



中山大學
SUN YAT-SEN UNIVERSITY

中大图书平台

分析与设计

项目组成员：杨奕嘉 15331366

邱旭铨 15331259

于海 15331380

郑品 15331422

尹张昊翔 15331379

张增辉 15332020

张千艺 15332018

指导老师： 衣杨

一、 引言.....	错误! 未定义书签。
1.1 文档用途.....	错误! 未定义书签。
1.2 阅读对象.....	错误! 未定义书签。
二、 需求分析.....	错误! 未定义书签。
2.1 问题描述.....	错误! 未定义书签。
2.2 角色构成.....	错误! 未定义书签。
2.3 用例图.....	错误! 未定义书签。
2.4 用例归约.....	错误! 未定义书签。
2.4.1 登录.....	错误! 未定义书签。
2.4.2 发布借出/求借图书信息.....	错误! 未定义书签。
2.4.3 删除借出/求借图书信息.....	错误! 未定义书签。
2.4.4 修改用户信息.....	错误! 未定义书签。
2.4.5 查询借出的的图书信息.....	错误! 未定义书签。
2.5 术语表.....	错误! 未定义书签。
2.6 补充归约.....	错误! 未定义书签。
2.6.1 可靠性.....	错误! 未定义书签。
2.6.2 性能.....	错误! 未定义书签。
2.6.3 可支持性.....	错误! 未定义书签。
2.6.4 安全性.....	错误! 未定义书签。
2.6.5 设计约束.....	错误! 未定义书签。
三、 架构设计.....	错误! 未定义书签。
3.1 架构设计.....	错误! 未定义书签。
3.1.1 模型层.....	错误! 未定义书签。
3.1.2 视图层.....	错误! 未定义书签。
3.1.3 控制层.....	错误! 未定义书签。
3.1.4 服务层.....	错误! 未定义书签。
3.2 架构图.....	错误! 未定义书签。
3.3 关键抽象.....	错误! 未定义书签。
四、 用例分析.....	错误! 未定义书签。
4.1 用户类用例析取.....	错误! 未定义书签。
4.2 订单类用例析取.....	错误! 未定义书签。
4.3 用户登录用例交互过程.....	错误! 未定义书签。
4.4 用户发布借出/求借图书信息用例交互过程.....	错误! 未定义书签。
4.5 用户删除借出/求借图书信息用例交互过程.....	错误! 未定义书签。
4.6 用户修改个人信息用例交互过程.....	错误! 未定义书签。
4.7 用户查询借出图书信息用例交互过程.....	错误! 未定义书签。
4.8 分析机制.....	错误! 未定义书签。
4.9 平台总的类分析.....	错误! 未定义书签。
五、 类的设计.....	错误! 未定义书签。
5.1 实体类设计.....	错误! 未定义书签。
六、 子系统及其接口设计.....	错误! 未定义书签。
6.1 子系统划分.....	错误! 未定义书签。
6.2 子系统设计.....	错误! 未定义书签。
6.2.1 用户管理子系统.....	错误! 未定义书签。

6.2.2 图书管理子系统.....	错误！未定义书签。
七、中大图书平台系统部件设计.....	错误！未定义书签。
7.1 中大图书平台系统运行时进程分析.....	错误！未定义书签。

一、引言

1.1 文档用途

本文档的编写目的是让读者更好的明白中大图书平台的设计过程和设计特点，能反映出一个整体的系统框架。

1.2 阅读对象

此文档适合以下人员阅读：

- 1) 中大图书平台项目组成员；
- 2) 对本项目感兴趣的人员。

二、需求分析

2.1 问题描述

在大学现有的条件下，图书馆很难满足学生对于某些辅导资料、课本的大量需求。这是由于图书馆采用的是租借模式，当学生需要长期持有某本书时，需要向图书馆反复续期就显得十分不便。

