

实验报告

【姓名】 张雨薇 【学号】 11300240095

一、 实验题目

图书销售管理系统的设计与实现

二、 开发环境

操作系统: Linux

数据库管理软件: Sqlite

编程语言: python html css javascript

客户端或 Web 开发环境: django

三、 数据库设计

本项目建立 4 张表, 分别用来记录用户信息, 库存书目信息, 进货信息和账目信息。

ER 图见附件。

bookstore_stock_book: 库存数目信息。包括图书 ID, ISBN, 标题, 作者, 出版社, 零售价和剩余库存数量

bookstore_import_book: 进货信息。包括订单 ID, ISBN, 标题, 作者, 出版社, 进货价和进货数量

bookstore_bill: 账目信息。包括账目标号, 账目日期, 收入/支出, 交易额

auth_user: 系统生成基础用户信息。主要包括用户 ID, 用户名, 密码。

bookstore_profile: 用户定义具体个人信息, 一对一指向 auth_user。包括真实姓名, 工号, 性别, 年龄。

四、 系统设计

系统主要分成四个部分: 用户管理, 库存管理, 进货管理和账目管理。其中用户管理可以实现超级管理员建立和修改账户, 普通管理员自行修改信息。库存管理分为查看, 编辑和出售。进货管理分为进货, 付款, 取消订单和到货录入库存。账目管理为查看某一段时间内账目计算收入支出总和。

1. 用户管理

系统用户分为超级管理员用户和普通管理员用户。普通管理员用户在前台进行管理, 由超级管理员创建。只能对图书进货、销售等信息进行管理, 只能查询和修改自己的用户信息。超级管理员在后台对用户及图书进行管理, 在数据库创建时创建。

每位用户除了用户名和密码信息外, 还有真实姓名, 工号, 性别, 年龄等基本信息。

系统所有功能只有用户登录了才能进行操作。

当用户没有登录时, 点击其他界面会跳回登录界面。同样, 当用户登出时也会跳转到登录界面。如下:

login

admin

Welcome to vv's Book Store

Please log in on the right side to use vv's book store's management.

Administrator Log In

User name:

Password:

login

登录不成功时会返回当前页面并显示错误信息。

Administrator Log In

Username and password do not match!

User name:

Password:

login

超级管理员：通过

admin

 登录后台进行用户管理和图书管理等功能。

Django administration

Welcome, yuwe!

Site administration

Auth

Groups

Add

Change

Users

Add

Change

Bookstore

Bills

Add

Change

Import_books

Add

Change

Profiles

Add

Change

Stock_books

Add

Change

Sites

Sites

Add

Change

Recent Actions

My Actions

lumig

User

xchengcheng

Profile

cc

User

ruanlu

Profile

doudou

User

qianxu

User

lushengu

Profile

lumig

User

可以创建和修改普通用户。现创建用户 LumiG:

Django administration

Welcome, yuwe! Change password / Log out

Home > Auth > Users > LumiG

The user "LumiG" was added successfully. You may edit it again below.

Change user

History

View on site

Username:

LumiG

Required. 30 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

algorithm: pbkdf2_sha256 iterations: 10000 salt: hpbuX***** hash: YA/Z5p*****

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

用户创建成功，密码加密保存。现在修改用户信息：

Django administration Welcome, yuwen. Change password / Log out

Home > Bookstore > Profiles > LumiG

Change profile History

User:	LumiG
Name:	<input type="text" value="lushenggu"/>
WorkID:	<input type="text" value="11300240007"/>
Age:	<input type="text" value="20"/>
Gender:	<input checked="" type="radio"/> Male <input type="radio"/> Female

[Delete](#) [Save and add another](#) [Save and continue editing](#) [Save](#)

普通管理员：

普通管理员用户名为 LumiG，右面可以进入库存图书，进货和查看账目（具体功能稍后详细叙述）。

[LumiG's account](#) | [logout](#) [admin](#)

Welcome to vv's Book Store!

Hello, LumiG!


This is vv's book store, and you are the administrator of my book store. You are here to help me manage my book store.

You can search the books in this book store and edit them. You can also sell them to customers.

When you found the books are not enough, import some. Remember to deal with the orders! Pay it or cancel it. When the books arrive, put them in the stock.


Later, you can check the bills generated during your process of selling and buying.

Thanks for your help, LumiG! Enjoy your time!




Inventory

You can search for books in the stock and edit them. You can also sell them to customers.



Import Books

You can import books and deal with the orders. When the books arrive, you can put them into stock.



Check Bill

You can check bills during a certain period of time in this book store.

