武汉纺织大学

Web应用开发课程设计

**图书馆管理系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11802**

**姓 名： 左四海**

**学 号： 1804241027**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月17日**

目录

[一·、课程设计题目 1](#_Toc58953617)

[二、课程设计目的 1](#_Toc58953618)

[三、需求概述 1](#_Toc58953619)

[四、设计准备 1](#_Toc58953620)

[系统准备 1](#_Toc58953621)

[知识准备 1](#_Toc58953622)

[迭代式开发 2](#_Toc58953623)

[技术栈 2](#_Toc58953624)

[五、系统分析与设计 3](#_Toc58953625)

[系统功能分析 3](#_Toc58953626)

[数据库分析与设计 3](#_Toc58953627)

[概念模型：ER图 4](#_Toc58953628)

[数据模型：表结构 4](#_Toc58953629)

[三层架构 8](#_Toc58953630)

[基于B/S的三层mvc架构 8](#_Toc58953631)

[类与页面的设计 9](#_Toc58953632)

[前端设计 9](#_Toc58953633)

[后端设计 10](#_Toc58953634)

[六、系统实现 16](#_Toc58953635)

[数据库实现（sql）脚本 16](#_Toc58953636)

[程序界面 18](#_Toc58953637)

[登录界面 18](#_Toc58953638)

[管理员页面 19](#_Toc58953639)

[用户界面 23](#_Toc58953640)

[七、总结 26](#_Toc58953641)

[课程设计小结 26](#_Toc58953642)

# 一·、课程设计题目

图书管理系统

# 二、课程设计目的

通过对图书管理系统的系统分析、系统设计、编码和调试等工作的实践，熟悉管理信息系统的开发过程、设计方法及编程技术，熟悉掌握数据库设计的基本理论及方法。

# 三、需求概述

图书管理系统的基本功能需求包括：读者管理、图书管理、借阅管理、用户登录与用户管理等；其它功能需求包括：读者查询与预约借书\*、统计与报表\*、数据备份\*、书架管理\*、期刊管理\*、期刊借阅\*、论文管理\*、论文借阅\*等。（\* 表示留待以后的迭代周期完成，下同）

# 四、设计准备

## 系统准备

操作系统：Windows 10

数据库系统：mysql

客户端开发工具：IDEA，Navicat

## 知识准备

熟悉mysql的使用；

熟悉Java语言及其数据库编程技术。

了解：迭代式开发过程、UML设计文档、设计模式；以及图书馆相关业务知识。

## 迭代式开发

迭代式开发（统一过程UP） 系统开发被组织成一系列固定的短期（一段为2-6周）小项目，称为迭代；每次迭代都产生可执行的系统。每次迭代都包括计划、需求、分析、设计、编码、测试等过程以及文档编写工作；第一次迭代考虑系统的核心功能，随后的迭代逐步扩展系统功能；每次迭代的成果（含需求、分析、设计、代码和文档等）均为下一次迭代的工作基础，直至满足最终需求。这种开发过程是基于面向对象方法的。

