本系统使用 **RM2PT**工具，旨在将需求模型自动转换为系统架构设计和面向对象详细设计。

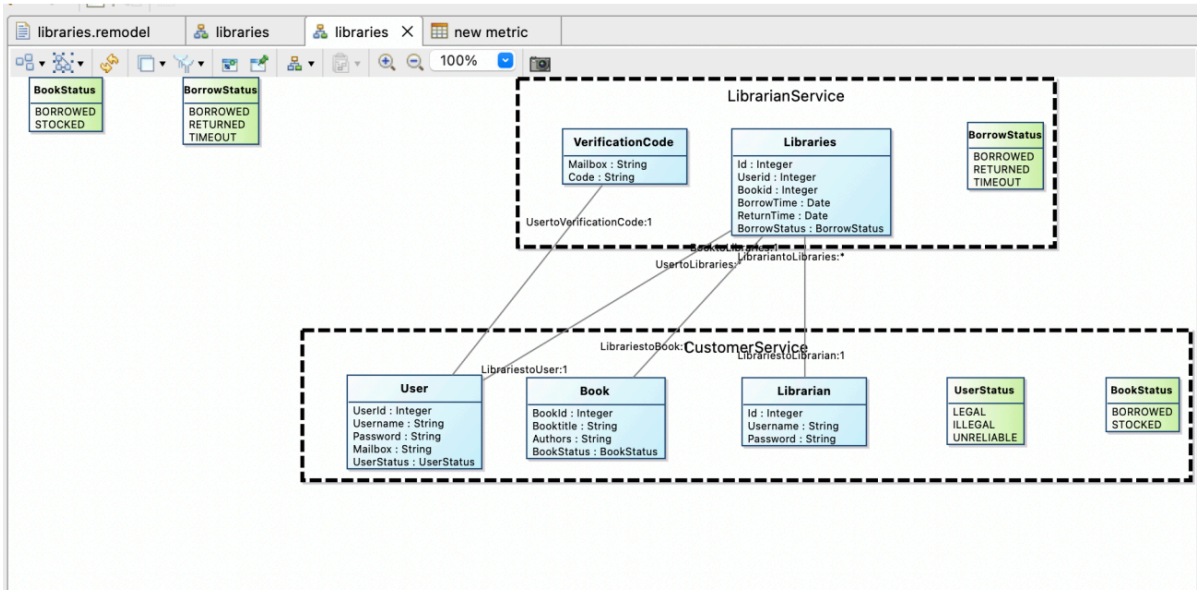
任务1：架构设计自动生成（RapidMS）

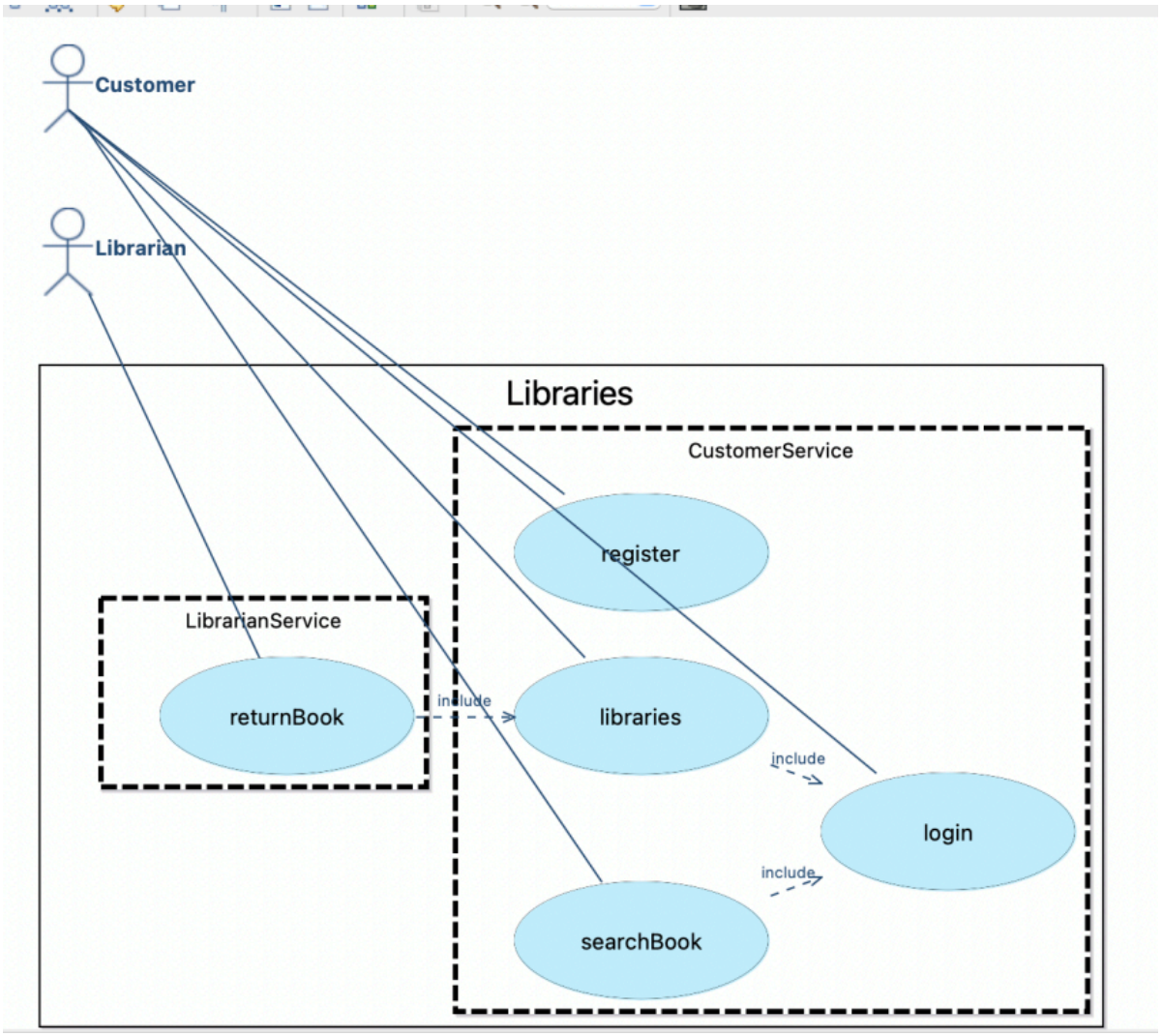