而与此同时，当应届生毕业时，会有大量的废弃书籍没法处理只能丢弃或出售给二手书店，不能让书籍在校内进行高效循环利用。

因此我们小组围绕这个问题，开发出可供于学生们自主交换、借阅书籍的平台，提供以下功能：

- ① 发布图书借出信息
- ② 通过借出信息联系借出人
- ③ 发布图书求借信息
- ④ 通过借出信息联系借出人

考虑到目前学生对微信的使用较多，我们决定使用微信小程序作为用户接入的界面，用户可以通过微信进行登录操作。

除此此外某些名著或经典读物因价格较高，图书馆或没有藏书或藏书较少，而有些读者会收藏该书，因此我们提供的平台对于结识书友也起到一定的积极影响。

2.2 角色构成

平台全部基于个人用户的角色构成：

用户：管理个人图书借出、求借信息，管理个人账户信息

2.3 用例图

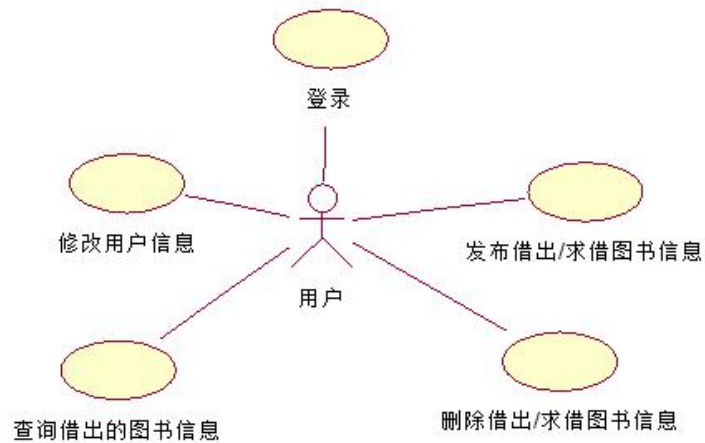


图 1 用例图。

2.4 用例归约

项目用例有登陆、发布借出/求借图书信息、删除借出/求借图书信息、修改用户个人信息、查询借出图书信息。

2.4.1 登录

(1) 简要描述

用户可以直接用微信账号登录。

(2) 参与者

用户。

(3) 场景描述

参与者在点开该小程序后，点击“我的”按钮可以直接使用微信账号进行登录操作。

(4) 前置条件

参与者点开小程序后，点击“我的”进入登录界面。

(5) 后置条件

返回参与者是否成功登录的信息。

(6) 事件流



图2 登录事件流。

2.4.2 发布借出/求借图书信息

(1) 简要描述

登录后的用户可以在平台上发布借出/求借图书信息，图书信息包含图书名、ISBN 编号、图书图片以及对图书的简要描述，并且选择供借阅/需借用图书的起止时间。

(2) 参与者

用户。

(3) 场景描述

用户在平台登陆后，点击发布借出图书，输入图书名称，ISBN 编号，上传图书图片，并对图书作简要描述，然后选择借出时间以及要求归还的时间，最后点击提交按钮发布图书信息。

用户在平台登陆后，点击发布求借图书，输入求借图书名称，ISBN 编号，上传求借图书的图片，可以对需求书籍作简要描述，然后选择开始借阅时间以及归还时间，最后点击提交按钮发布图书信息。

(4) 前置条件

用户已登录，用户状态是可用状态。

(5) 后置条件

将借出/求借图书信息存储进入数据库之中，并返回是否发布成功的信息，反馈用户。

(6) 事件流

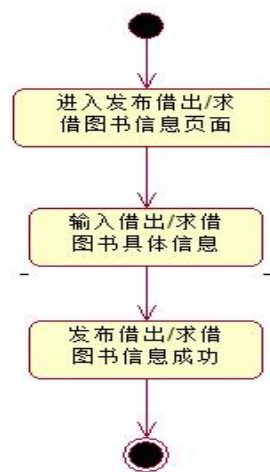


图 3 发布借出/求借图书信息事件流。

2.4.3 删除借出/求借图书信息

(1) 简要描述

登录后的用户可以删除自己的借出/求借图书信息。

(2) 参与者

用户。

(3) 场景描述

用户在登录后，若需要删除借出图书订单信息或求借图书订单信息，可以在自己的借出/求借图书信息列表界面，选择要删除的借出/求借图书信息，点击删除按钮，再次确认后删除相关图书信息。

