设计文档

孔令仁2018201661

# 大作业设计

[大作业设计](#header-n0)  
 [前言](#header-n20)  
 [业务描述](#header-n24)  
 [数据库设计（ER图）](#header-n41)  
 [ER图](#header-n42)  
 [解释](#header-n45)  
 [转换为关系模型](#header-n67)  
 [应用系统模块设计、功能点描述](#header-n351)  
 [读者功能-登录](#header-n357)  
 [读者功能-退出登录](#header-n367)  
 [读者功能-修改密码](#header-n370)  
 [读者功能-校外注册](#header-n380)  
 [读者功能-图书信息查询](#header-n385)  
 [读者功能-图书借阅](#header-n389)  
 [读者功能-图书归还&延期](#header-n392)  
 [管理功能-管理员登录与登出](#header-n397)  
 [管理功能-用户管理](#header-n408)  
 [管理功能-图书管理](#header-n418)  
 [管理功能-超级管理员](#header-n423)  
 [管理功能-数据展示](#header-n427)  
 [系统描述实现](#header-n430)  
 [python环境](#header-n431)  
 [数据库配置](#header-n443)  
 [网页模板设计](#header-n451)  
 [初始数据](#header-n477)  
 [用户端功能](#header-n511)  
 [登录](#header-n512)  
 [登出](#header-n521)  
 [注册](#header-n524)  
 [图书信息查询](#header-n533)  
 [图书借阅](#header-n539)  
 [图书归还与续借](#header-n545)  
 [管理端基础](#header-n549)  
 [用户管理](#header-n552)  
 [被禁用的账号](#header-n555)  
 [逾期不还书封号](#header-n558)  
 [固定参数统计图](#header-n562)  
 [图书管理](#header-n564)  
 [已知图书管理](#header-n565)  
 [图书入库](#header-n571)  
 [超级管理](#header-n577)  
 [数据展示](#header-n580)  
 [总结](#header-n585)  
 [参考文献和资料](#header-n632)

## 前言

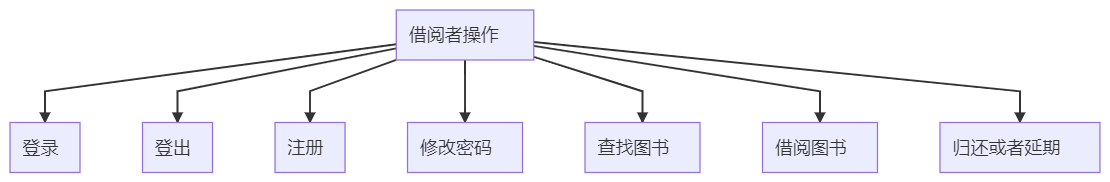
本作业旨在完成一个图书管理系统，从而实践学习到的数据库知识。

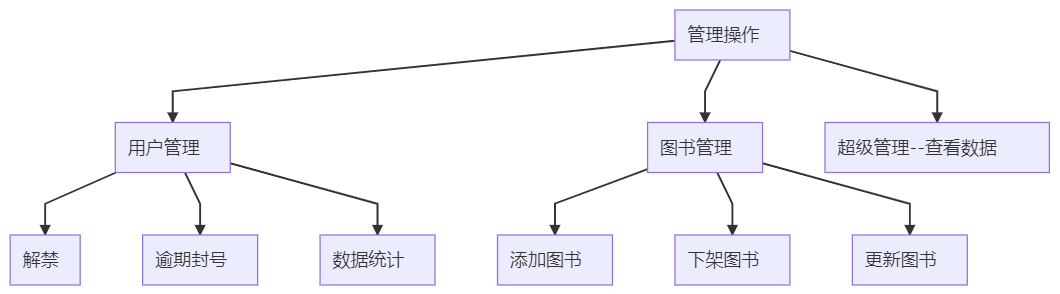
图书管理系统的模式参考学校的图书馆借阅流程

## 业务描述

业务目标是实现一个学校的图书馆管理系统，作为学校的图书馆管理系统，包含两个部分：

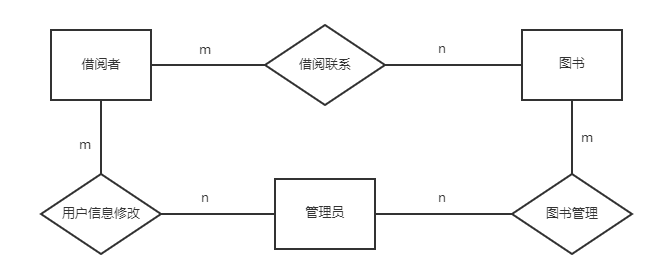
* 学生/老师：作为借阅者，可以登录、操作（借阅还书延期等）、退出登录
* 管理者：
  + 用户管理：通过管理系统，管理借阅者的情况，处理用户的借阅问题
  + 图书馆员工：通过管理系统，添加、修改、下架图书
  + 超级管理：查看其它管理者操作记录





## 数据库设计（ER图）

### ER图



### 解释

包含3个实体和3个联系：

* 实体：
  + 借阅者：id（学生教师号）、姓名、密码、用户类型、在借书数目、备注、是否被禁用
  + 图书：id（书号）、书名、简介、类型、作者、出版社、价格、数量、备注、是否下架
  + 管理员：id（账号）、密码、权限类型、备注
* 联系：
  + 借阅联系：借出时间、借阅期限、归还时间、状态、备注
  + 用户信息修改：修改日期、修改类型、备注
  + 图书管理：操作日期、操作类型、操作数值、原始数值、最终数值、备注

### 转换为关系模型

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 借阅者UserList | 学生/教师号 | 姓名 | 密码 | 用户类型 | 在借书数量 | 备注 | 是否已经被禁用 |
| 字段名 | Uno | Uname | Upassword | Utype | Uget | more | Urestrict |
| 字段类型 | char | nvarchar | char | nvarchar | int | nvarchar | bit |
| 限制 | 主键，11位 | 非空，最长10 | 32 | 最长10 | 非负，非空 | 2000 | 非空 |
| 备注 | 如S/T/U2018201661 | 真名字 | 初始化学号；MD5加密 | 有限的几种 | 业务检查即可 | 初始为空 | 0代表否 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 图书Book | 书号 | 书名 | 简介 | 类型 | 作者 | 出版社 | 价格 | 剩余数量 | 备注 | 是否下架 |
| 字段名 | Bno | Bname | breif | Btype | author | Press | price | num | more | removed |
| 字段类型 | char | nvarcahr | nvarchar | nvarchar | nvarchar | nvarchar | numeric | 整数 | nvarchar | bit |
| 限制 | 主键，13位 | 100以内，非空 | 1000以内 | 最长20 | 最长100 | 最长100 | 两位小数 | 非负 | 2000以内 | 非空 |
| 备注 | ISBN号 | 含副标题 |  |  |  |  | 人民币 |  | 初始为空 | 0代表否 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 管理员AdminList | 账号 | 密码 | 权限类型 | 备注 |
| 字段名 | Ano | Apassword | Atype | more |
| 字段类型 | char | char | nvarchar | nvarcahr |
| 限制 | 主键，20以内 | 32 | 最长10 | 2000以内 |
| 备注 |  | MD5加密 | 不同权限不同效果 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 借阅联系Borrow | 借书人 | 被借书 | 借出时间 | 借阅期限 | 归还时间 | 状态 | 备注 |
| 字段名 | Uno | Bno | outDate | maxtime | returnDate | isReturn | more |
| 字段类型 | char | char | date | int | date | bit | nvarchar |
| 限制 | 11 | 13 |  | 非负 | 归还晚于借出 | 非空 | 2000 |
| 备注 | 外键 | 外键 |  |  |  | 是否归还 |  |