## **技术栈**

前端： Thymeleaf、Layui、Ajax、JQuery

后端 ： springboot , mybatis

开发环境： IDEA 、SpringBoot 2.3、Maven

数据库：MySQL 5.7

# 五、系统分析与设计

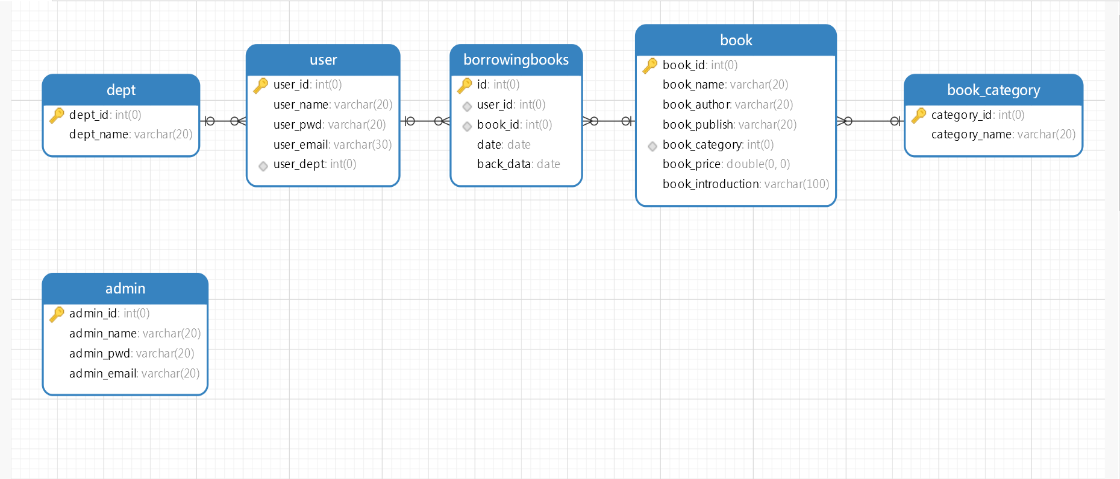
## 系统功能分析

## 数据库分析与设计

### 概念模型：ER图

### 数据模型：表结构

表模型



#### 表格

|  |  |  |  |
| --- | --- | --- | --- |
| Admin表（管理员表） | | | |
| 字段序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | admin\_id | Int | 管理员id【主键递增】 |
| 2 | admin\_name | Varchar(20) | 管理员名称，非空 |
| 3 | admin\_pwd | Varchar(20) | 管理员密码，非空 |
| 4 | admin\_email | Varchar(20) | 管理员邮箱，默认为null |

|  |  |  |  |
| --- | --- | --- | --- |
| User表（用户表） | | | |
| 字段序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | user\_id | int | 用户id（主键递增） |
| 2 | user\_name | varchar(20) | 用户名称，非空 |
| 3 | user\_pwd | varchar(20) | 用户密码，非空 |
| 4 | user\_email | varchar（30） | 用户邮箱，默认为null |
| 5 | user\_dept | int | 用户所在院系（外键） |

|  |  |  |  |
| --- | --- | --- | --- |
| book表（书籍表） | | | |
| 字段序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | book\_id | int | 书籍id（主键递增） |
| 2 | book\_name | varchar（20） | 书籍名称，非空 |
| 3 | book\_author | varchar(20) | 书籍作者 |
| 4 | book\_public | varchar | 出版社 |
| 5 | book\_category | int | 书籍类别，建立非空索引 |
| 6 | book\_price | int | 书本价格 |
| 7 | book\_introduction | varchar（100） | 书本描述 |

|  |  |  |  |
| --- | --- | --- | --- |
| book\_category(书籍类别) | | | |
| 字段序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | category\_id | int | 类别id（主键递增） |
| 2 | category\_name | varchar(20) | 类别名称 |

|  |  |  |  |
| --- | --- | --- | --- |
| borrowingbooks（借阅记录表） | | | |
| 字段序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | id | int | 借阅记录编号（主键） |
| 2 | user\_id | int | 记录所属用户（外键） |
| 3 | book\_id | int | 记录对应书籍（外键） |
| 4 | data | data | 借书时间 |
| 5 | back\_data | data | 还书时间 |

|  |  |  |  |
| --- | --- | --- | --- |
| dept（学生院系表） | | | |
| 字段序号 | 字段名称 | 字段类型 | 字段说明 |
| 1 | dept\_id | int | 院系id（主键） |
| 2 | dept\_name | varchar(20) | 院系名字 |

#### 具体实现

数据库具体实现

1. 创建数据库

CREATE DATABASE IF NOT EXISTS library-manager-system default charset utf8mb4 COLLATE utf8mb4\_general\_ci;

1. 创建数据模型对应数据表

Dept表

CREATE TABLE `dept` (

`dept\_id` int NOT NULL AUTO\_INCREMENT,

`dept\_name` varchar(20) DEFAULT NULL,

PRIMARY KEY (`dept\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=5 DEFAULT CHARSET=utf8;

Admin表

CREATE TABLE `admin` (

`admin\_id` int NOT NULL AUTO\_INCREMENT,

`admin\_name` varchar(20) DEFAULT NULL,

`admin\_pwd` varchar(20) DEFAULT NULL,

`admin\_email` varchar(20) DEFAULT NULL,

PRIMARY KEY (`admin\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=utf8;

book\_category表

CREATE TABLE `book\_category` (

`category\_id` int NOT NULL AUTO\_INCREMENT,

`category\_name` varchar(20) DEFAULT NULL,

PRIMARY KEY (`category\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=50 DEFAULT CHARSET=utf8;