(4) 前置条件

用户已登录，用户状态是可用状态。

(5) 后置条件

将借出/求借图书的订单信息从数据库存储中删除，返回是否删除成功的信息。

(6) 事件流

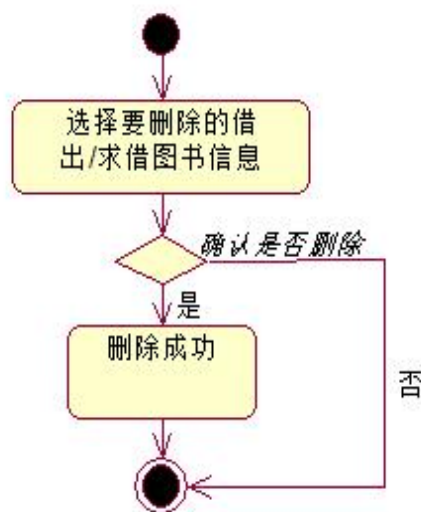


图 4 删除借出/求借图书信息事件流。

2.4.4 修改用户信息

(1) 简要描述

修改登陆后的用户个人信息，包含用户名、微信号、邮箱、性别和个人简介。

(2) 参与者

用户。

(3) 场景描述

用户在登陆后，若需要修改个人资料，可以在自己的个人信息界面，修改个人信息，包含用户名、微信号、邮箱、性别和个人简介，修改完成后保存修改后的个人信息。

(4) 前置条件

用户已登录，用户状态是可用状态。

(5) 后置条件

用户修改个人信息后将返回修改成功的信息，提示修改成功。数据库更新，保存用户修改后的信息。

(6) 事件流



图 5 修改用户信息事件流。

2.4.5 查询借出的的图书信息

(1) 简要描述

用户可以进入个人中心，查询自己借出的图书信息，方便对自己的书籍去处有一个清晰的感知。

(2) 参与者

用户。

（3） 场景描述

用户若遗忘自己当前借出去了哪些书以及这些书的去处,可以通过进入个人中心查询借出的图书信息,其中还包括具体借到谁手中。

（4） 前置条件

点击查询借出图书信息按钮。

（5） 后置条件

获得已借出去的图书的信息详细列表。

（6） 事件流

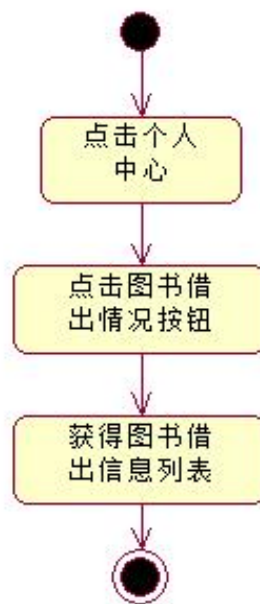


图 6 查询借出的图书信息事件流。

2.5 术语表

表 1 术语表。

名词术语	定义	英文名
借出图书	借出者提供的图书	Lend Book
求借图书	求借者希望借取的图书	Borrow Book
用户	使用此平台的用户	User

2.6 补充归约

2.6.1 可靠性

系统可用性：中大图书系统能够 24 小时运行。

2.6.2 性能

响应时间：对于用户的操作，平均反应时间不超过 2 秒。
可容纳量：允许 1000 名用户注册并且不会造成系统卡顿。

2.6.3 可支持性

访问权限：用户具有个人账号信息设置、查询借出图书信息、发布借出图书信息、查询求借图书信息、发布求借图书信息的权限。

2.6.4 安全性

阻止非法用户获取他人的个人信息以及订单信息。

2.6.5 设计约束

平台采取前后端分离方式开发，前端采用微信小程序作为接入界面，后端采用 Spring boot 架构，前后端的对接接口采用 RESTful 规范。
接口：必须提供基于微信小程序的接口。
联机用户文档：用户手册需要提供详细的用户使用帮助说明，包括系统的基本介绍、功

能以及开发者信息。

法律、版权声明：系统需要严格保护用户信息，承诺不公开、编辑或透露用户个人信息。用户需要遵守响应的条款。本平台受约于中华人民共和国国家法律。

三、架构设计

3.1 架构设计

本平台开发采用 MVCS 架构，即分成模型层、视图层、控制层、服务层四层架构。4 层架构的方式使得数据持久化、视图显示、访问控制和数据处理得以分离开来，使得只需要统一接口规范后，每层能够进行独立开发，不会相互影响。这样的架构不仅使整个网站的开发清晰明了，开发模块化，而且大大提高了开发的效率。

3.1.1 模型层

本平台的数据主要分为两部分：

- (1) 用户数据。包括了所有用户账号信息。
- (2) 借出/求借图书信息。包括了图书的信息和借出/求借人的联系信息。

模型层负责将数据持久化到数据库中，不处理任何逻辑判断。

3.1.2 视图层

本平台的视图层主要包括以下界面：

- (1) 主页
- (2) 搜索页
- (3) 具体借出/求借图书信息页
- (4) 借出/求借图书信息列表页
- (5) 个人信息管理页

视图层负责显示样式以及对提交数据的简单检查（空值、类型错误）。

3.1.3 控制层

本平台的控制器主要包括以下部分：

- (1) 用户注册登录的控制逻辑。处理用户的登录、注册行为，并在相应动作后返回操作成功信息或错误信息。
- (2) 处理借出/求借图书信息的控制逻辑。处理创建、修改、查询、删除借出/求借图书信息行为，并在相应动作后返回操作成功信息、错误信息或图书信息。
- (3) 用户管理个人信息的控制逻辑。处理编辑、查询用户信息的行为，并在相应动作后返回操作成功信息、错误信息或用户详细信息。

控制层负责对用户权限的处理，阻止用户进行越权访问。

3.1.4 服务层

- 服务层主要包括以下部分：
- （1）向模型层查询、更新用户信息
 - （2）向模型层查询、更新借出/求借图书信息
- 服务层负责对信息的合法性进行检查和确认。

3.2 架构图

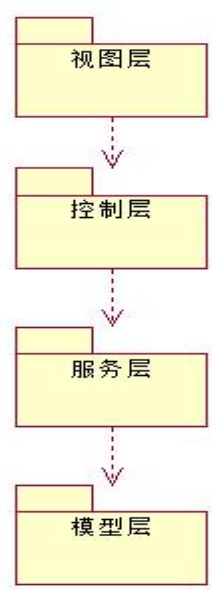


图 7 架构图。

3.3 关键抽象

系统关键抽象即实体类图，系统实体类描述了系统中的类及其相互之间的各种关系，它反映了系统中包含的各种对象的类型以及对象间的各种静态关系，主要描述了系统实体层中各实体类的属性及其相互关系。

根据对象的不同属性与功能，可以在项目中抽象出以下几个类：

- (1) **User**：用户类。所有用户都被抽象为 **User** 类，包括用户信息的数据，用户查询、修改个人信息、执行借出/求借图书的行为。
- (2) **Order**：订单类。针对用户借出/求借的图书对象可抽象成一个订单类，包括图书信息、图书状态（借出/未借出）、图书借出者、图书求借者信息、借书期限。