利用借书人+书+借出日期构成联合主键，因为同一个人可以借同一个书多次，但是一天内不能重复借（如果用时间戳更加严谨，但此处进行简化）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户信息修改UAlter | 用户 | 管理者 | 修改时间 | 修改类型 | 备注 |
| 字段名 | Uno | Ano | actDate | AlterType | more |
| 字段类型 | char | char | date | nvarchar | nvarchar |
| 限制 | 11 | 20 |  | 非空，20 | 2000 |
| 备注 | 外键 | 外键 |  |  |  |

用户信息修改类型有：创建、改密码、禁用、解禁

采用用户+管理者+修改时间作为联合主键

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 图书管理BAlter | 图书 | 管理者 | 操作时间 | 操作类型 | 操作数值 | 原始数值 | 最终数值 | 备注 |
| 字段名 | Bno | Ano | actDate | AlterType | act | old | new | more |
| 字段类型 | char | char | date | nvarchar | int | int | int | nvarchar |
| 限制 | 13 | 20 |  | 非空，20 |  |  |  | 2000 |
| 备注 | 外键 | 外键 |  |  |  |  |  |  |

图书的操作类型有：入库、损失、出库

采用图书+管理者+操作时间作为联合主键

## 应用系统模块设计、功能点描述

功能整体分为两块：读者的操作（借书还书等）、管理员的操作（查看/修改-图书/借阅数据）

图例：

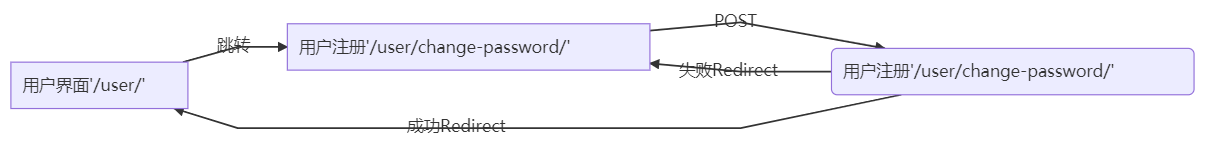
矩形代表实际网页

圆角代表处理相应信息的url

### 读者功能-登录

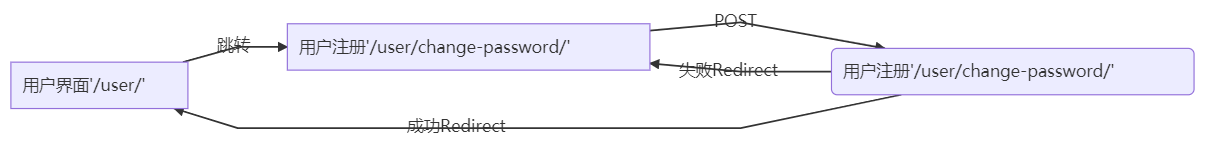
登录是读者端的基本功能，在用户的入口user网页中，有三种情况：

* 新用户进入网页：显示为一个登录输入账号密码的框，用来进行登录
* 登录成功以后：界面显示登录成功，且用户可以使用登录后的功能
* 登录有误：显示登录有误，并再次显示登录框



### 读者功能-退出登录

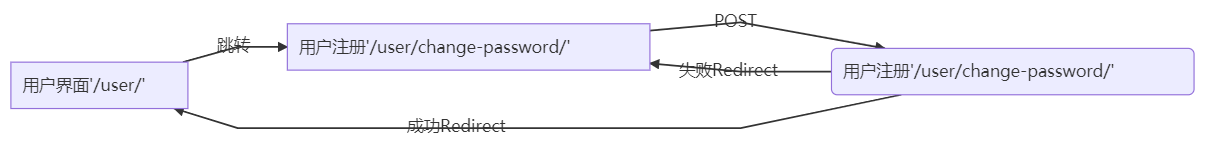
在登录后，点击退出登录按钮即可退出登录，重新回到初始登录界面



### 读者功能-修改密码

在登录之后，如果想修改密码，可以在修改密码的界面修改密码，会有几种情况：

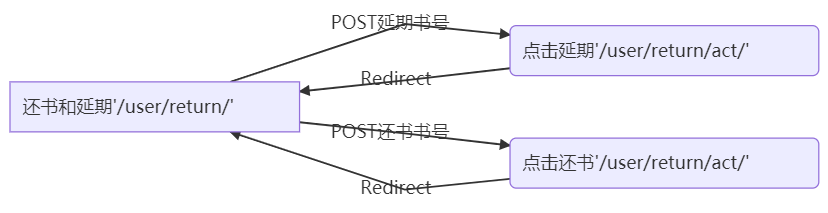
* 两次新密码不一致：页面内通过JS识别并提示，禁止提交修改
* 旧密码不正确：在判定之后返回修改页，提示密码错误
* 成功：返回user重新登录



### 读者功能-校外注册

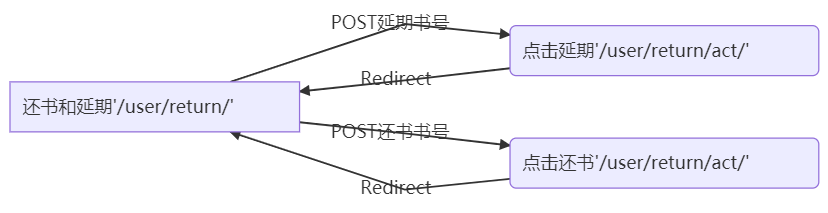
对于校外读者，可以进行注册，填写信息注册之后回到初始登录界面，然后就可以用注册的账号登录了；如果注册信息有问题则返回注册页面。