User表

CREATE TABLE `user` (

`user\_id` int NOT NULL AUTO\_INCREMENT,

`user\_name` varchar(20) DEFAULT NULL,

`user\_pwd` varchar(20) DEFAULT NULL,

`user\_email` varchar(30) DEFAULT NULL,

`user\_dept` int DEFAULT NULL,

PRIMARY KEY (`user\_id`),

KEY `user\_dept\_key` (`user\_dept`),

CONSTRAINT `user\_dept\_key` FOREIGN KEY (`user\_dept`) REFERENCES `dept` (`dept\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=27 DEFAULT CHARSET=utf8;

Book表

CREATE TABLE `book` (

`book\_id` int NOT NULL AUTO\_INCREMENT,

`book\_name` varchar(20) NOT NULL,

`book\_author` varchar(20) DEFAULT NULL,

`book\_publish` varchar(20) DEFAULT NULL,

`book\_category` int DEFAULT NULL,

`book\_price` double DEFAULT NULL,

`book\_introduction` varchar(100) DEFAULT NULL,

PRIMARY KEY (`book\_id`),

KEY `book\_category` (`book\_category`) USING BTREE,

CONSTRAINT `book\_ibfk\_1` FOREIGN KEY (`book\_category`) REFERENCES `book\_category` (`category\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=74 DEFAULT CHARSET=utf8;

borrowingbooks表

CREATE TABLE `borrowingbooks` (

`id` int NOT NULL AUTO\_INCREMENT,

`user\_id` int DEFAULT NULL,

`book\_id` int DEFAULT NULL,

`date` date DEFAULT NULL,

`back\_data` date DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `user\_id` (`user\_id`) USING BTREE,

KEY `book\_id` (`book\_id`) USING BTREE,

CONSTRAINT `borrowingbooks\_ibfk\_1` FOREIGN KEY (`book\_id`) REFERENCES `book` (`book\_id`),

CONSTRAINT `borrowingbooks\_ibfk\_2` FOREIGN KEY (`user\_id`) REFERENCES `user` (`user\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=67 DEFAULT CHARSET=utf8;