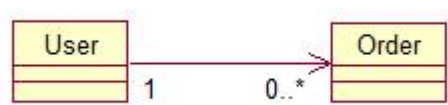


图 8 关键抽象类。

四、用例分析

4.1 用户类用例析取

用户通过微信小程序端口进行用户的注册、登录、修改用户信息，通过 `UserInterface` 访问后端进行操作。

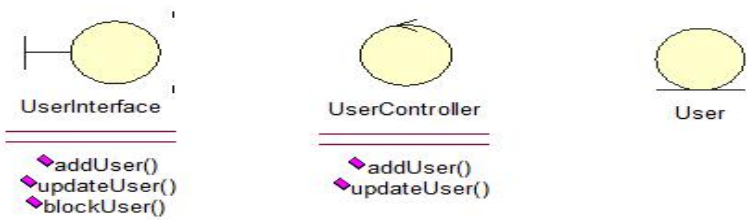


图 9 用户类用例析取。

4.2 订单类用例析取

用户通过微信小程序端口进行添加、删除、修改借出/借入图书信息，通过 `OrderInterface` 访问后端进行操作。

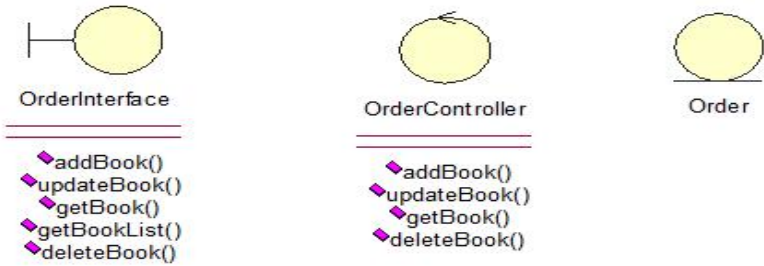


图 10 订单类用例析取。

4.3 用户登录用例交互过程

用户使用微信账号登录系统，系统通过用户登录认证后返回成功信息。用户登录时序图如下：

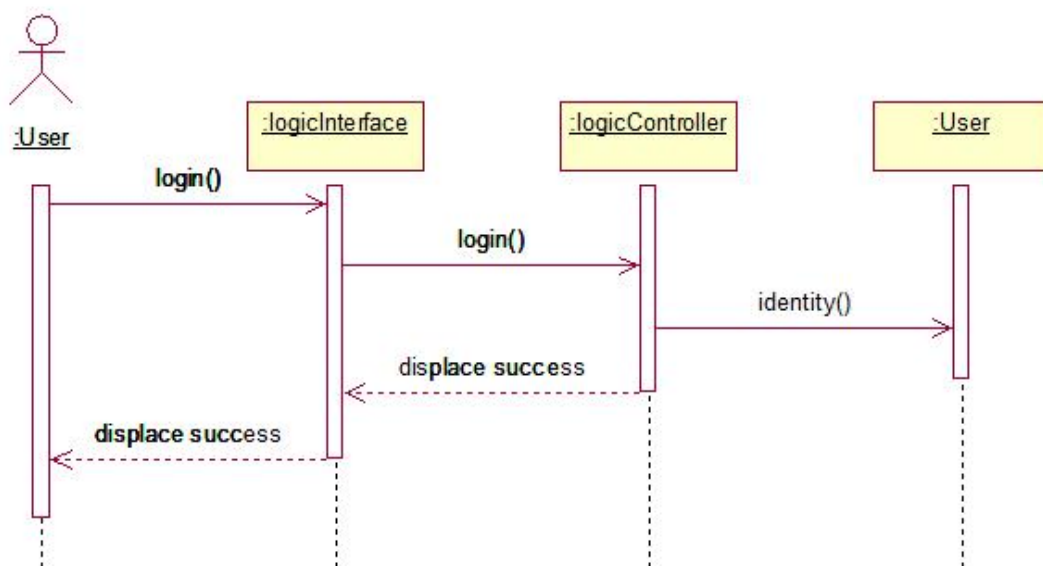


图 11 登录时序图。

4.4 用户发布借出/求借图书信息用例交互过程

用户输入借出/图书信息，由 OrderInterface 访问后端进行数据操作，添加成功后返回成功信息。发布借出求借图书信息时序图如下：