注：对于校外注册，初始权限设置为禁用，需要等待管理员授权才可以进行借阅操作。



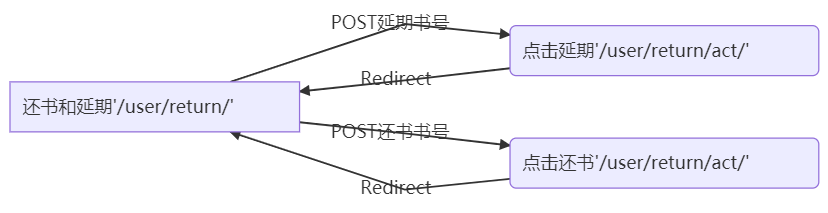
### 读者功能-图书信息查询

在这个界面，可以根据关键词查找匹配有关的书，按照5个一页的方式展示在页面中。 如果想要重新搜索，可以点击链接返回到搜索界面。



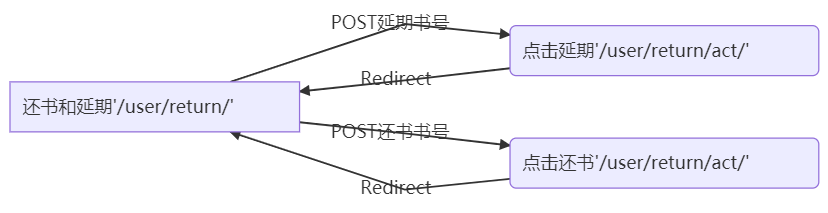
### 读者功能-图书借阅

在图书查找的过程中，可以通过点击借书按钮，把某书添加到借阅列表（这里的设计参照购物车的概念） ，而后可以继续查找借阅，或者点击确认借书。



### 读者功能-图书归还&延期

这个界面列出了用户所有的正在借阅图书，对于这些书，如果没有延期，则可以通过延期按钮续借7天，否则就只可以进行还书操作。



### 管理功能-管理员登录与登出

管理员方面的登录和读者的登录类似，在入口admin网页中，有三种情况：

* 初始进入网页：显示为一个登录输入账号密码的框，用来进行登录
* 登录成功以后：界面显示登录成功，且用户可以使用登录后的功能
* 登录有误：显示登录有误，并再次显示登录框

登出也是类似的原理



### 管理功能-用户管理

页面有三个内容：

* 显示被禁的账号情况，可以点击解禁
* 提供一个检测功能，点击按钮可以根据日期筛选逾期未还的账号，进行封号处理
* 提供一个当前账号的数量汇总图【完成老师对于表和图同一网页的加分要求】



### 管理功能-图书管理

图书管理页面展示了图书（在库和下架分别显示）。

对于在库图书提供下架按钮，对于下架图书提供上架（并输入数目）按钮

此外可以在子页面输入新书的全部信息来实现新书上架。



### 管理功能-超级管理员

提供了这样一个界面，对于具有超级管理员权限的用户，可以在这个页查看所有U/B两个实体表的修改记录。 这个页面支持账号权限的验证，同时可以向前向后翻页看，每页显示5条数据。



### 管理功能-数据展示

提供了一个界面，通过选择标签，可以查看该标签下的图书数量，以Echarts动态JS呈现出来



## 系统描述实现

### python环境

使用的是venv虚拟环境，python为3.7版本

pip freeze > requirements.txt

安装的依赖项使用这个方法导出到文件中，可以直接用指令安装

pip install -r requirements/requirements.txt

项目使用的库基本是常见的库，所以可以不安装虚拟环境（requirements里面有一些是和项目无关的库，比如jupyter）

需要用到的包：

* json：处理JS，从而将参数可以传递到模板的JS里面
* datetime：处理日期数据，从而检查图书借阅逾期的问题

### 数据库配置

CREATE TABLE UserList  
(  
 Uno char(11) PRIMARY KEY,  
 Uname nvarchar(10) NOT NULL,  
 Upassword char(32) NOT NULL,  
 Utype nvarchar(10) NOT NULL,  
 Uget int NOT NULL,  
 more nvarchar(2000),  
 Urestrict bit NOT NULL,  
)

CREATE TABLE Book(  
 Bno char(13) PRIMARY KEY,  
 Bname nvarchar(100) NOT NULL,  
 brief nvarchar(1000) NOT NULL,  
 Btype nvarchar(20),  
 author nvarchar(100) NOT NULL,  
 Press nvarchar(100) NOT NULL,  
 price numeric(5,2) NOT NULL,  
 num int,  
 more nvarchar(2000),  
 removed bit NOT NULL  
)

CREATE TABLE AdminList(  
 Ano char(20) PRIMARY KEY,  
 Apassword char(32) NOT NULL,  
 Atype nvarchar(10) NOT NULL,  
 more nvarchar(2000)  
)

CREATE TABLE Borrow(  
 Uno char(11),  
 Bno char(13),  
 outDate date NOT NULL,  
 maxtime int NOT NULL,  
 returnDate date,  
 isReturn bit NOT NULL,  
 more nvarchar(2000),  
 FOREIGN KEY (Uno) REFERENCES UserList(Uno)   
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (Bno) REFERENCES Book(Bno)   
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (Uno,Bno,outDate)  
)

CREATE TABLE UAlter(  
 Uno char(11),  
 Ano char(20),  
 actDate date NOT NULL,  
 AlterType nvarchar(20) NOT NULL,  
 more nvarchar(2000),  
 FOREIGN KEY (Uno) REFERENCES UserList(Uno)   
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (Ano) REFERENCES AdminList(Ano)   
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (Uno,Ano,actDate)  
)

CREATE TABLE BAlter(  
 Bno char(13),  
 Ano char(20),  
 actDate date NOT NULL,  
 AlterType nvarchar(20) NOT NULL,  
 act int,  
 old int,  
 new int,  
 more nvarchar(2000),  
 FOREIGN KEY (Bno) REFERENCES Book(Bno)   
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (Ano) REFERENCES AdminList(Ano)   
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (Bno,Ano,actDate)  
)