通过需求模型，自动生成系统的概念架构设计，主要包括：

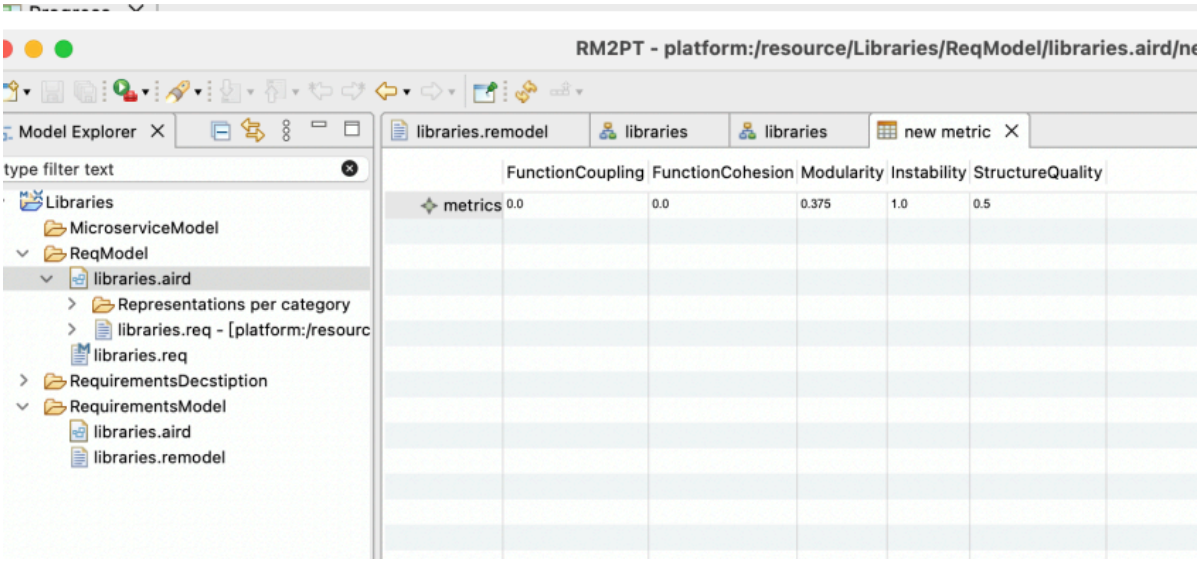
**Conceptual Class Diagram：描述系统中的主要业务实体及其关系，帮助开发人员理解系统核心概念及其联系。

