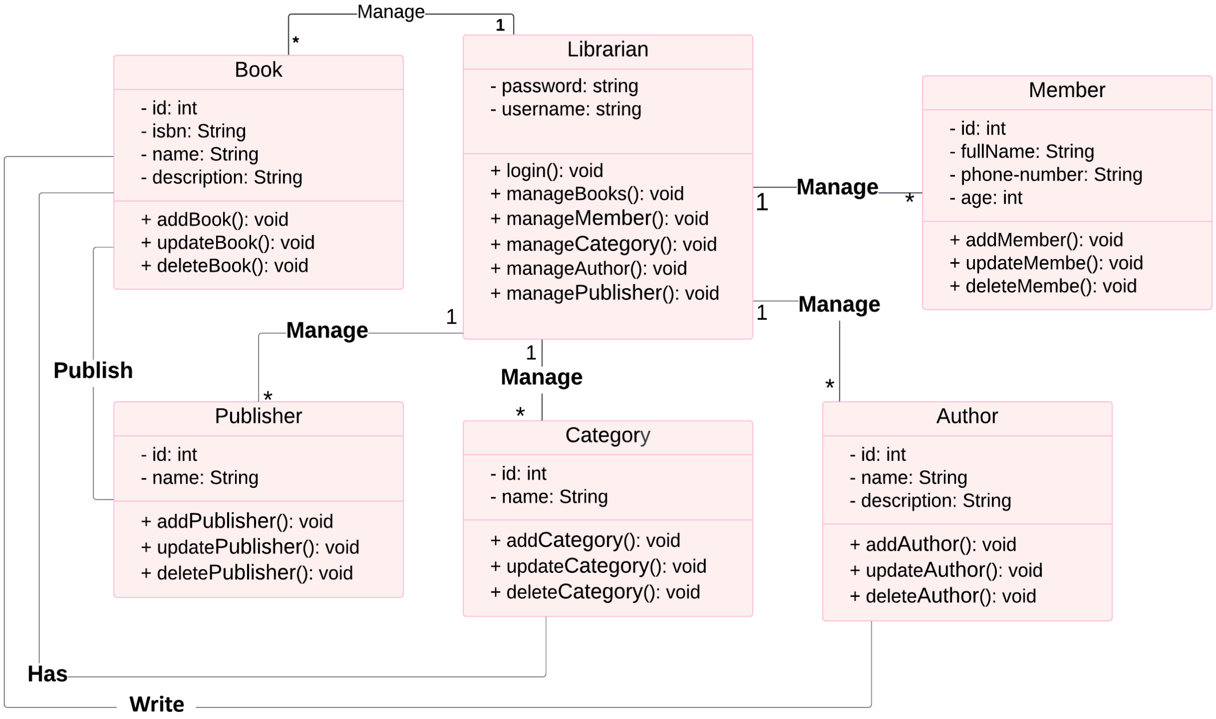
  ***Database Access Layer:*** Provides methods for interacting with the database to retrieve, store, update, and delete data related to books, members, categories, publishers, and authors.

- Actor Components:

  ***Librarian Actor Component:*** Represents the functionalities specific to librarians, such as managing books, members, categories, publishers, and authors.

# Library Management System

## Class Diagram:



## ***OCL**(Object Constraint Language):-

Context Librarian

Invariant ValidPassword: self.password.isValid()

- Rule: password is valid.

Context Librarian :: Login

pre: not password.isEmpty() and not username.isEmpty()

- Rule: You must enter your password and username to log in.

Context Librarian

inv: self.username.size() >= 4 and self.password.size() >= 6

- Rule: Ensure that both the username and password meet certain length requirements.

# Library Management System

```
context Member :: addMember

pre: self.age>=8
```

- Rule: Ensure that the members of the library are older than eight years old.

```
Context Member

inv: self.allInstances()->isUnique(id)
```

- Rule: Ensure that each member in the system has a unique id.

```
context Book

inv: self.allInstances()-> isUnique(id)
```

- Rule: Ensure that each book in the system has a unique Id.

```
Context Publisher

inv: self.allInstances()->isUnique(id)
```

- Rule: Ensure that each Publisher in the system has a unique id.

```
context Author

inv: self.allInstances()-> isUnique(id)
```

- Rule: Ensure that each author in the system has a unique Id.