### 网页模板设计

在本地static文件夹下载了boostrap4的前端css和js文件，加载到模板中。

模板使用的是框架配套的jinja2，各个网页通过继承base.html文件来达成各自的效果：

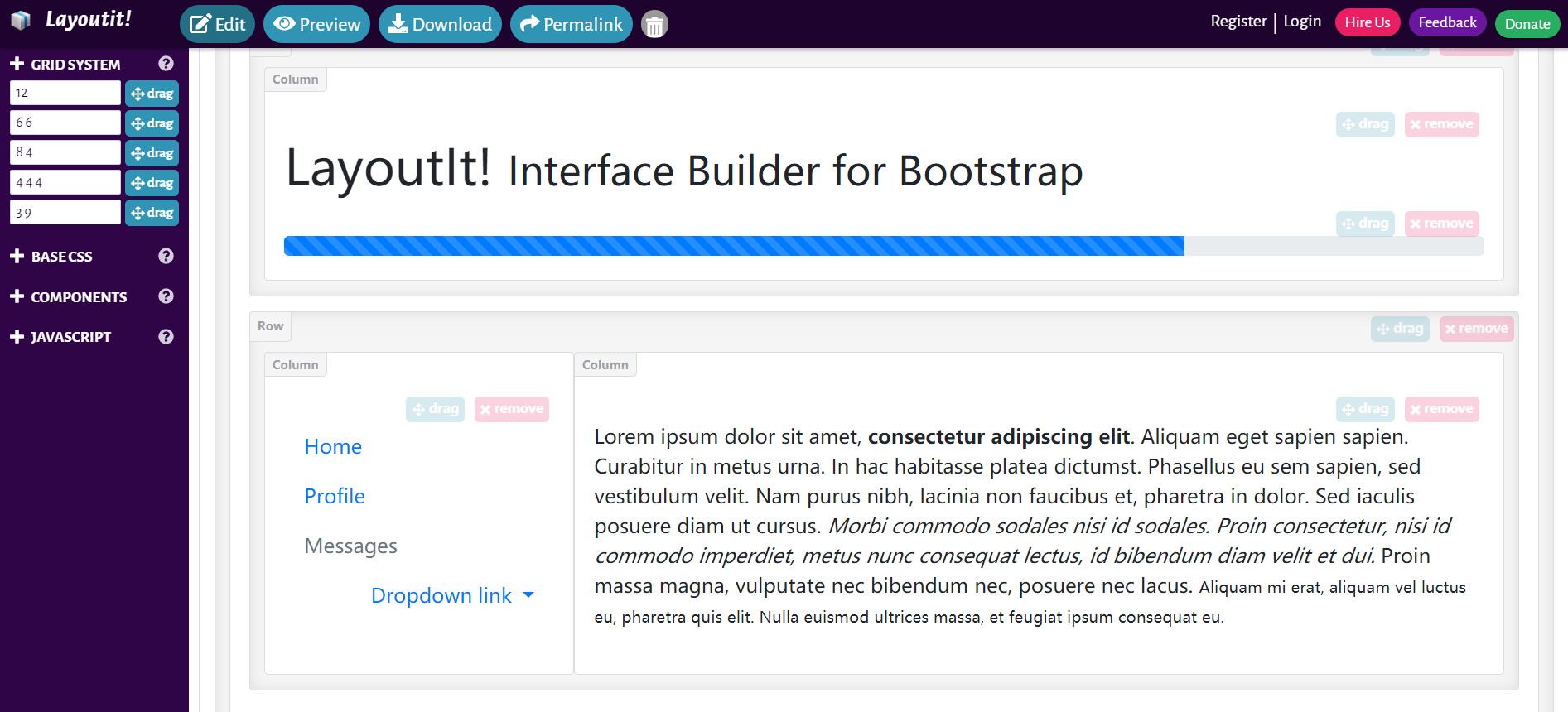
* base.html：基HTML文件，提供了对于整体网页架构的约束，含有几个块（block）
  + site\_name：网页名字（HTML-header部分），应该在模板中替换
  + header：网页的顶部标题
  + left\_body：左侧侧边栏，占据3/12的比例，用来放置跳转连接
  + right\_body：右侧主体，放网页的主体内容

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
 "http://www.w3.org/TR/html4/loose.dtd">  
<script src="https://how2j.cn/study/js/jquery/2.0.0/jquery.min.js"></script>  
<!--<link href="https://how2j.cn/study/css/bootstrap/3.3.6/bootstrap.min.css" rel="stylesheet">-->  
<!--<script src="https://how2j.cn/study/js/bootstrap/3.3.6/bootstrap.min.js"></script>-->  
<link href="{{ url\_for('static', filename = 'css/bootstrap.min.css') }}" rel="stylesheet">  
<script src="{{ url\_for('static', filename = 'js/bootstrap.min.js') }}"></script>  
  
<html>  
 <head>  
 <title>{% block site\_name %}网页名字，应该在模板中替换{% endblock %}</title>  
 </head>  
 <body>  
 <div class="container-fluid">  
  
 <div class="row">  
 <div class="col-md-12">  
 <h1 class="text-center">  
 {% block header %}顶部标题{% endblock %}  
 </h1>  
 </div>  
 </div>  
  
  
 <div class="row">  
 <div class="col-md-3">  
 {% block left\_body %}  
 <ul>  
 <li class="list-item">  
 左侧列表栏  
 </li>  
 <li class="list-item">  
 建议每次都替换  
 </li>  
 </ul>  
 {% endblock %}  
 </div>  
 <div class="col-md-9">  
 {% block right\_body %}  
 <p>  
 右侧版面主题  
 </p>  
 {% endblock %}  
 </div>  
  
 </div>  
 </div>  
 </body>  
</html>

对于网页的基本框架，可以在boostrap-demo页面看到，全体网页都是采用如下的结构：

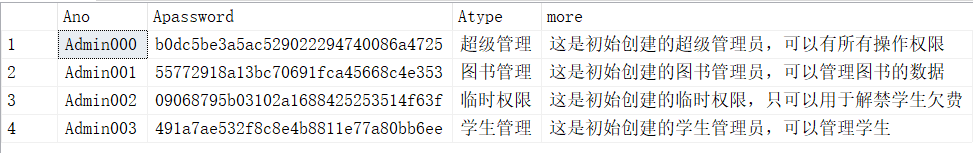
* 开头一行，放置页面大标题（下面加了一个进度条作为美化）
* 左侧3/12比例为列表，用于网页导航的链接放置
* 右侧9/12为页面主体