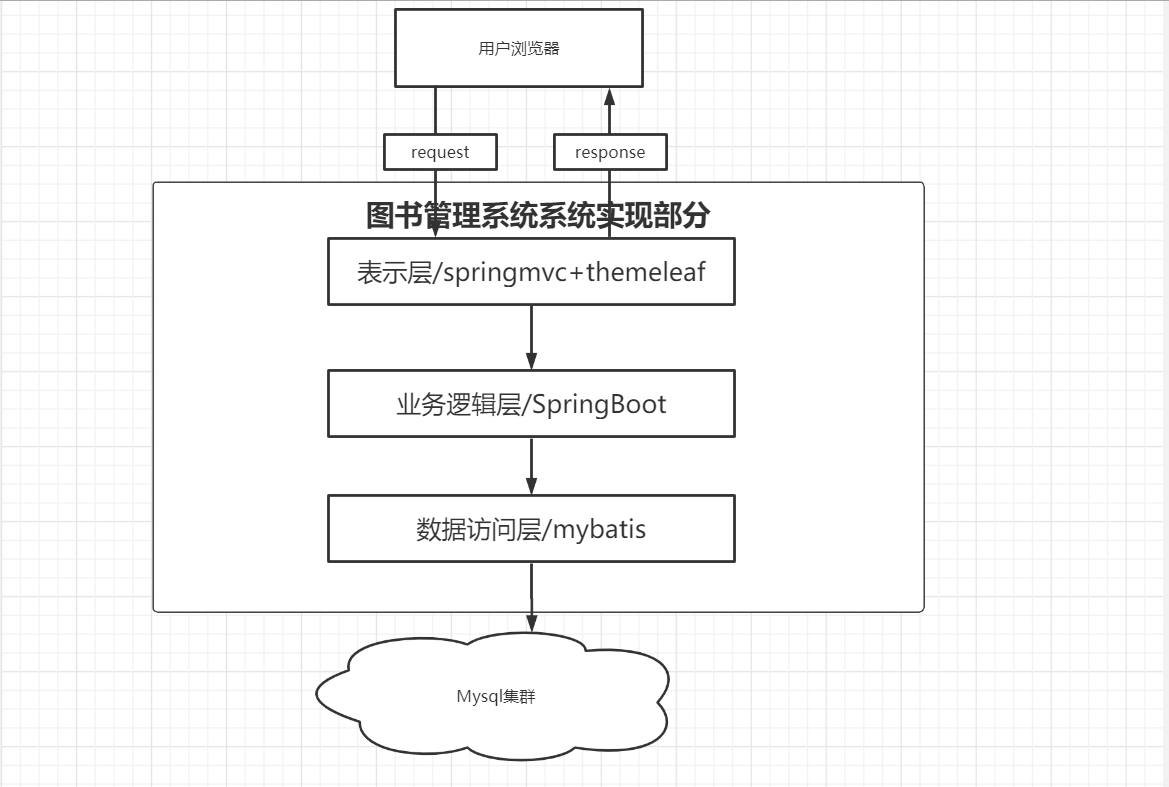
## 三层架构

### 基于B/S的三层mvc架构

三层架构是指：视图层View、服务层Service、持久层Dao，分别完成不同的功能。

View层：用于接收用户提交请求的代码在这里编写。

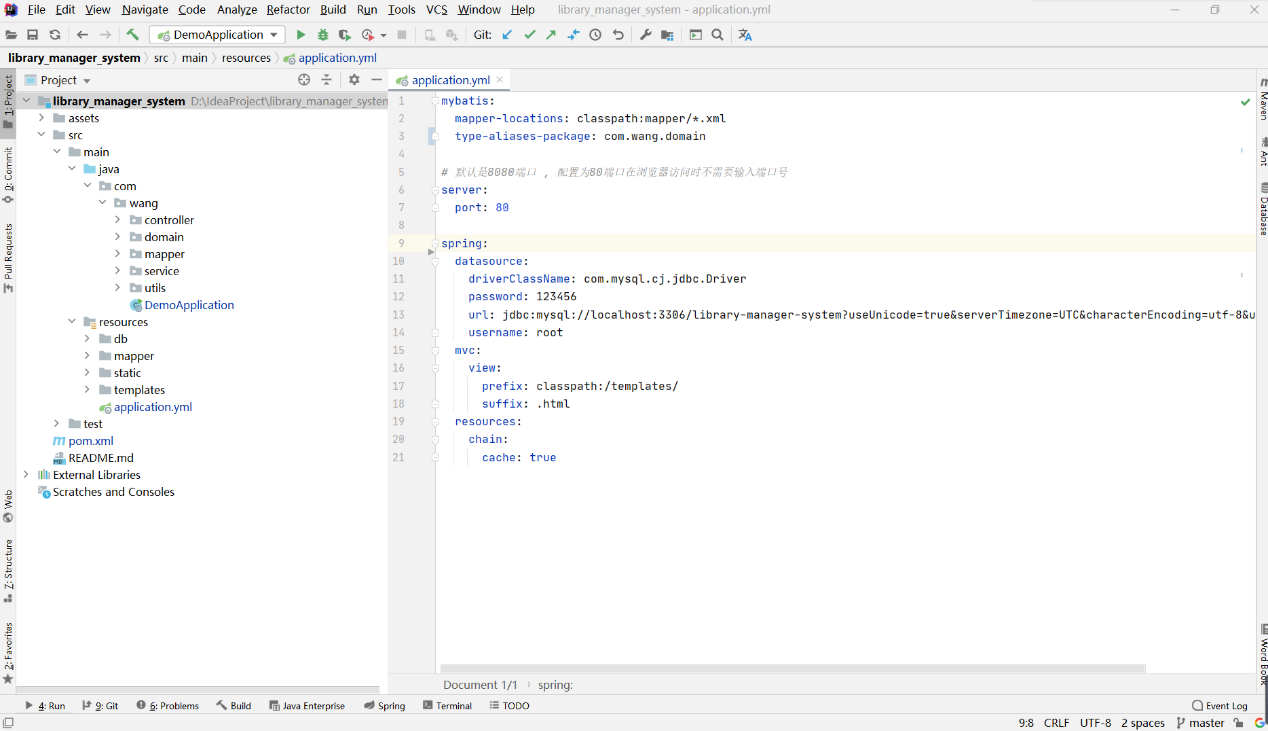
Service层：系统的业务逻辑主要在这里编写。

Dao层：直接操作数据库的代码在这里编写。

为了更好的降低各层间的耦合度，在三层架构程序设计中，采用面向抽象编程。即上层对下层的调用，是通过接口实现的。而下层对上层的真正服务提供者，是下层接口的实现类。服务标准（接口）是相同的，服务提供者（实现类）可以更换。这就实现了层间的耦合。