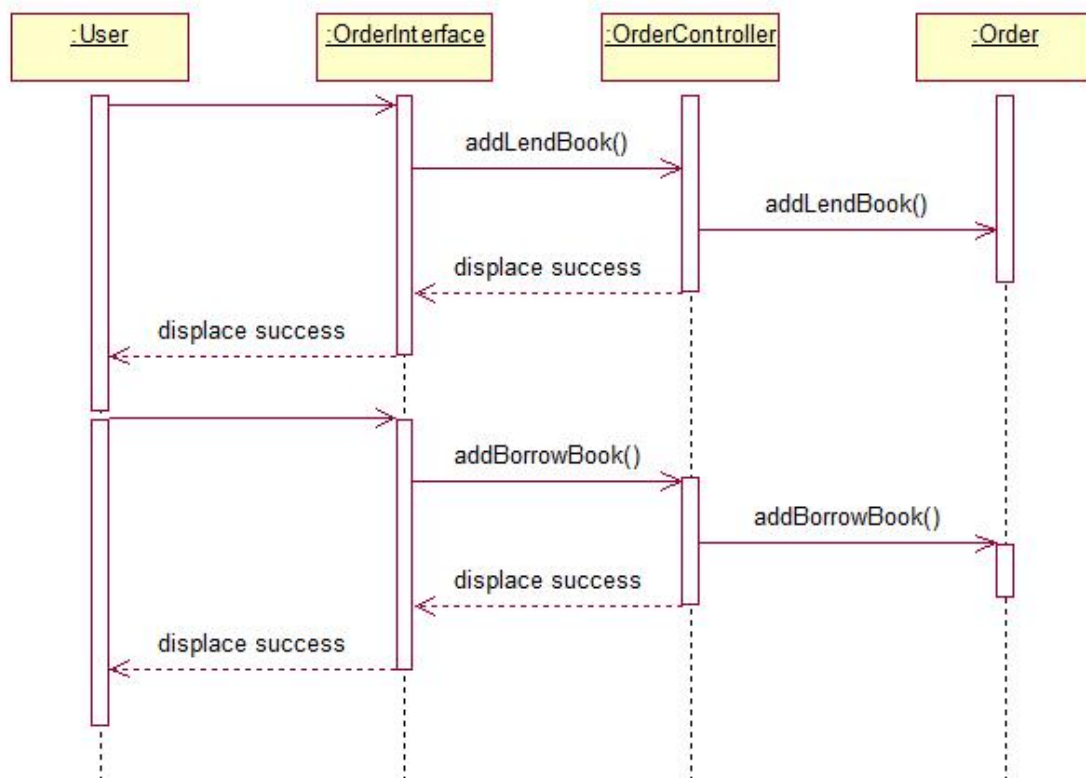


图 12 发布借出/求借图书信息时序图。

4.5 用户删除借出/求借图书信息用例交互过程

用户选择要删除的借出/求借图书信息，由 OrderInterface 访问后端进行数据操作，添加成功后返回成功信息。删除借出求借/图书信息时序图如下：

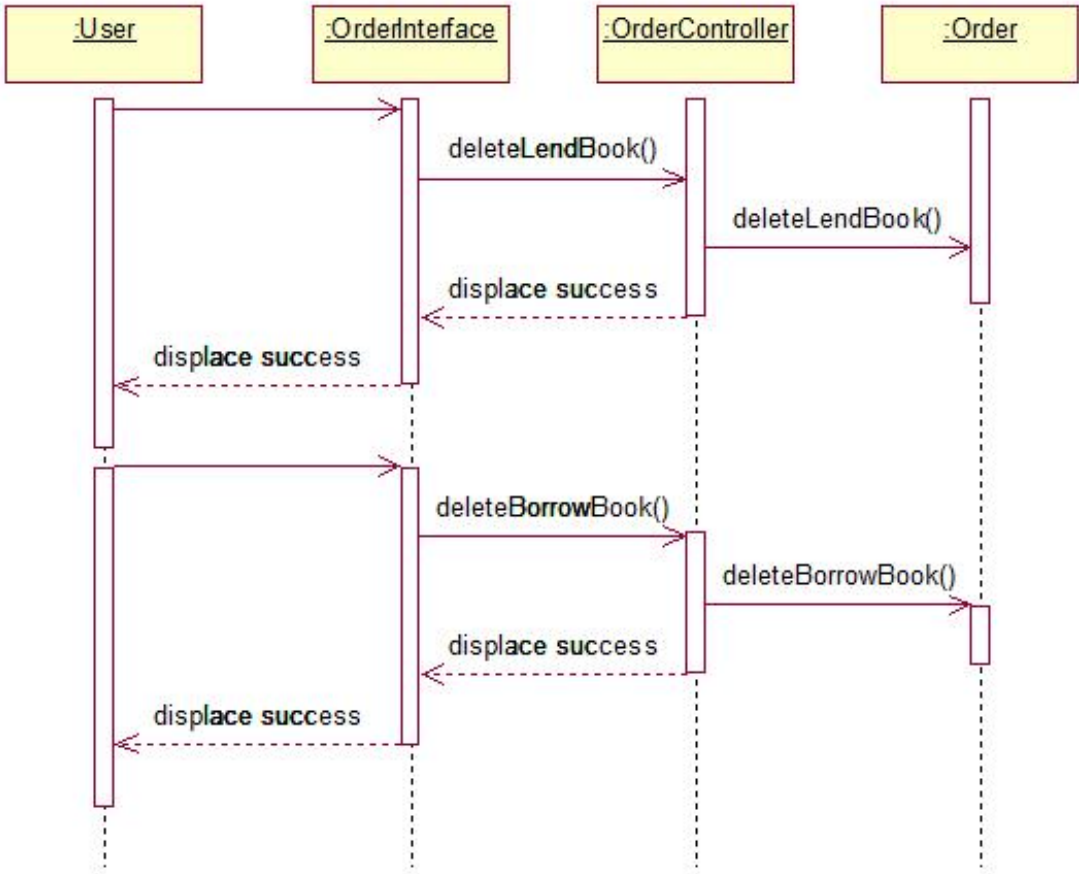


图 13 删除借出/求借图书信息时序图。

4.6 用户修改个人信息用例交互过程

用户在个人信息界面修改个人信息，由 UserInterface 访问后端进行数据操作，修改成功后返回成功信息。修改个人信息用例时序图如下：

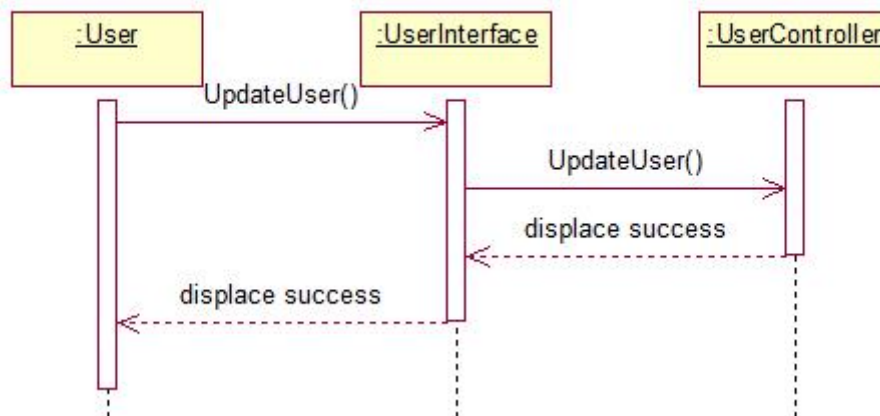


图 14 修改用户信息时序图。

4.7 用户查询借出图书信息用例交互过程

用户通过订单界面查询借出图书信息，由 OrderInterface 对后端尽心数据访问，返回图书列表和成功信息。查询借出图书信息时序图如下：

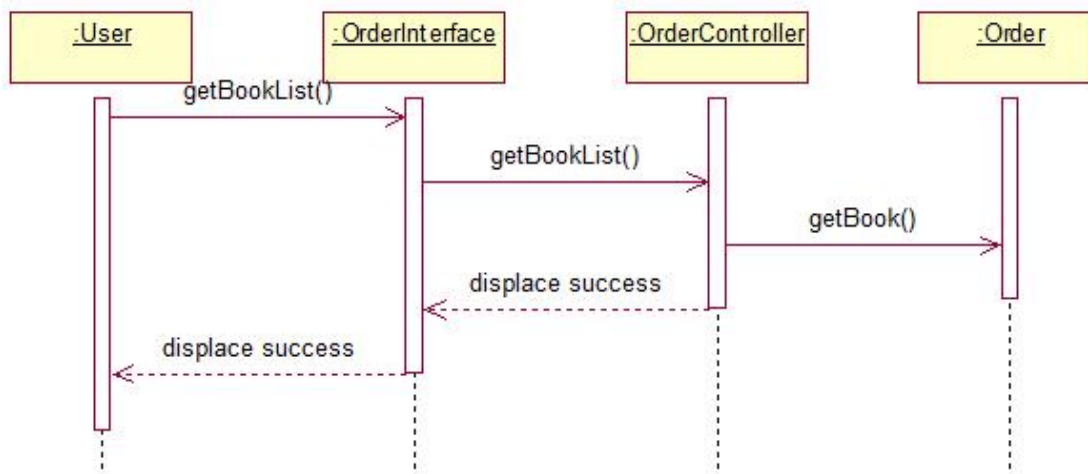


图 15 查询借出的图书信息时序图。

4.8 分析机制

表 2 分析机制。