设计通过一个Boostrap框架构建网站来实现，基本结构效果如图



### 初始数据

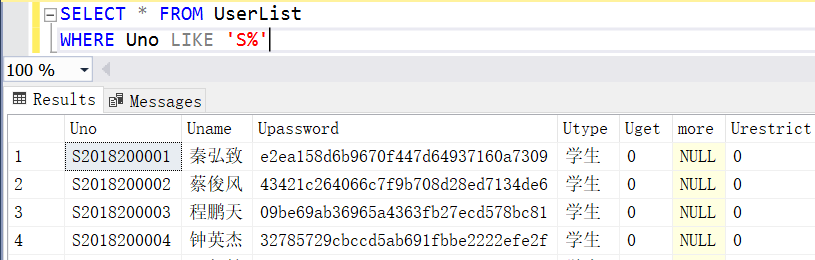
管理员表手工设定4个账号，插入之后效果如下：

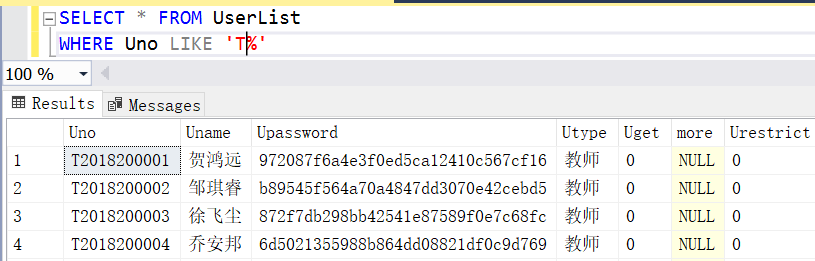


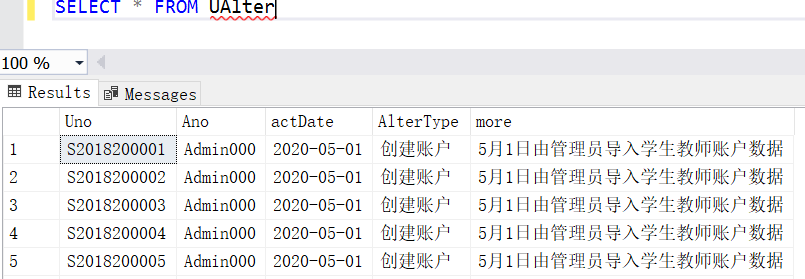
上述功能在data\_user.py的initAdmin函数实现

然后随机生成一些学生和老师（借阅者）：

* 姓名是利用网上的起名器，都是三个字的，存在Sname文件里面
* id分老师学生，是递增输入；老师学生是随机设定的，用于后续不同权限
* 备注暂时置空，被禁用设置为0（没有被禁）
* 在插入借阅者数据的同时，为用户数据操作记录表添加记录【同时对两个表进行操作】



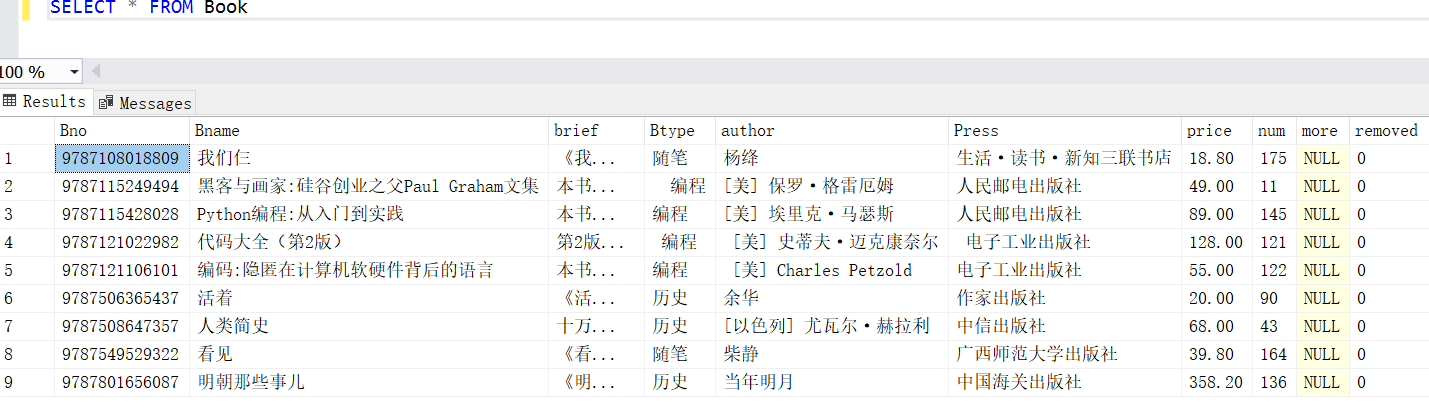


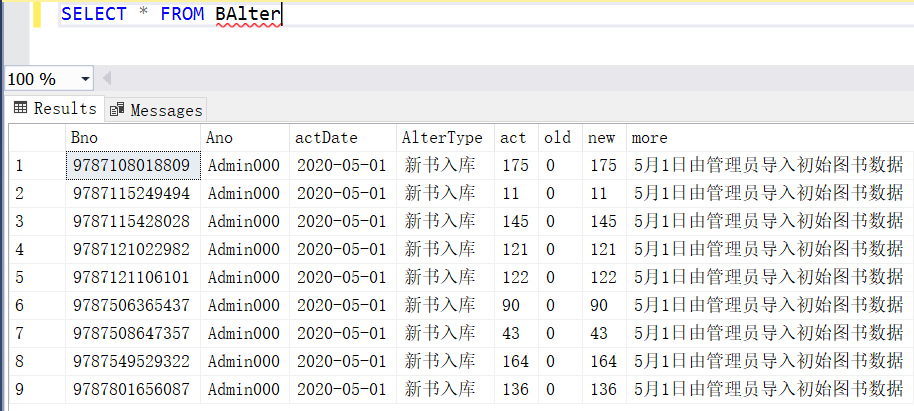


功能在函数randUser(n)实现

再生成一批图书：

* 此处由于图书要比较高的真实性，所以采用豆瓣的数据
* 先手动从豆瓣摘取一些数据，存储在本地
* 然后将图书数据插入数据库
* 插入的同时留存图书入库记录【同时插入两表】