在左上角点击 [lumig's account](#) | [logout](#) 可以查看和修改个人信息或者登出。

查看和修改个人信息如下：

[LumiG's account](#) | [logout](#) [admin](#)

Hello! LumiG

name :

workID :

gender : ☒ Male ☐ Female

age :

2. 库存书籍管理（Inventory）


对库存书籍的管理包括：查询 Search，编辑 Edit 和出售 Sell。


在 Inventory 模块打开之后，显示前 10 条图书信息。右侧对图书进行接下来的操作。


Inventory

ID	ISBN	Title	Author	Publisher	Price	No
1	123	Are you happy?	lumig	cc	23	17
2	456	Lucky	vv	yuwei	23	5
3	66	hi	cc	cc	34	5
4	11300 24	Computer Science	CS	Fudan	45	9

What to do next?


[Search](#)


[Edit](#)


[Sell](#)

1) 搜索 Search

可以根据图书 ID，书名，作者，出版社进行搜索。其中 ID 是精确查询，其他都是模糊查询，只要包括所查找的字段即可。

Inventory

ID	ISBN	Title	Author	Publisher	Price	No
1	123	Are you happy?	lumig	cc	23	17
2	456	Lucky	vv	yuwei	23	5
3	66	hi	cc	cc	34	5
4	11300 24	Computer Science	CS	Fudan	45	9

Search

Book ID:

ISBN:

Title:

Author:

Publisher:


Search


当成功找到时返回以下界面：


Search results showed :

ID	ISBN	Title	Author	Publisher	Price	No
2	456	Lucky	vv	yuwei	23	5
4	11300 24	Computer Science	CS	Fudan	45	9

What to do next?


[Search](#)


[Edit](#)



[Sell](#)


当找不到对应书籍时返回：


Search results showed :

No books matched your search criteria.

What to do next?


[Search](#)


[Edit](#)


[Sell](#)

2) 编辑 Edit

根据给出图书的 id 号编辑图书信息。可以编辑标题，作者，出版社和零售价格。

Inventory

ID	ISBN	Title	Author	Publisher	Price	No
1	123	Are you happy?	lumig	cc	23	17
2	456	Lucky	vv	yuwei	23	5
3	66	hi	cc	cc	34	5
4	1130024	Computer Science	CS	Fudan	45	9

Edit

Book ID:

Title:

Author:

Publisher:

Retail Price:


Edit


比如更改 ID=1 这本书的 publisher 为 vv，会输出修改成功的结果，并显示修改后这本书的信息。


Edit successfully!

ID	ISBN	Title	Author	Publisher	Price	No
1	123	Are you happy?	lumig	vv	23	17

What to do next?

 Search

 Edit

 Sell

Id 为空或者不存在则会跳到找不到书的界面。

Inventory

No books found.
No books matched your search criteria.

What to do next?

 Search

 Edit

 Sell

3) 出售 Sell

根据图书 ID 输入数量进行售书。

Inventory

ID	ISBN	Title	Author	Publisher	Price	No
1	123	Are you happy?	lumig	vv	23	17
2	456	Lucky	vv	yuwei	23	5
3	66	hi	cc	cc	34	5
4	1130024	Computer Science	CS	Fudan	45	9

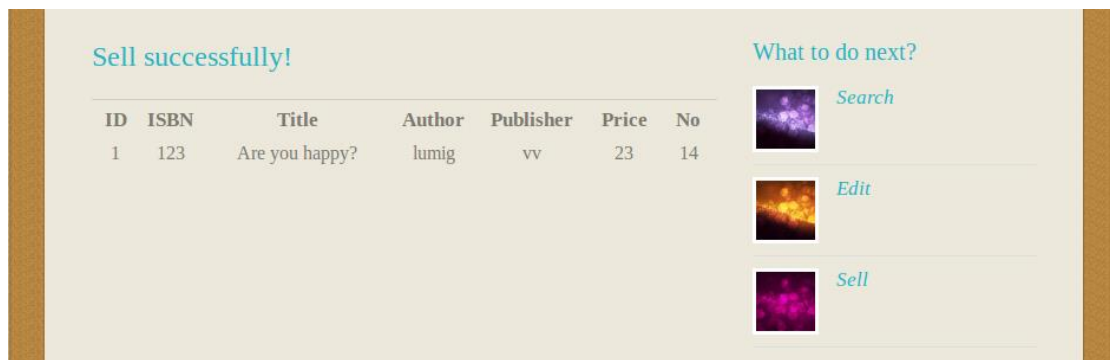
Sell

Book ID:

Number:

如图为卖出 ID 为 1 的书 3 本，在点击 sell 之后，会在 bill 中增加买书增加的款项。并会

显示售出成功的界面，同时看到数量相应减少：



如果 ID 不存在或者没有输入购书数量则会报错。

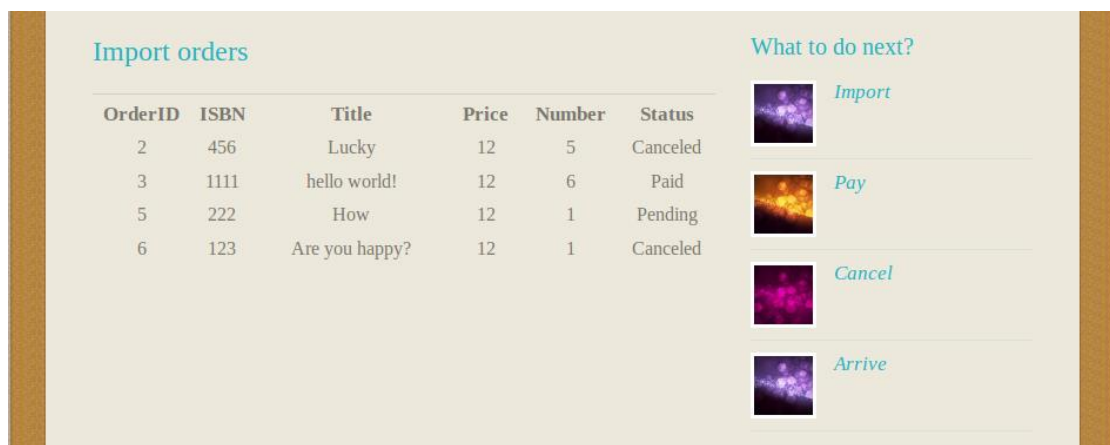
当输入数目大于库存数目时会显示库存不足，并且不对数据库进行修改：



3. 进货管理 Import Book

对进货的管理分为：进货 import，付款 pay，取消订单 cancel，书籍到货录入库存 Arrive
进货管理首页显示所有进货订单，订单状态包括：未付款 Pending, 已付款 Paid，已取消 Canceled

进货管理首页



1) 进货 import

可以根据图书 ID（在 search 中查到的已有的库存书籍）或者新输入图书信息购进新书。如果两个同时输入以 ID 优先。

如购入库存中已有 ID=1 的书：

Import orders

OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Pending
6	123	Are you happy?	12	1	Canceled

Import

ID:

ISBN:

Title:

Author:

Publisher:

Import Price:

Number:


buy


显示进货成功的结果：


Order added successfully!


OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Pending
6	123	Are you happy?	12	1	Canceled
7	123	Are you happy?	25	12	Pending

What to do next?

 Import

 Pay

 Cancel

 Arrive

如图为购进新书：

Import orders

OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Pending
6	123	Are you happy?	12	1	Canceled
7	123	Are you happy?	25	12	Pending

Import

ID:

ISBN:

Title:

Author:

Publisher:

Import Price:

Number:

buy

进货结果：

Order added successfully!

OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Pending
6	123	Are you happy?	12	1	Canceled
7	123	Are you happy?	25	12	Pending
8	99	Love	99	1	Pending

What to do next?

Import

Pay

Cancel

Arrive

如果同时输入 ID 和图书信息，以 ID 为准，仍读取数据库中已存入的图书信息。
接下来可以对已下单的书进行付款和取消

2)付款 Pay

付款首页面会显示所有还未付款的订单，输入订单号进行付款。如下：

Unpaid orders

OrderID	ISBN	Title	Price	Number	Status
5	222	How	12	1	Pending
7	123	Are you happy?	25	12	Pending
8	99	Love	99	1	Pending

Pay

Order ID:

Pay

付款后会显示付款成功，并跳回进货管理的首页，显示 ID 为 8 的订单号的状态为已付款：

Paid successfully!

OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Pending
6	123	Are you happy?	12	1	Canceled
7	123	Are you happy?	25	12	Pending
8	99	Love	99	1	Paid

What to do next?

Import

Pay

Cancel

Arrive

3)取消订单 Cancel

取消订单界面显示所有已下单但是未付款的订单信息。在右侧输入单号取消订单：

Unpaid orders

OrderID	ISBN	Title	Price	Number	Status
5	222	How	12	1	Pending
7	123	Are you happy?	25	12	Pending

Cancel

Order ID:


Cancel


取消订单成功后，跳回进货管理首页，显示 ID 为 5 的订单号的状态为已取消：


Cancel successfully!


OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Canceled
6	123	Are you happy?	12	1	Canceled
7	123	Are you happy?	25	12	Pending
8	99	Love	99	1	Paid

What to do next?

 Import

 Pay

 Cancel

 Arrive

4) 书籍到货录入库存 Arrive

到货界面显示所有已付款订单，输入订单号并根据进货价格制定零售价：

Paid orders

OrderID	ISBN	Title	Price	Number	Status
3	1111	hello world!	12	6	Paid
8	99	Love	99	1	Paid

Arrive

Order ID:

Retail price:


Put into stock


成功后显示成功放入库存，跳回进货订单管理首页，发现在所有订单中已经去掉了这一条。


You have put the books into stock!


OrderID	ISBN	Title	Price	Number	Status
2	456	Lucky	12	5	Canceled
3	1111	hello world!	12	6	Paid
5	222	How	12	1	Canceled
6	123	Are you happy?	12	1	Canceled
7	123	Are you happy?	25	12	Pending

What to do next?




 Import

 Pay

 Cancel

 Arrive

下面我们来查看一下库存来确保这条订单中的书确实已经放入库存中了。查看的流程如以上所叙述的查询库存的功能。

Inventory							What to do next?	
ID	ISBN	Title	Author	Publisher	Price	No		Search
1	123	Are you happy?	lumig	vv	23	14		Edit
2	456	Lucky	vv	yuwei	23	5		Sell
3	66	hi	cc	cc	34	5		
4	1130024	Computer Science	CS	Fudan	45	9		
5	99	Love	LumiG	LumiG	999	1		

可以看到，库存中新加入了一本叫 Love 的书，并且零售价变为了刚才设置的 999

4. 查看账目 Check Bill

由于之前的操作，现在数据库中已经生成了几条账目的信息，进入查看账目的界面就可以看到。并且显示了总的收入和支出。

Bills			Check certain periods:	
Total Income: 851 Total Expenditure: 1132.			Start Time:	<input type="text"/>
			End Time:	<input type="text"/>
			<input type="button" value="Search"/>	
Date	In/Out	Amount		
May 11, 2013, 10:20 p.m.	Out	516		
May 11, 2013, 10:21 p.m.	In	529		
May 12, 2013, 9:14 a.m.	In	46		
May 12, 2013, 11:23 a.m.	In	46		
May 12, 2013, 11:37 a.m.	In	0		
May 12, 2013, 9:24 p.m.	In	115		
May 12, 2013, 9:25 p.m.	Out	24		
May 12, 2013, 9:42 p.m.	In	46		
May 12, 2013, 9:45 p.m.	Out	115		
May 12, 2013, 9:46 p.m.	Out	306		
May 13, 2013, 3:32 a.m.	In	69		
May 13, 2013, 3:37 a.m.	Out	72		
May 13, 2013, 4 a.m.	Out	99		

在右侧输入开始时间和截止时间可以查询一段时间内的账目
当输入 2013-05-11 和 2013-05-13 会出现如下的查询结果：

Bills			Check certain periods:	
From 2013-05-10 to 2013-05-12			Start Time:	<input type="text"/>
			End Time:	<input type="text"/>
			<input type="button" value="Search"/>	
Total Income: 529 Total Expenditure: 516.				
Date	In/Out	Amount		
May 11, 2013, 10:20 p.m.	Out	516		
May 11, 2013, 10:21 p.m.	In	529		

五、 提交文件说明

ER diagram:数据库 ER 图

project/project/templates: css js images 文件夹为 css 模板用到的图片等，其他

html 文件见下表

base.html	模板
bill.html	查询账目信息界面
buy.html	进货管理主界面
buyin_form.html	进货界面
edit_form.html	编辑界面
homepage.html	主页
inventory.html	库存管理界面
login.html	登录界面
manage.html	用户个人信息管理界面
pay.html	付款界面
search_form.html	查询界面
sell_form.html	出售界面

注：除 base 外其他所有界面继承 base 模板

project/project/settings.py 系统配置文件
project/project/urls.py 所有 url 的配置，对应到 views 文件中的函数
project/project/views.py 包括所有视图函数（源代码包含注释）
project/bookstore/admin.py 超级管理员权限设置
project/bookstore/models.py 模型设计

六、 实验总结

这次数据库中期实验真正让我学到了很多東西。

没有找到人和我一组，就只能逼着自己写了。我之前从来没写过图形界面的东西，桌面程序或者网页都没接触过，这是我第一个遇到的最头疼的问题，问了很多大神要学什么，怎么把数据库和界面连接起来等等，最后选择了 python 和 django。