MicroServiceModel：将系统拆分为多个微服务，定义各个服务的职责边界及交互方式，支持系统的分布式部署与弹性扩展。

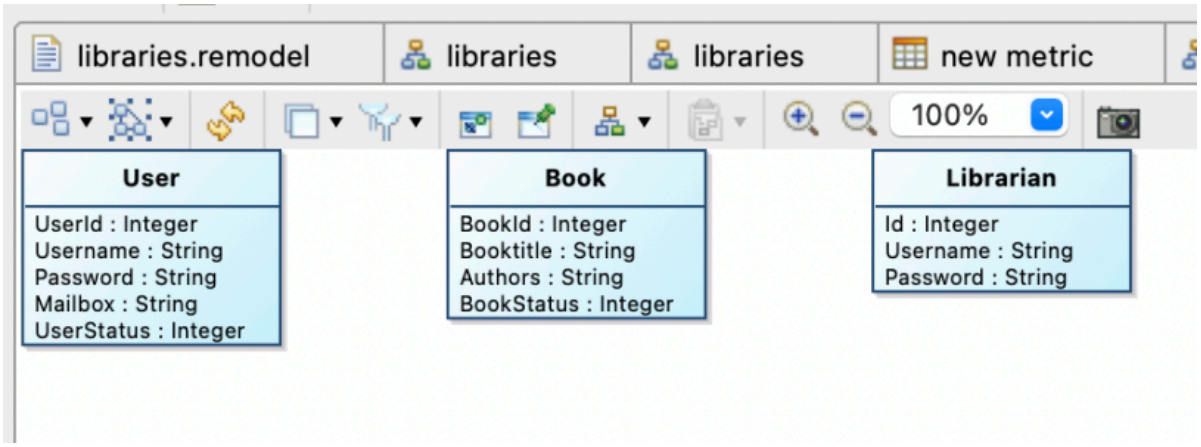
截图如下：



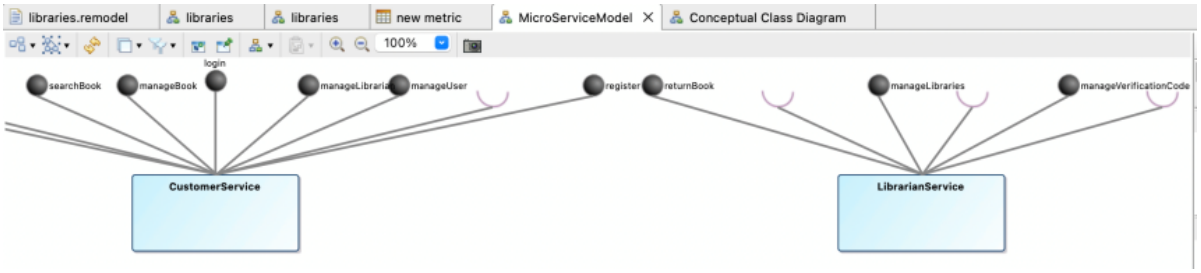




Conceptual Class Diagram