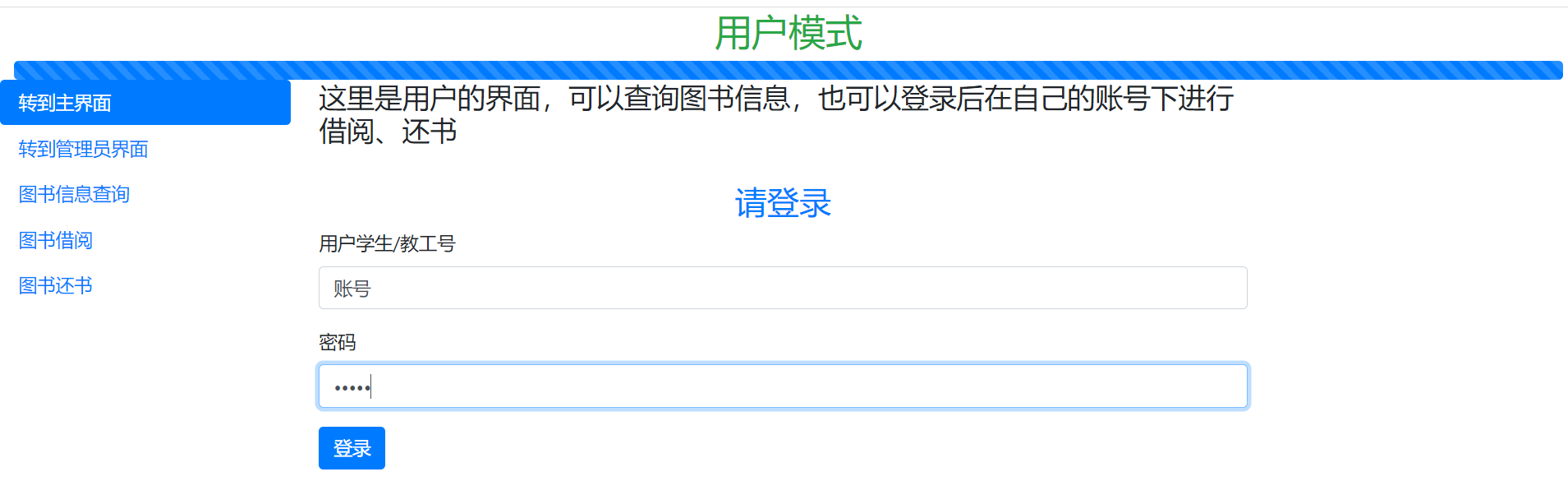


### 用户端功能

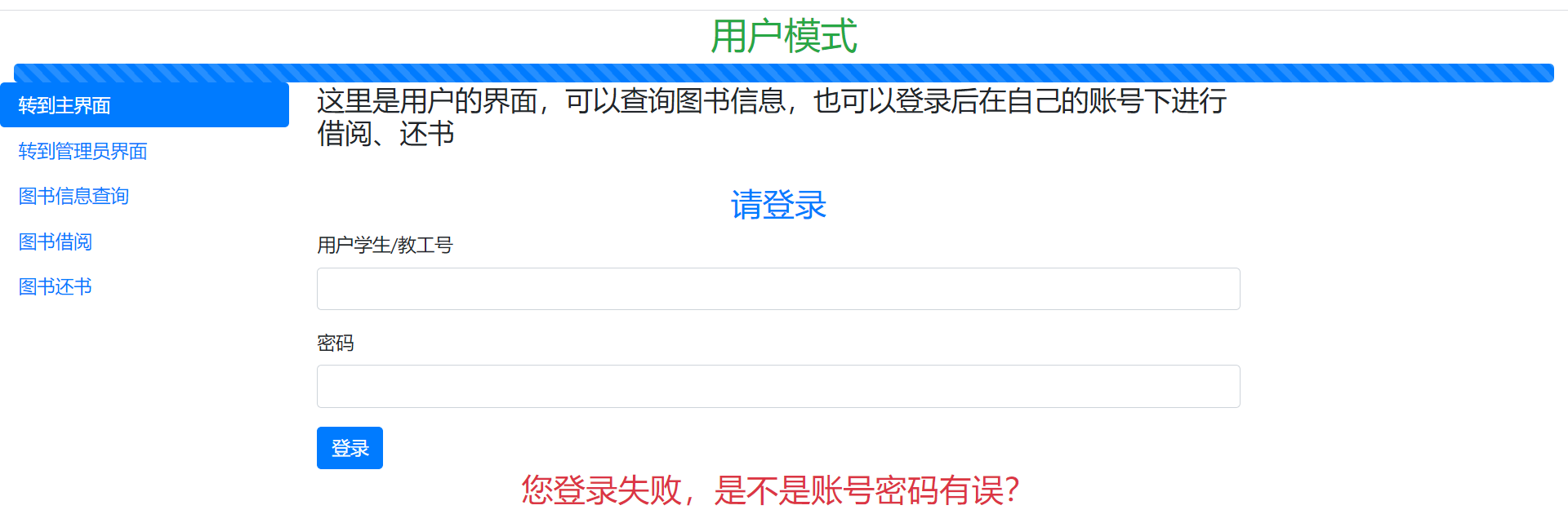
#### 登录

涉及函数：user,user\_login

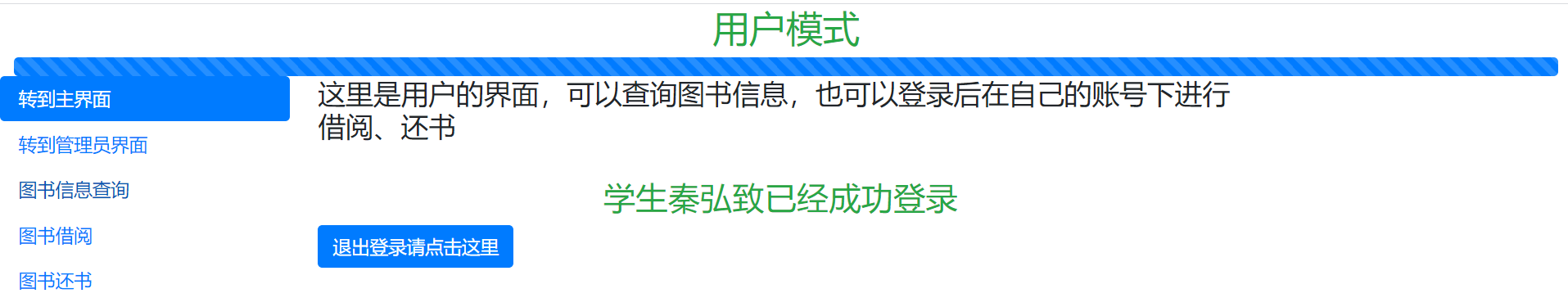
测试流程见截图，首先在user界面输入账号密码，然后根据是否成功有登录后和登录失败两个界面；



初始在/user/网页如图



登录失败，通过flask的redirect函数可以传递参数的特点，返回到/user/?error=True，从而网页获得一个GET的参数error，判定为登录失败，弹出红字提示。