搭建此项目的mvc三层架构

如图，在IDEA中创建maven项目目录结构



其中每一个包对应三层架构中一层

Controller：View层，处理用户请求

Domain: model层 实体类模块

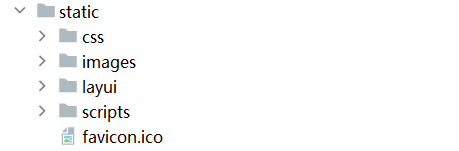
Mapper：DAL数据访问层

Service：BBL业务逻辑层

## 类与页面的设计

前端设计：

#### 静态资源

静态资源都放在static包下：如图片，样式，图标等

#### 页面资源

Admin：存放管理员页面

User：存放用户页面

Commom：存放公共页面资源



### 后端设计

#### 视图层

|  |  |  |
| --- | --- | --- |
| 表示层  VIEW | AdminController | 处理管理员相关请求 |
| BookController | 处理书籍相关请求 |
| BorrowingController | 处理借阅相关请求 |
| UserController | 处理用户相关请求 |

#### 业务逻辑层

#### Model层

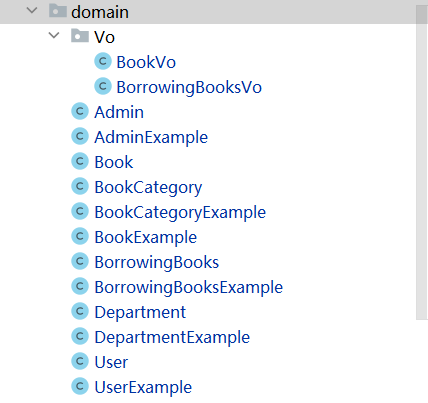
Model层设计主要依赖于orm思想

ORM Object Relational Mapping 对象关系映射，是为了解决面向对象的类，与关系数据库的表之间，存在的不匹配的现象，通过使用描述对象和关系之间映射的元数据，在程序中的类对象，与关系数据库的表之间建立持久的关系，用于在程序中描述数据库表。本质上就是将数据从一种形式转换到另外一种形式。

其中包括对象实体类和传输对象类还有条件对象

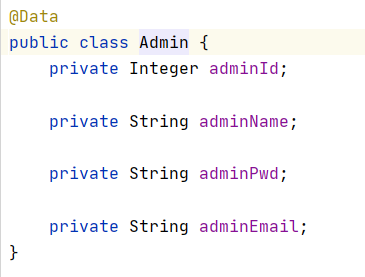
实体对象：与数据库表先对应的类

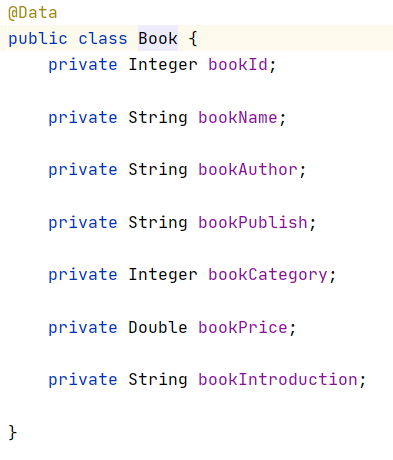
传输对象：用于在mvc三层间数据的传输

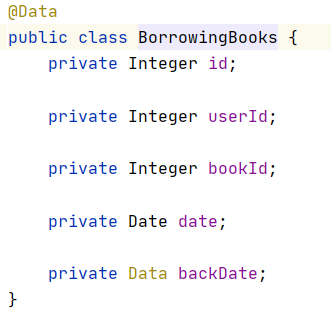
条件对象：通过对实体对象进行条件的包装，使得更灵活的进行数据库的增删改查操作，后缀example，如

以下各实体类对应表

Admin表



Book表

Brrowing表

其中核心就是sql字段类型和java字段类型之间的转换

同理，其他数据相关表均可知道。此处略。

DAL层设计：（也称dao层或者mybats中的mapper层）

DAL层的类主要实现对应数据表的插、删、改、查等操作，以及存储过程的调用系统实现

其中每一个mapper接口都对应一个实现他的mybatis映射文件

AdminMapper

能过通过条件对象对数据表进行各种参数的增删改查（包括用户和书籍以及借阅情况）



BookMapper：对书籍的增删改查操作

BorrowingBooksMapper：对借阅记录的增删改查的各种操作

userMapper：拥有用户权限能进行的各种操作。

其他的mapper类（DAO类）也相似。

# 六、系统实现

## 数据库实现（sql）脚本

SET FOREIGN\_KEY\_CHECKS=0;

-- ----------------------------

-- Table structure for ms\_admin

-- ----------------------------

DROP TABLE IF EXISTS `ms\_admin`;