类	分析机制
User	持久性、安全性
Order	持久性、安全性

User 类持久性：

粒度：每个用户占用 2k 字节以内

容量：上限 2000 用户

访问频率：

创建：每天 100 次

读取：每天 1000 次

更新：每天 100 次

删除：每天 100 次

User 类安全性：

用户信息只有自己能更改，其他人不能更改。其他用户不能查看用户敏感信息（openid, unionid, session_key）

Order 类持久性：

粒度：每个订单占用 2k 字节以内

容量：上限 10000 订单

访问频率：

创建：每天 200 次

读取：每天 2000 次

更新：每天 2000 次

删除：每天 200 次

Order 类安全性：

订单信息只有自己能更改，其他人不能更改但能查看。

4.9 平台总的类分析

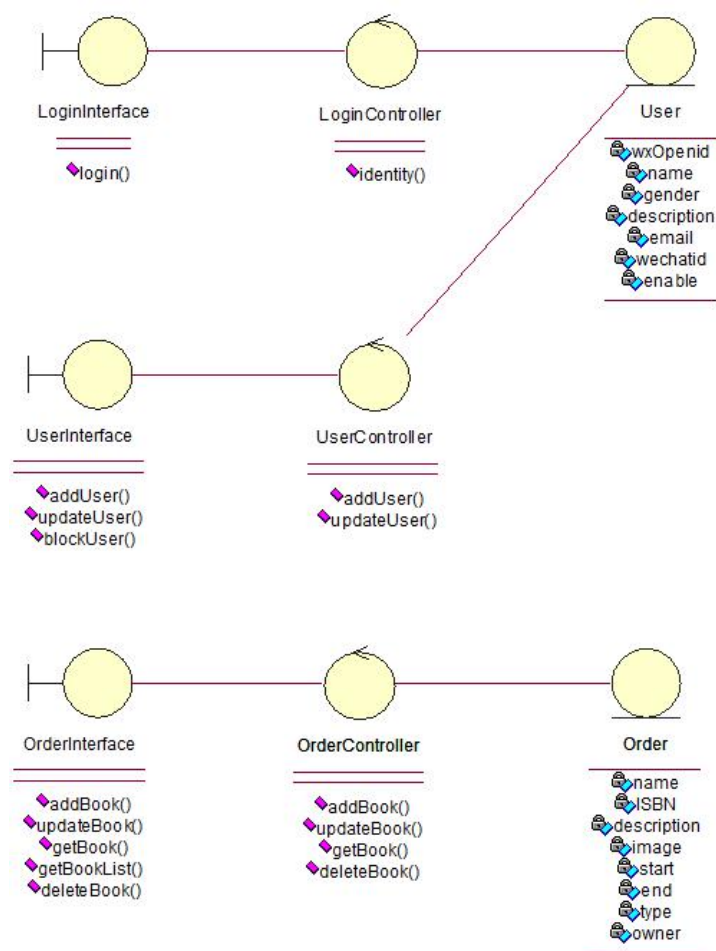


图 16 平台总的类分析

五、类的设计

5.1 实体类设计

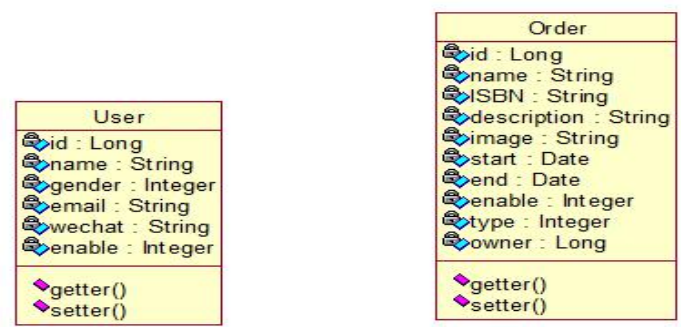


图 17 实体类设计。

六、子系统及其接口设计