登录成功后效果如图，**学生**是用户账号类型，**秦弘致**为用户姓名，下方有一个退出按钮

#### 登出

涉及函数user\_logout

点击上面的登出按钮，发送信息到user/logout/，将登录的信息清除后返回/user/页面，和初始进入相同。

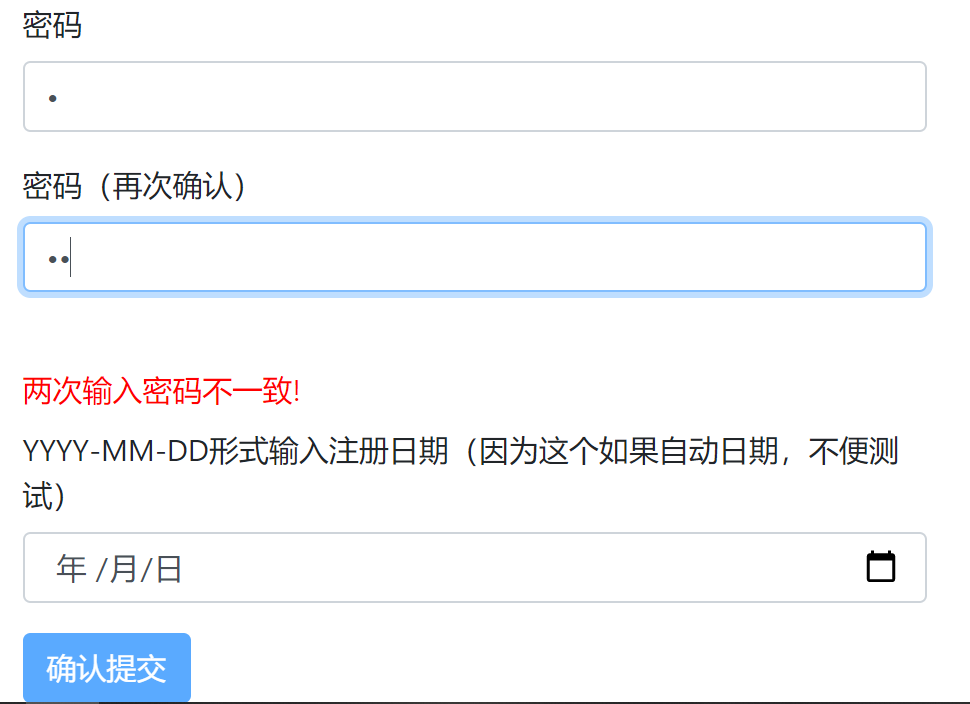
#### 注册

注册功能实现了对于表UserList和UAlter的同时插入，涉及的函数是user\_signup

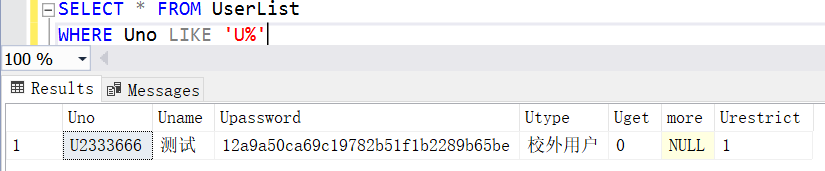
注册页面：

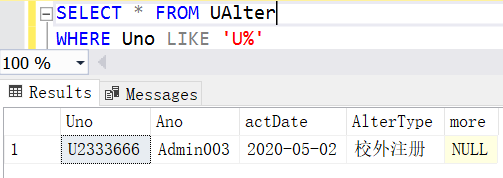


利用JS的函数，实现了对于两次密码的一致性检查，如果不一致，按钮会被禁用，从而保证密码是准确的。



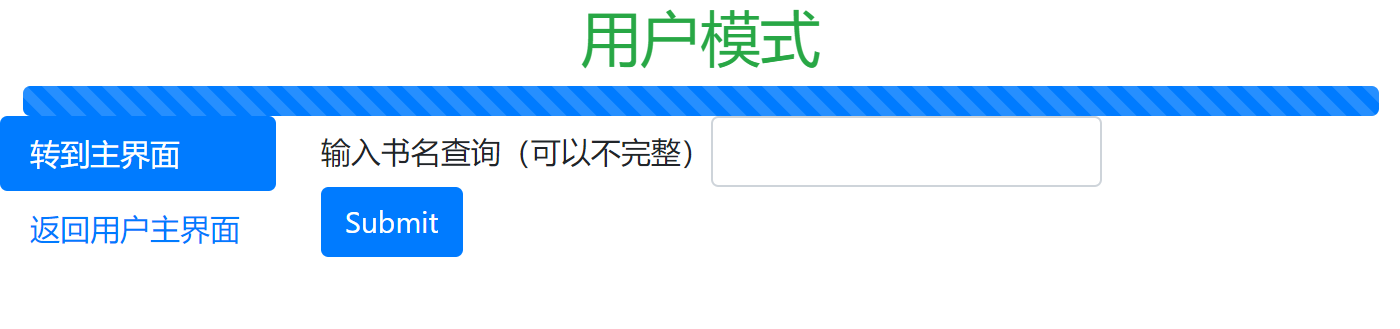
相应的数据库插入信息





#### 图书信息查询

涉及的页面是/user/search/函数是book\_search



在查询之后，查询信息&页码通过GET方法发送到这个页面自身：





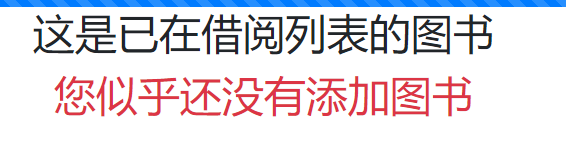
#### 图书借阅

显示页面是/user/borrow/，涉及的函数为book\_borrow,book\_add,book\_act

点击上面的“我要借书”按钮将书添加到借书的“购物车”，如果点击“看看有没有其他书”就回到查找界面，如果选好日期并借阅，那么将执行借书操作，然后清空列表，回到用户主界面/user/。