CREATE TABLE `ms\_admin` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`admin\_number` varchar(11) NOT NULL COMMENT '登录号',

`admin\_pwd` varchar(255) DEFAULT NULL COMMENT '登录密码',

`admin\_name` varchar(255) DEFAULT NULL COMMENT '名称',

`login\_pre\_time` date DEFAULT NULL COMMENT '上次登录时间',

`del\_flg` int(1) DEFAULT NULL COMMENT '是否删除(标记用)',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of ms\_admin

-- ----------------------------

INSERT INTO `ms\_admin` VALUES ('1', '1101', '123456', 'Bob', '2019-04-12', '1');

INSERT INTO `ms\_admin` VALUES ('2', '1102', '123456', '大天狗', '2019-04-09', '1');

INSERT INTO `ms\_admin` VALUES ('3', '1103', '123456', '酒吞童子', '2019-01-09', '1');

-- ----------------------------

-- Table structure for ms\_book

-- ----------------------------

DROP TABLE IF EXISTS `ms\_book`;

CREATE TABLE `ms\_book` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`title` varchar(255) DEFAULT NULL COMMENT '书名',

`ISBN` varchar(255) DEFAULT NULL COMMENT '统一使用07年新颁布的13位数字',

`author` varchar(255) DEFAULT NULL COMMENT '作者',

`introduction` varchar(255) DEFAULT NULL COMMENT '简介',

`price` varchar(255) DEFAULT NULL COMMENT '价格',

`publish\_time` varchar(255) DEFAULT NULL COMMENT '出版时间',

`category\_id` int(10) DEFAULT NULL COMMENT '类别',

`image` varchar(255) DEFAULT NULL COMMENT '图片url',

`create\_time` date DEFAULT NULL COMMENT '上架时间',

`create\_admin` varchar(255) DEFAULT NULL COMMENT '上架管理员',

`update\_pre\_admin` varchar(255) DEFAULT NULL COMMENT '上一次修改信息的管理员',

`del\_flg` int(1) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=344 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of ms\_book

-- ----------------------------

INSERT INTO `ms\_book` VALUES ('1', '活着', '9787506365437', '余华', '《活着》是作家余华的代表作之一，讲述了在大时代背景下，随着内战、三反五反，大跃进，文化大革命等社会变革，徐福贵的人生和家庭不断经受着苦难，到了最后所有亲人都先后离他而去，仅剩下年老的他和一头老牛相依为命。', '25.30', '1923年6月', '1', '', '2019-04-13', 'Bob', '大天狗', '1');

-- ----------------------------

-- Table structure for ms\_category

-- ----------------------------

DROP TABLE IF EXISTS `ms\_category`;

CREATE TABLE `ms\_category` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`category\_id` int(11) DEFAULT NULL COMMENT '类别号',

`category\_name` varchar(255) DEFAULT NULL COMMENT '类别名称',

`del\_flg` int(1) DEFAULT NULL COMMENT '0表示已删除，1表示未删除',

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=10 DEFAULT CHARSET=utf8;

-- ----------------------------

-- Records of ms\_category

-- ----------------------------

INSERT INTO `ms\_category` VALUES ('1', '1', '小说', '1');

INSERT INTO `ms\_category` VALUES ('2', '2', '文学', '1');

INSERT INTO `ms\_category` VALUES ('3', '3', '动漫', '1');

INSERT INTO `ms\_category` VALUES ('4', '4', '文化', '1');

INSERT INTO `ms\_category` VALUES ('5', '5', '传记', '1');

INSERT INTO `ms\_category` VALUES ('6', '6', '艺术', '1');

INSERT INTO `ms\_category` VALUES ('7', '7', '童书', '1');