MicroServiceModel



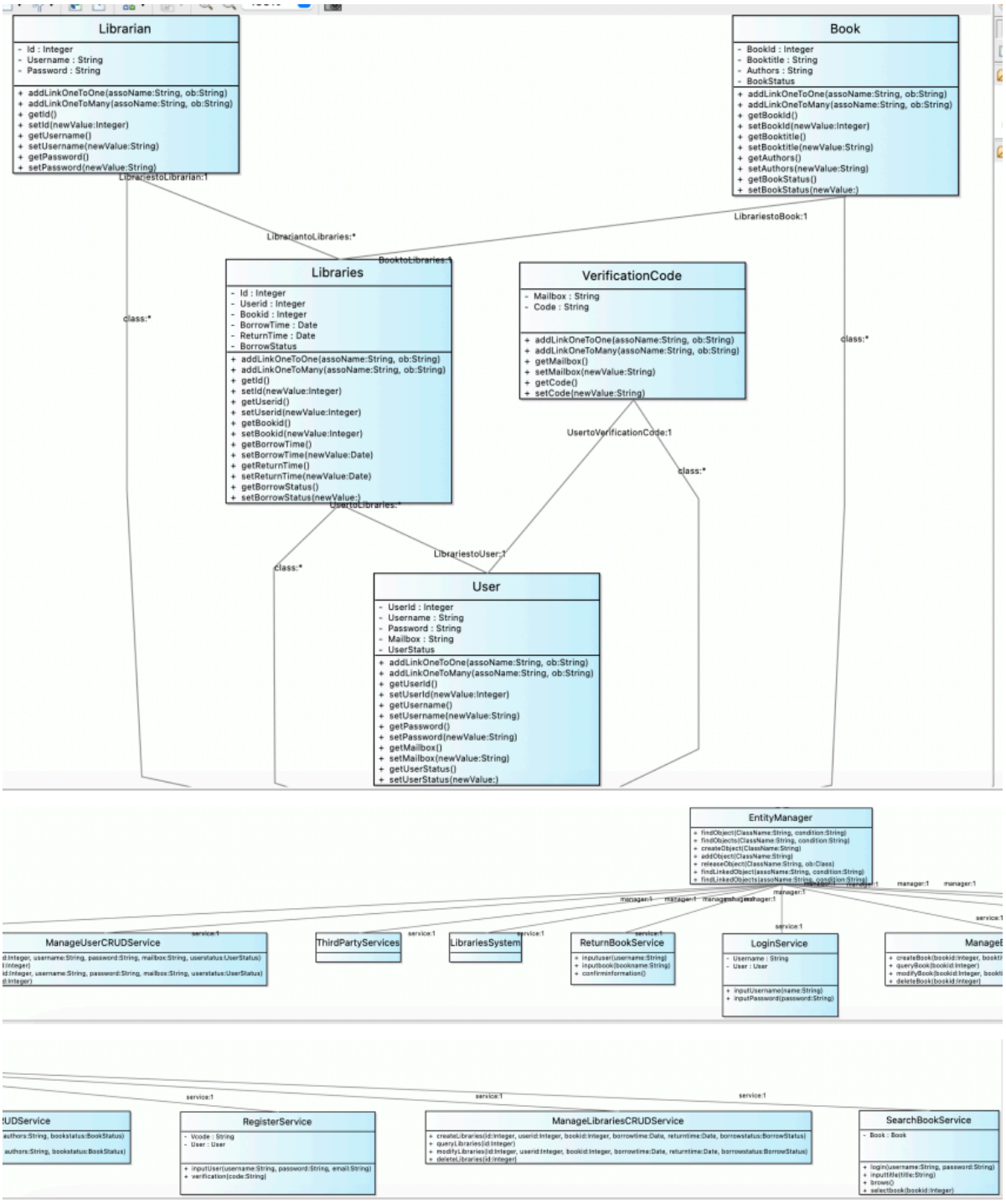
任务2：面向对象详细设计自动生成

在架构设计基础上，进一步自动生成系统的详细设计，主要包括：

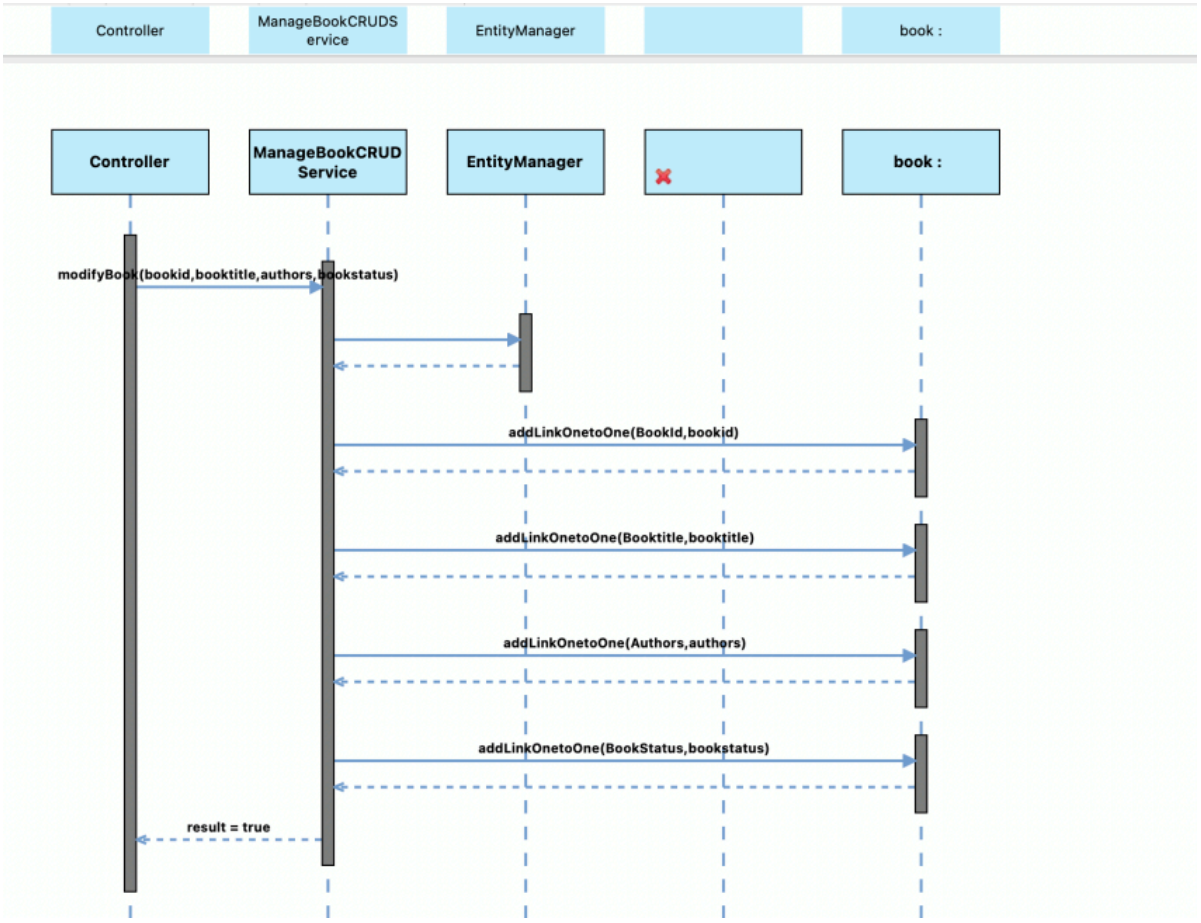
**Class Diagram：详细描述系统各个类的属性、方法以及类之间的关系，作为后续代码实现的蓝图。