本系统在系统框架和系统关键抽象的基础上，将系统划分成两个逻辑上相互独立，功能上相互依赖的子系统（模块），并对子系统进行分析和设计。

6.1 子系统划分

本系统在系统框架和系统关键抽象的基础上，将系统划分成两个逻辑上相互独立，功能上相互依赖的子系统（模块），分别为用户管理子系统，图书管理子系统。每个子系统（模块）包含了视图层，控制层，服务层以及模型层的类，如：用户管理子系统中，包括了表示层的用户注册和登录的视图界面以及个人信息的界面，控制层和服务层的校验类，模型层的用户类等等。子系统设计将对用户管理子系统以及图书管理子系统进行详细介绍。

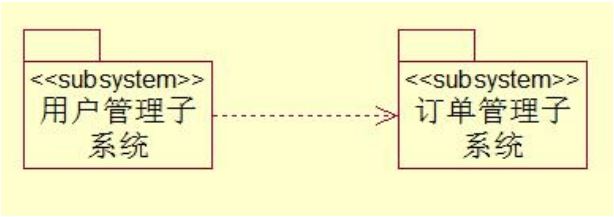


图 18 子系统的相互依赖关系。

6.2 子系统设计

本小节以用户管理子系统和图书管理子系统为例，介绍子系统设计，包括子系统接口，子系统的内部模块划分，各种类以及类的依赖关系。

6.2.1 用户管理子系统

用户管理子系统实现了用户实体的业务逻辑功能，如：用户注册，用户登录和管理个人信息功能。

依据系统架构以及用户实体的业务逻辑功能，可将用户管理子系统分成四个子模块，包括：用户表单，用户业务逻辑，用户服务逻辑以及用户实体，分别对应于系统框架的表示层，控制层以及模型层，下面对相应的子模块进行说明。