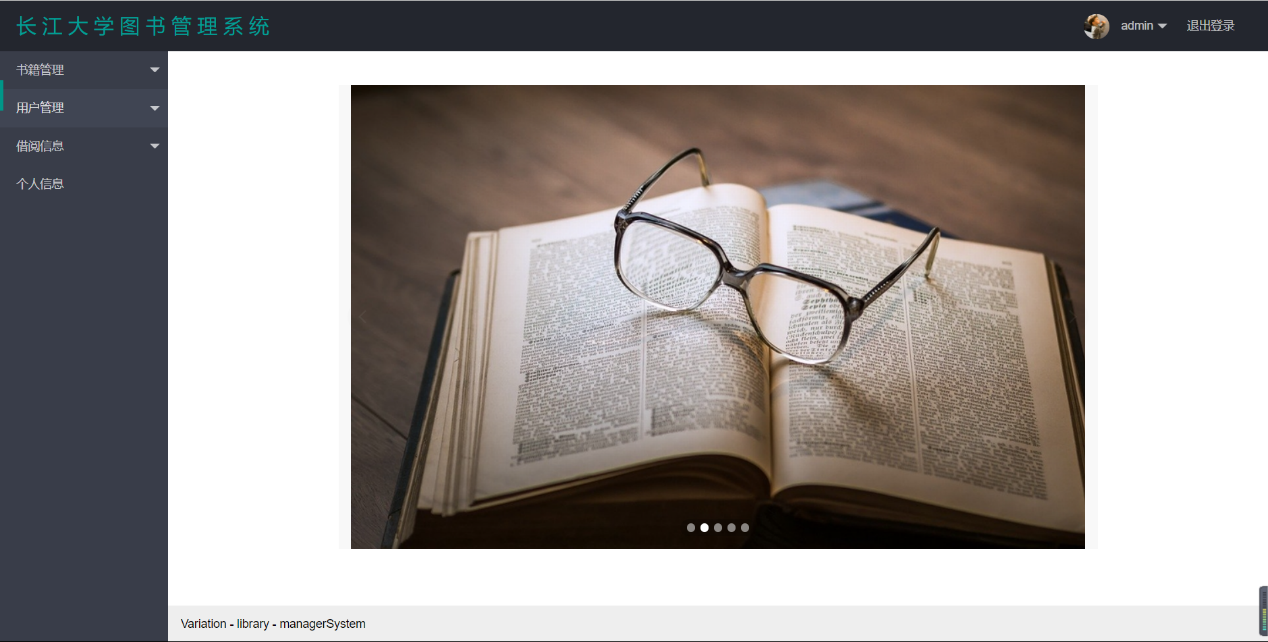
INSERT INTO `ms\_category` VALUES ('8', '8', '古籍', '1');

INSERT INTO `ms\_category` VALUES ('9', '9', '历史', '1');