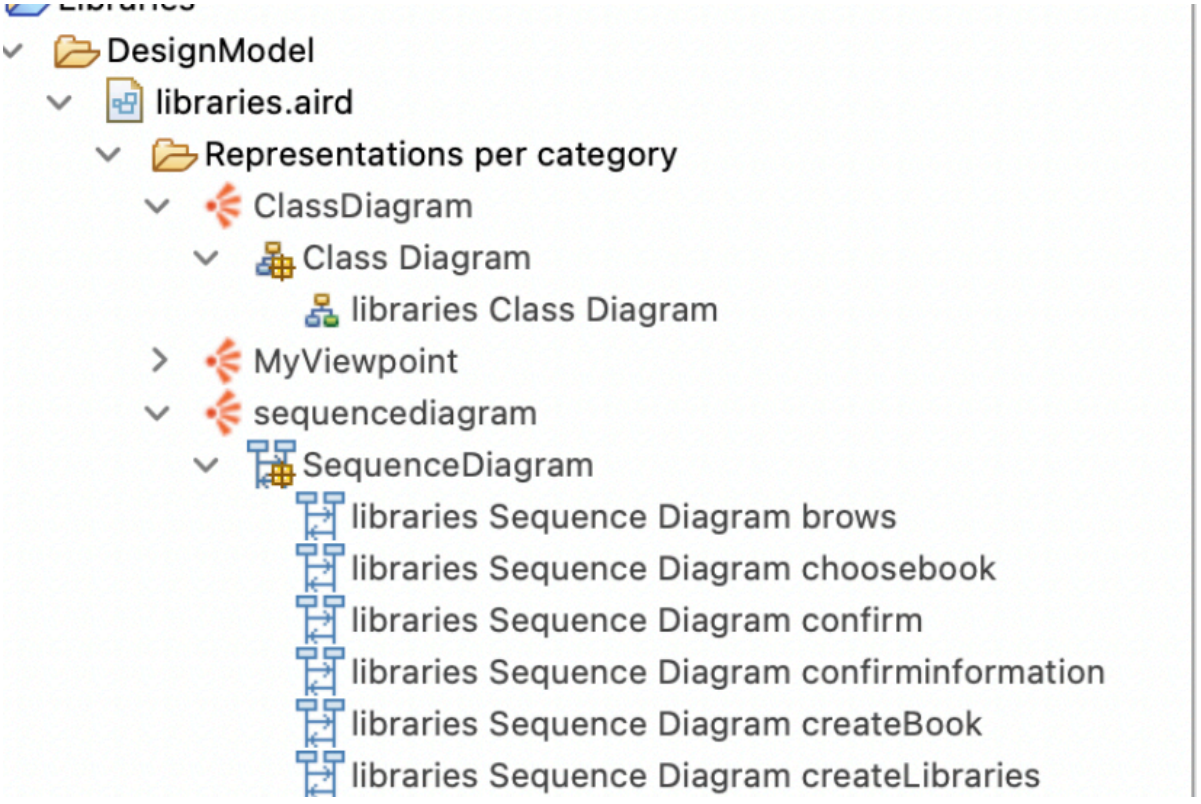
Sequence Diagram：展示不同对象之间的交互流程，体现业务用例在系统中的执行步骤，帮助开发人员理解系统运行机制。