1) 用户表单子模块

对应于系统框架的表示层，该模块主要包含了用户表示层的各种表单，如：用户登录表单，用户注册表单和管理个人信息表单等。

2) 用户业务逻辑子模块

对应于系统框架的控制层，该模块主要包括处理用户行为的各种业务逻辑类，如：注册，登录等功能。

3) 用户服务逻辑子模块

对应系统框架的服务层，主要提供向模型层查询，更新用户信息的服务，并对信息的合法性进行检查和确认。

4) 用户实体

对应于系统框架的模型层， 该模块包含了用户实体类，实现数据的持久化。

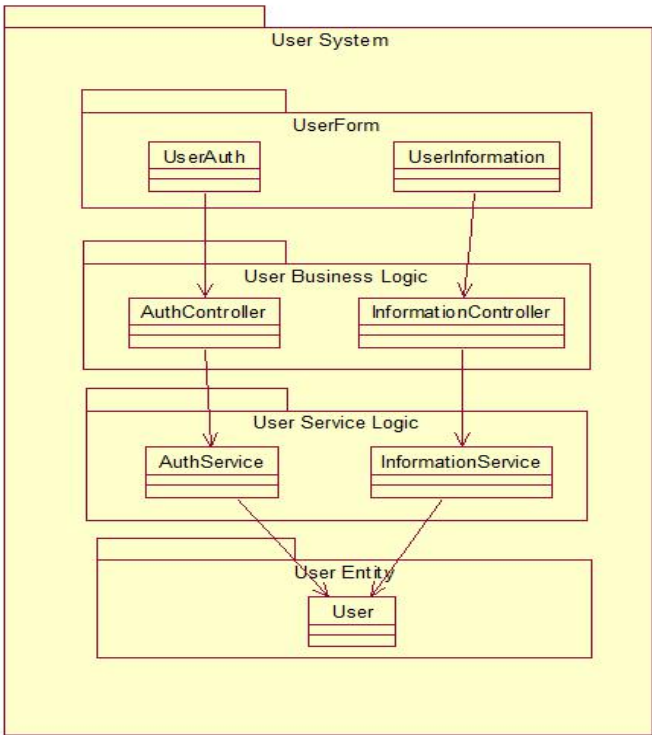


图 19 用户管理子系统的接口类图。

6.2.2 图书管理子系统

图书管理子系统实现了图书实体的业务逻辑功能，如：发布借出/求借图书信息等功能。

依据系统架构以及图书实体的业务逻辑功能，可将图书管理子系统分成四个子模块，包括：图书管理表单模块、图书管理业务逻辑模块、图书管理服务模块、图书管理实体模块，分别对应于系统框架的表示层，控制层，服务层以及实体层。

- 1) 图书管理表单子模块
对应于系统框架的表示层，该模块主要包含了图书表示层的各种表单，如：查看书目列表表单、预定书目表单以及退书表单等。
- 2) 图书管理业务逻辑子模块
对应于系统框架的控制层，该模块主要关于用户实体的各种业务逻辑类，如：预定书目等功能。
- 3) 图书管理服务子模块
对应系统框架的服务层，主要提供向模型层查询，更新图书订单信息的服务，并对信息的合法性进行检查和确认。
- 4) 图书管理实体模块
对应于系统框架的模型层， 该模块包含了图书实体类，实现数据的持久化。

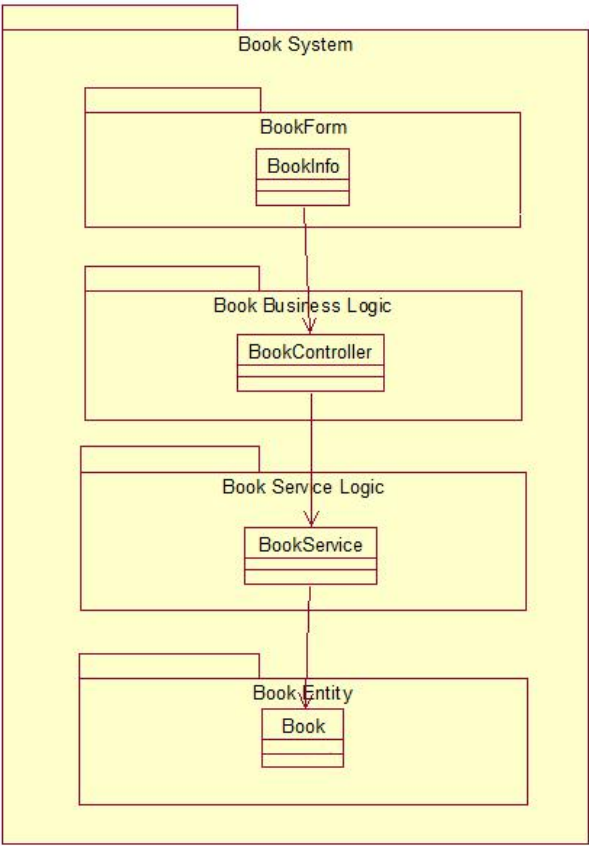


图 20 图书管理子系统的接口类图。

七、中大图书平台系统部件设计

7.1 中大图书平台系统运行时进程分析

本系统设置模块有两个功能：图书管理模块和用户管理模块。如图 20 所示，系统包括 MainProcess 主进程，authApplication 进程，authManagement 进程，userApplication 进程，userManagement 进程，bookApplication 进程，bookManagement 进程。

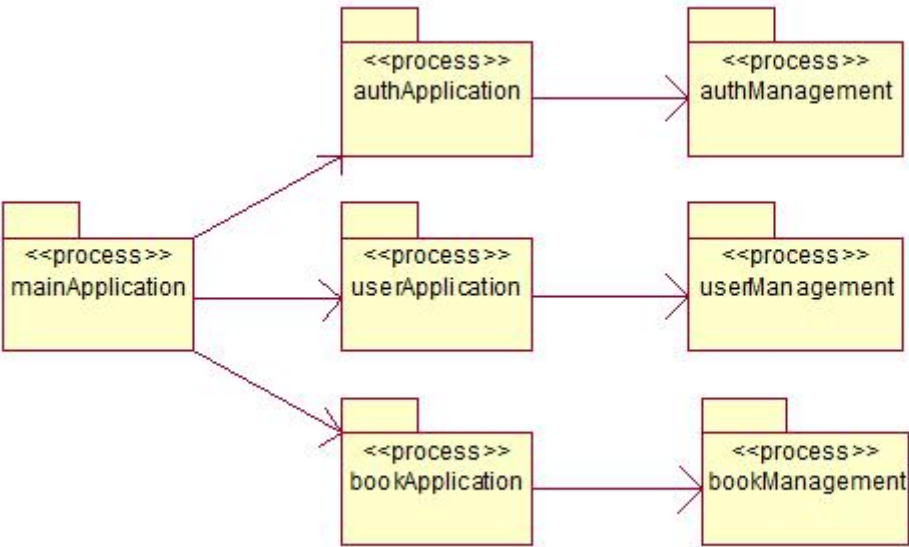


图 21 中大图书平台系统运行时架构示意图。