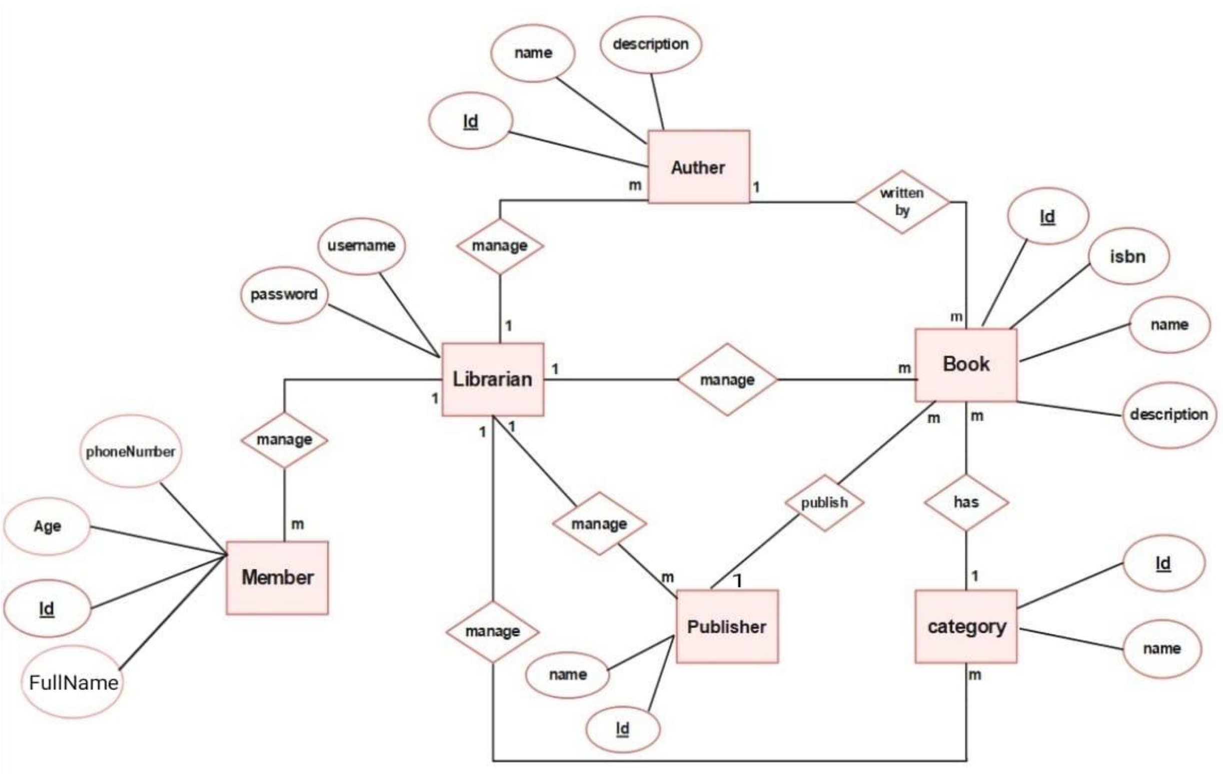
```
context Category

inv: self.allInstances()-> isUnique(id)
```

- Rule: Ensure that each category in the system has a unique Id.

# Library Management System

ERD Diagram:



**Entities:**

- *Librarian:*
  Attributes: username, password.
- *Book:*
  Attributes: ID (Primary Key), ISBN , Name, Description.
- *Member:*
  Attributes: ID (Primary Key), FullName, PhoneNumber, Age.
- *Publisher:*
  Attributes: ID (Primary Key) , Name.
- *Category:*
  Attributes: ID(Primary Key), Name.
- *Author:*
  Attributes: ID(Primary Key), Name, Description.

# Library Management System

**Relationships:**

- *Manage:*

  *(Association between Librarian and Book):*

  - A Librarian manages the catalogue, which includes adding, updating, or removing books.
  - Many Books is managed by a one Librarian.
  - Attributes: Management Date, Operation Type (Add/Update/Remove)

  *(Association between Librarian and Member):*

  - A Librarian manages the adding, updating, or removing members.
  - Many members is managed by a one Librarian.

  *(Association between Librarian and Publisher):*

  - A Librarian manages the catalogue, which includes adding, updating, or removing publishers.
  - Many Publishers is managed by a one Librarian.

  *(Association between Librarian and Category):*

  - A Librarian manages the catalogue, which includes adding, updating, or removing Categories.
  - Many Categories is managed by a one Librarian.

  *(Association between Librarian and Author):*

  - A Librarian manages the catalogue, which includes adding, updating, or removing authors.
  - Many Authors is managed by a one Librarian.

- *Publish:*

  *(Association between Book and Publisher):*

  - A Book is published by a Publisher.

  - A Publisher can have multiple books.

- *Has:*
  *(Association between Book and Category):*

  - A Book has a one category.
  - Category has many Books.

- *Write:*
  *(Association between Book and Author):*

  - A Book is written by a Author.
  - Author write many Books.