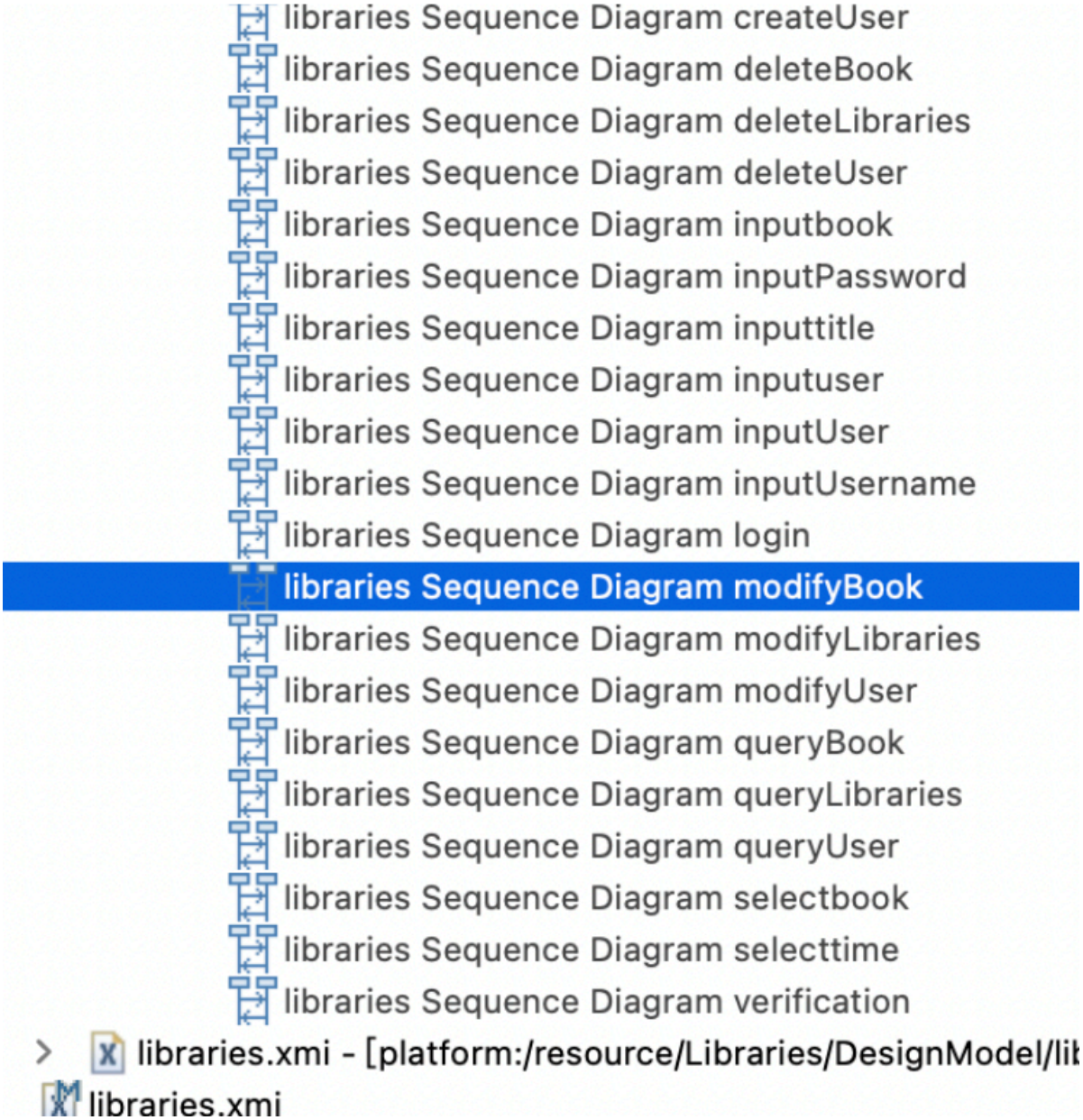
Class Diagram



若干Sequence Diagram如下：


Controller
ManageBookCRUDS ervice
EntityManager
book :
Controller
ManageBookCRUD Service
EntityManager
book :
modifyBook(bookid,booktitle,authors,bookstatus)
addLinkOnetoOne(BookId,bookid)
addLinkOnetoOne(Booktitle,booktitle)
addLinkOnetoOne(Authors,authors)
addLinkOnetoOne(BookStatus,bookstatus)
result = true


DesignModel
libraries.aird
Representations per category
ClassDiagram
Class Diagram
libraries Class Diagram
MyViewpoint
sequencediagram
SequenceDiagram
libraries Sequence Diagram brows
libraries Sequence Diagram choosebook
libraries Sequence Diagram confirm
libraries Sequence Diagram confirminformation
libraries Sequence Diagram createBook
libraries Sequence Diagram createLibraries



任务3 大模型生成设计模型与微服务拆分

通用Prompt

你是一个专业的软件架构师，精通领域建模、系统设计和需求分析。现在请你根据给定的需求模型（需求用例、需求规范、实体及关系说明），自动生成对应的系统设计模型。设计模型应符合面向对象和微服务架构的最佳实践，并与需求模型保持一致。