清空之后的“购物车”是这个样子：



#### 图书归还与续借

涉及页面/user/return/，涉及的函数是book\_return,book\_return\_act,book\_return\_addtime

续借以后不可以继续续借，没有续借的可以还书或者续借



### 管理端基础

基本逻辑同用户端，显示页面为/admin/对应函数是admin



### 用户管理

总体如图/admin/user-control/：

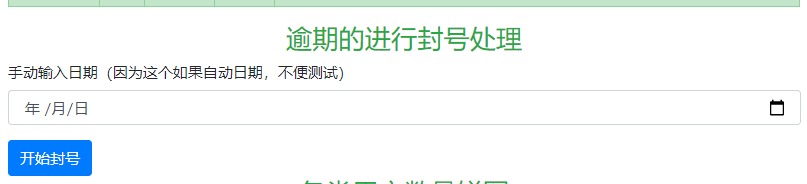


#### 被禁用的账号

被禁用的用户可以在管理这边进行解禁，点击按钮即可。

涉及的处理函数为free\_act

#### 逾期不还书封号





根据流程增加了新的违规学生

#### 固定参数统计图

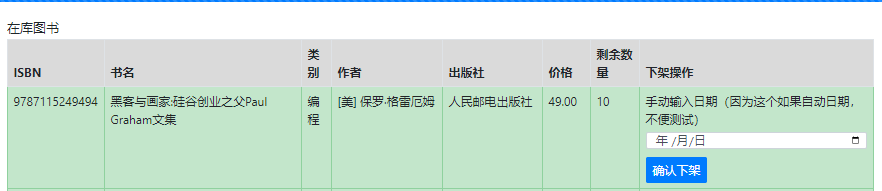
显示了不同类用户的数目柱状图

### 图书管理

#### 已知图书管理

页面/admin/book-control/对应函数为admin\_book\_reset,admin\_book\_remove

在库图书的显示



已经被下架图书的显示

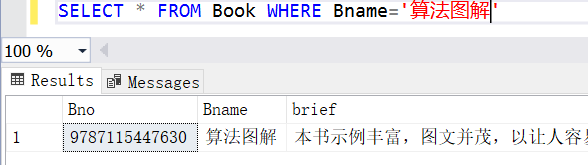


#### 图书入库

对应的网页是/admin/book-control/add/，函数是admin\_book\_add

添加图书入库操作：





### 超级管理

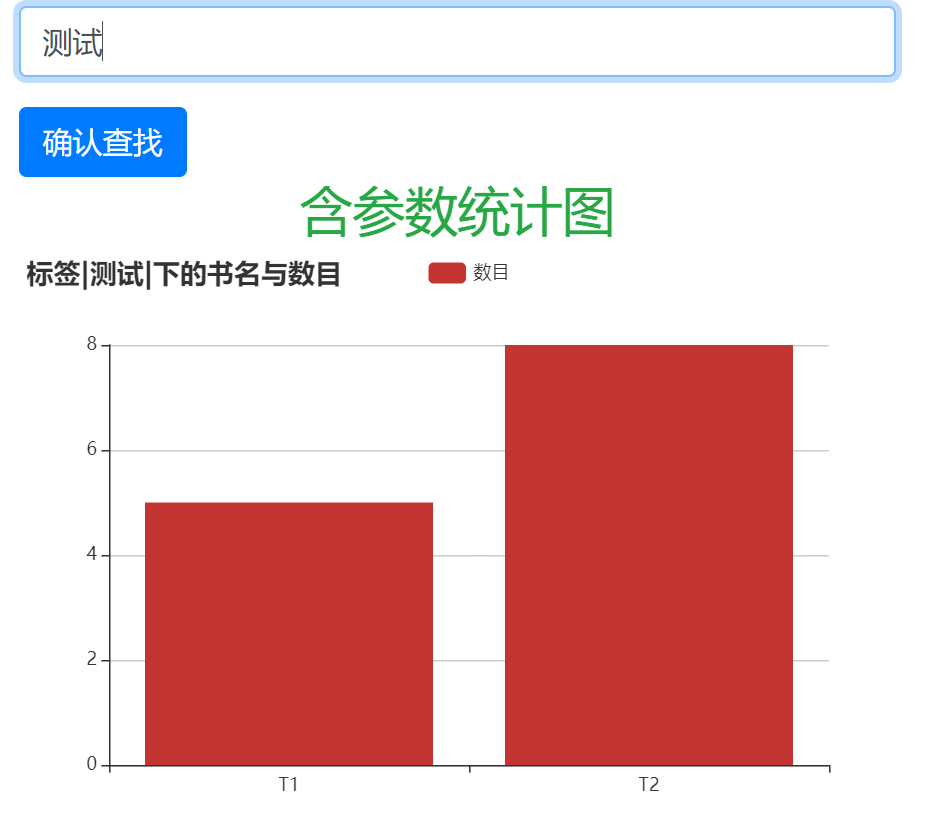
展示历史上对应用户和实体的操作记录，5个一页，可以翻页，效果如图



### 数据展示

对应页面/admin/data/函数,admin\_data

这是一个根据参数产生统计图的页面（主要是图书管理中汇总统计比较少有，所以就选择了根据指定标签找图书和数目）



## 总结

1基础

* 使用基础数据集，建库建表插入基础数据
* 固定参数(直接在代码里写死)，对数据库进行汇总统计，用html table显示结果
* 固定参数(直接在代码里写死)，对数据库进行汇总统计，用柱状图或者饼图显示结果

2进阶

* 允许用户输入参数，对数据库进行汇总统计，用html table显示结果【图书信息查询】
* 允许用户输入参数，对数据库进行汇总统计，用柱状图或者饼图显示结果【数据展示】

3高级

* 使用基础数据集、可以增删改基础数据集
* 网站包括home,insert,delete,upate,query（对数据库进行汇总统计）等完整的功能

4加分项

* 图和table同时显示在一个网页上【用户管理页】
* 如果能够实现1:n的两个表格数据的输入【类似购物车式的图书借阅】
* 数据模型构建完整
* 功能模型构建完整

5我的项目特色

* 基于Boostrap美化了前端的呈现效果，支持不同浏览器以及缩放。
* 提供了基于session的用户登录认证，并且对于没有登录的情况重定向到登录页面
* 为用户、图书管理员、用户管理员、超级管理提供了不同的权限和外模式，保证了安全性和独立性
* 提供了运用JS函数的表单检查，保证两次输入密码相同才可以注册/修改密码
* 在用户的基础业务之外为管理工作也设置了数据库和操作界面，每次的增删改都有迹可循
* 图片使用echart，因为是JS所以可以随鼠标移动而改变
* 配置了虚拟环境（不过这个项目可以不用虚拟环境）

## 参考文献和资料

虚拟环境的配置：

* https://blog.csdn.net/guying4875/article/details/80905472
* https://blog.csdn.net/happy*bigqiang/article/details/51168614?utm*medium=distribute.pc\_relevant.none-task-blog-baidujs-4
* https://blog.csdn.net/Growing\_hacker/article/details/89518534

boostrap参考：

* https://how2j.cn/k/boostrap/boostrap-setup/539.html如何使用boostrap（不涉及flask）
* https://v4.bootcss.com/docs/getting-started/download/ boostrap安装到本地
* https://www.layoutit.com/build 网页设计

flask整体教程：

* https://www.w3cschool.cn/flask/ W3教程
* https://blog.csdn.net/hanbo6/article/details/82563015；https://blog.csdn.net/wei18791957243/article/details/85237246用户登录功能的session方法

SQL server相关：

* https://docs.microsoft.com/zh-cn/sql/relational-databases/tables/create-foreign-key-relationships?view=sql-server-ver15 创建外键关系

数据资源：

* https://www.qqxiuzi.cn/zh/xingming/姓名生成器
* 豆瓣读书

FLASK+ECHARTS：

* https://blog.csdn.net/u013421629/article/details/78183315?utm*medium=distribute.pc*relevant*t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase&depth*1-utm*source=distribute.pc*relevant\_t0.none-task-blog-BlogCommendFromMachineLearnPai2-1.nonecase Flask+Echarts 实现动图图表
* https://blog.csdn.net/weixin\_34186128/article/details/91974308【通过tojson过滤器可以传参给js】
* https://my.oschina.net/tinyhare/blog/756485【通过定义js变量来传递数据】

JavaScript：

* https://blog.csdn.net/wangjian530/article/details/90343105 利用js检验表单数据

时间处理：

* https://blog.csdn.net/cmzsteven/article/details/64906245 python时间处理包
* https://www.cnblogs.com/huzhe123/p/9308057.html格式化讲解
* https://blog.csdn.net/appleheshuang/article/details/9139025 时间加减法