从拿到期中作业开始学起，从头看着网上的教程 Django Book 学 Django，真是毫无基础，甚至我都没写过 python 语言。就这么边看边学，学了几个星期，把基础功能学了一遍还没开始做 PJ，他们都劝我，你看也没有多少时间了，要不然你换一个写吧，用 C#吧，挺多女生都用这个，也挺快就写好了，django 这个就当自己学着玩了，重新再学那个也挺快的。但我觉得，我都学了这么久了，再换另外一个真不甘心。所以咬咬牙说，不，就接着学这个。

所以现在看到我自己一个人写出的这个东西，我和他们调侃说，一周之前的我绝对想不出自己能写出这样一个网页。我去大神寝室问问题，被嘲笑说写的好简陋……但是我还是觉得以我的能力我很知足很满意了。在做的过程中半夜一个 error 的调试语句看不懂，室友都睡了我一个人坐在那儿哭，就觉得自己怎么这么笨，可是现在看来，所有努力都可以得到收获的。

抒情好久，具体说说实验中遇到的问题和学到了什么吧。

开始在 Django 下配置数据库这些完全是依据网上的教程来的，但是想连接 Mysql 的时候发生了奇怪的问题，询问之下改用了 sqlite，又是一个我从没听过的东西。我还问别人，

我觉得我写的语句也没用到标准 sql 语句也没对数据库进行操作，被嘲笑说，那你添加的那些书本账目，改动的那些都在哪里改的啊，凭空出现的吗……

越写越觉得吧，Django 真的蛮好用的，把查询语句和 html 完全的割开了，可以先把 url 配置好，对应到查询，再去写 html。然后又去学了 html，又是一个我没写过的东西。不过还好，就按照 w3school 上的 html 跟着做了一遍，反正我也不要求做的多好看，简单功能都可以用，表单输入提交都可以做就 ok 了。

所以我在掌握了这些基础知识之后终于开始写了，就写了一个白色底最简陋的那种网页，但是自己一点一点把所有的功能都实现出来了。本来我想就这样挺好的了，然后有人跟我说，你去套一个 CSS 的模板吧。CSS，又一个我不知道的东西。我上网找到了模板找人教我怎么用，怎么用 {% block %} 把要自己写的东西拿出来放在 html 里。于是又用了一段时间，把之前写的极其简陋的界面套上了模板，瞬间觉得高端了……在套模板的时候我发现了我之前写的网页的一个致命的缺陷：建了太多的网页！虽然现在看起来 html 文件还是很多，但是已经少了一些了。最开始写的时候被人警告过这个问题，但是因为之前实现的功能很简单，一般就是一个输入框，再加一个提交按钮。所以当时我查询建了一个查询界面，查询结果出来再建一个界面吧，编辑要一个界面，售书要一个，进货，付款，取消这些统统都要！结果改起来的时候觉得工作量好大……这就导致了我想重新规划一下我的逻辑，不能 10 几条功能并行啊！于是再改过之后就变成了现在的只分成库存，进货和账单三个大目录，每个下面再有几个小目录的样子。

还有一些很琐碎的问题，比如自己测试的时候想到卖书的时候超出库存数数量怎么办？没有输入 id 怎么办？都没有进行错误处理，就会自己跳转到 404，于是又重新对错误处理了一遍，觉得自己当初代码还是写的不够好不够清晰。再比如，拿给别人看的时候，有人提出很犀利的 3 个问题，你这个表格里面显示的如果书的名字太长会怎么样？我说表格会变长，甚至到网页外面去。于是又上网搜索解决办法，把表格变得稍微漂亮一点……

说到这里其实差不多了，其实用 Django 还是很强大的，就是在出现问题的时候没有人可以问……上网搜索的话基本上就要去 stackoverflow 或者去 Django document 上看官方说明，一般只有英文看起来好难过。周围的人用的也很少，知道的也不过是和我一样才开始学习的，不过人家是大神嘛，还是懂的更多，也算可以有人能问到。以及还好当时没有换开发语言，现在回头看真的学到不少东西。

这一个月从接到题目到写完，学到了很多以前从不知道的东西，也发现自己潜力无穷，有人说你一个妹子学这个干嘛，但是我不要和别人不一样，大家都能做到的为什么我做不到。所以最后作出这个成果的时候我还是小小的骄傲了一下的：)

最后感谢陆盛谷同学的支持和帮助。