需求：

基于以下需求模型描述：

{{实验一的需求描述}}

请生成：

1. 架构设计模型：

Conceptual Class Diagram（概念类图）：请列出系统中核心概念及其关系，使用面向对象设计表示。

MicroServiceModel（微服务模型）：请根据业务边界划分微服务，说明每个微服务的职责与接口，画出微服务之间的交互关系。

2. 面向对象详细设计模型：

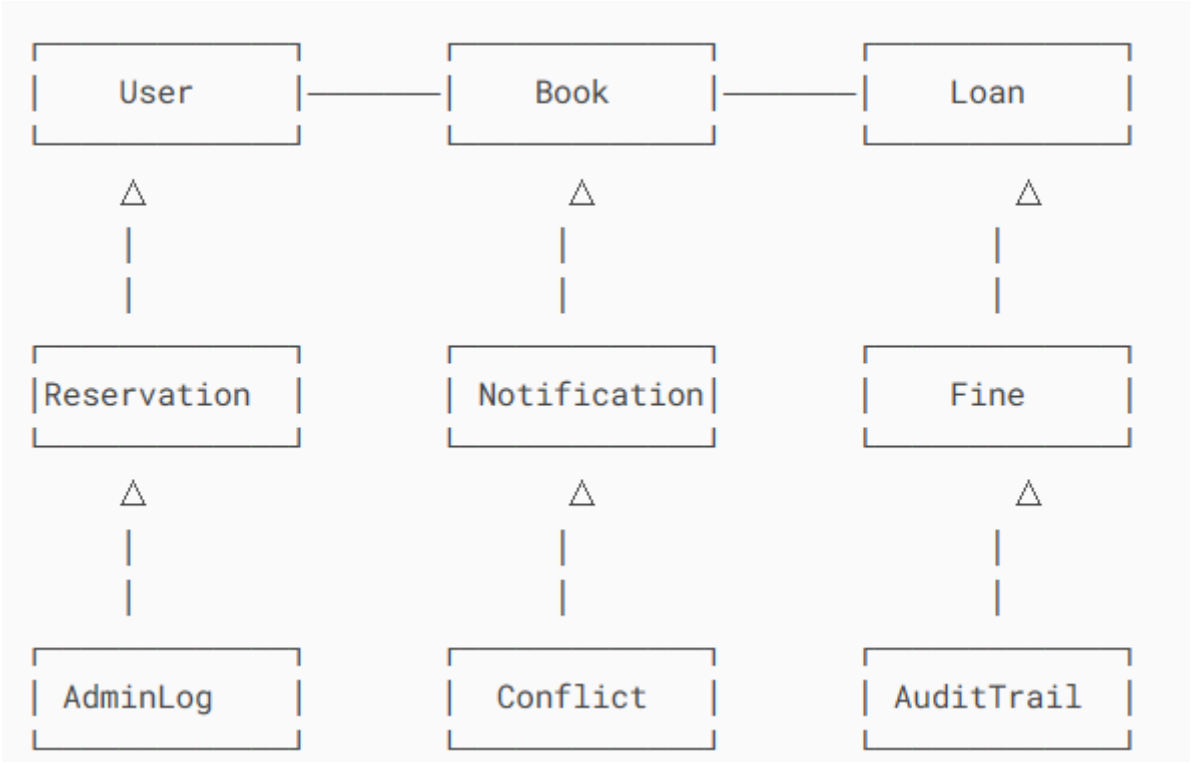
Class Diagram（类图）：在概念类图的基础上，补充类的属性、方法和接口，体现系统详细设计。