## 程序界面

### 登录界面

### 管理员页面

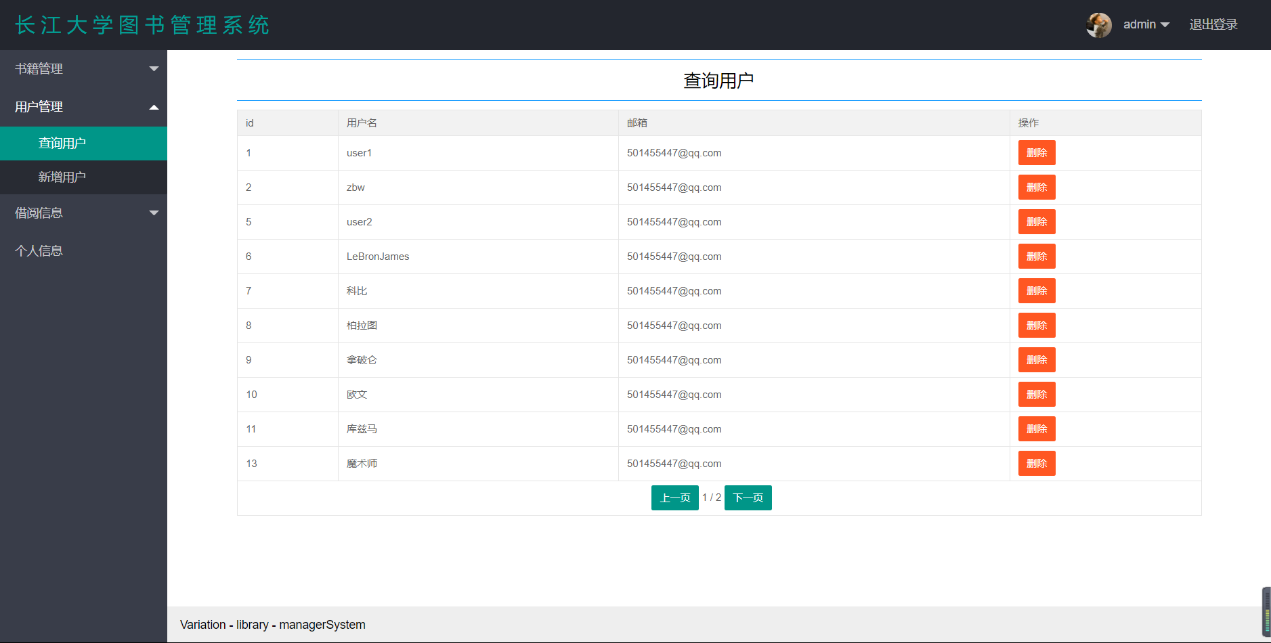
管理员首页

#### 添加书籍页面

#### 新建书籍类别页面

#### 通过类别查找书籍页面

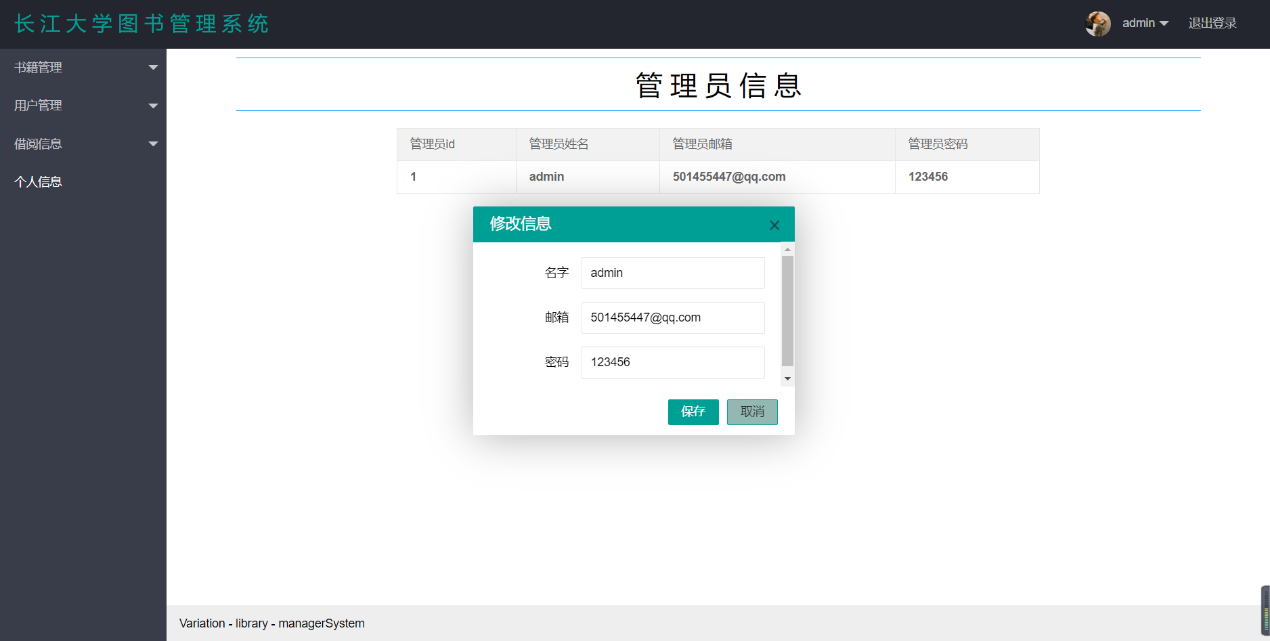
#### 查询用户页面



#### 新增用户界面

#### 借阅记录页面

#### 修改信息页面



### 用户界面

#### 用户登录首页

#### 用户书籍借阅记录页面

#### 用户借书页面

#### 用户还书页面

#### 查询书籍页面

#### 个人信息以及修改页面

# 七、总结

## 课程设计小结

数据库课程设计，不能急于开始编码。我在本次的数据库课程设计中，一开始并没有认 真进行需求分析以及数据库的设计，而是急于去编码实现界面等前台的东西。结果编到第二 个界面的时候，发现数据库设计不合理，只好重新回去修改。我的解决方法是：详细进行需 求分析，争取做到对整个系统了然于胸；详细进行系统功能划分，分到最小模块为止，然后 再一个一个模块添加代码即可，当然，要考虑模块与模块之间的耦合程度以及模块与模块之 间的信息共享。 在本次数据库课程设计中遇到不少困难，例如：在编写学生借用图书的模块的时候，由 于原来没有用过存储过程，在后台编写以及前台调用的时候都没有头绪；在编写图书信息管 理模块时，dataview 的更新总也实现不了等等诸如此类的困难很多。在这时，我一般采取 以下措施：上网查资料；在网上论坛上提问题；和同学讨论；在图书馆查找相关图书资料。 总之，这些是我比较常用的方法。最重要的一点是，遇到困难时的心态要平和，冷静地去查 找问题并解决问题。