Sequence Diagram（序列图）：针对核心业务场景，生成对应的交互流程图，描述对象间消息传递顺序。

大模型输出

概念类图

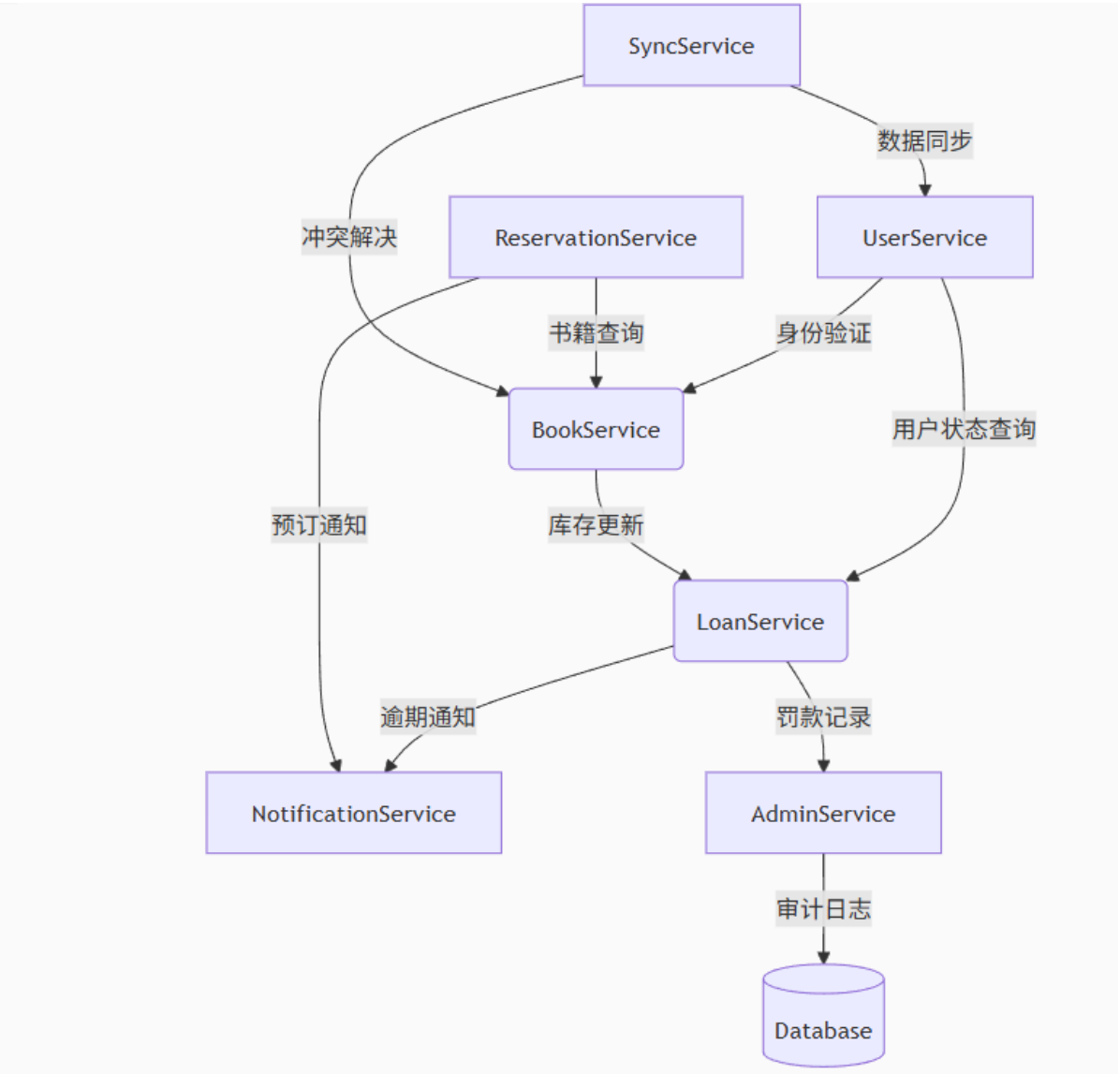
概念类图（Conceptual Class Diagram）



核心关系说明：

- **User**与**Book**通过**Loan**建立借阅关系（1对多）
- **User**可创建多个**Reservation**（1对多）
- **Loan**可能产生**Fine**（1对1）
- 系统通过**Notification**与用户通信
- **AdminLog**记录管理员操作，**AuditTrail**跟踪系统活动
- **Conflict**解决数据同步问题

微服务模型（MicroServiceModel）



1. 交互说明：
    1. **核心服务三角**：UserService ↔ BookService ↔ LoanService
    2. **NotificationService**作为消息中枢接收各类事件
    3. **AdminService**具有跨服务权限，记录审计日志
    4. **SyncService**通过事件驱动模式同步各服务数据
    5. 虚线表示异步消息，实线表示同步调用

类图（Class Diagram）

在概念类图基础上，增加属性与方法：

```
// 用户域
class User {
  -String userId
  -String username
  -PasswordCredential credential
  -UserProfile profile
  -AccountStatus status
  +AuthToken login()
  +void updateProfile()
```

```
  +void freezeAccount()
}

enum AccountStatus {
  ACTIVE, FROZEN, SUSPENDED, PENDING
}

// 图书域
class Book {
  -String bookId
  -ISBN isbn
  -BookMetadata metadata
  -Inventory inventory
  +boolean validate()
  +void adjustStock()
}

class Inventory {
  -int totalCopies
  -int available
  -int reserved
  +boolean isAvailable()
}

// 借阅域
class Loan {
  -String loanId
  -BookItem item
  -User borrower
  -LoanPeriod period
  -Fine fine
  +void renew()
  +ReturnReceipt return()
}

class LoanPeriod {
  -LocalDate start
  -LocalDate due
  -int renewCount
  +boolean isOverdue()
}

// 通知域
class Notification {
  -String notificationId
  -NotificationType type
  -NotificationTarget target
  -NotificationContent content
  +void send()
  +DeliveryReport getStatus()
}
```

核心场景序列图（以“预定”和“借书”为例）

📚 场景1：借书（Student 借书）

```
sequenceDiagram
    participant User
    participant Frontend
    participant LoanService
    participant BookService
    participant UserService
    participant NotificationService

    User->>Frontend: 选择书籍→借阅
    Frontend->>LoanService: POST /loans {userId, bookId}
    LoanService->>UserService: GET /users/{userId}/status
    UserService-->>LoanService: 200 OK (ACTIVE)
    LoanService->>BookService: PUT /books/{bookId}/hold
    alt 库存充足
        BookService-->>LoanService: 200 OK (库存已预留)
        LoanService->>LoanService: 创建借阅记录
        LoanService->>NotificationService: POST /notifications (到期提醒)
        LoanService-->>Frontend: 201 Created (借阅成功)
        Frontend->>User: 显示成功消息
    else 库存不足
        BookService-->>LoanService: 409 Conflict
        LoanService-->>Frontend: 400 Bad Request
        Frontend->>User: 显示"库存不足"
    end
```

场景2：借书（Student 预定图书）

```
sequenceDiagram
    participant User
    participant Frontend
    participant ReservationService
    participant BookService
    participant NotificationService
    participant LoanService

    User->>Frontend: 查看无库存书籍
    Frontend->>BookService: GET /books/{bookId}/availability
    BookService-->>Frontend: {"available":0, "waitlist":3}
    User->>Frontend: 点击"加入等待队列"
    Frontend->>ReservationService: POST /reservations {user,book}
    ReservationService->>BookService: GET /books/{bookId}/alternatives
    BookService-->>ReservationService: [相似书籍列表]
    ReservationService->>ReservationService: 计算队列位置
    ReservationService->>NotificationService: 注册书籍可用通知
    ReservationService-->>Frontend: 201 Created (位置:4)
    Frontend->>User: 显示预订成功和推荐

    loop 库存监控
        BookService->>BookService: 检查归还书籍
        alt 预订书籍归还
            BookService->>ReservationService: PUT /reservations/notify-available
```

```
            ReservationService->>LoanService: 检查用户资格
            LoanService-->>ReservationService: 可借阅
            ReservationService->>NotificationService: 发送可用通知
            NotificationService->>User: 短信/邮件通知
        end
    end
```