**图书借阅管理系统 实验报告**

­­

16307130194 陈中钰

16级 计算机科学技术学院

Contents

[1 数据库设计 2](#_Toc514716823)

[2 实现方式 3](#_Toc514716824)

[3 技术重点 5](#_Toc514716825)

[4 其他特色 7](#_Toc514716826)

[5 实验改进 8](#_Toc514716827)

[6 实验感想 8](#_Toc514716828)

# 数据库设计

* 1. 需求分析
* 记录图书馆各分馆基本信息；
* 记录图书基本信息、图书所在分馆、是否能被借出或预约、图书评论、增加、删除图书；
* 记录用户基本信息、修改信息、修改密码、注册、登陆、登出；
* 查找图书、借阅、归还图书、评论图书、删除评论、预约图书、预约完成；
* 超级用户管理员工、员工管理读者并处理读者需求；
* ……
  1. ER图

Figure ER图

* 1. 关系模型

可以用mysqldump -h localhost -u root -p book > D:\book.sql并输入密码来导出数据库表结构及数据，在此简要给出各个表的结构（其中：红色为主键、蓝色为外键），其他详情请看附件book.sql。除了如下6个表外，还使用了django自动生成的AuthUser表，用来储存用户信息及权限级别，而使用django生成的表的好处是可以随时取用信息而不用导入。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| tables |  |  |  |  |  |  |  |  |
| books（记录书本信息） | isbn | title | authors | publisher | price |  |  |  |
| librarise（记录图书馆信息） | lno | lname | laddr |  |  |  |  |  |
| comments（记录书评） | comno | isbn | text | id |  |  |  |  |
| loans（借阅记录） | loanno | id | loan\_date | due\_date | renewed | stono | return\_date | admin\_id |
| reserves（预约记录） | reno | stono | id | lno | admin\_id |  |  |  |
| storages（存储记录） | stono | isbn | lno |  |  |  |  |  |

* 1. 关系模型分析

一般数据库只需要达到3NF或BCNF即可，分析可以得到当前关系模型至少是BCNF，因此合格。

# 实现方式

* 1. 技术架构

django 2.0.5 + python 3.6.5 + Anaconda + pycharm 2018.1.1 + mysql 5.7.21

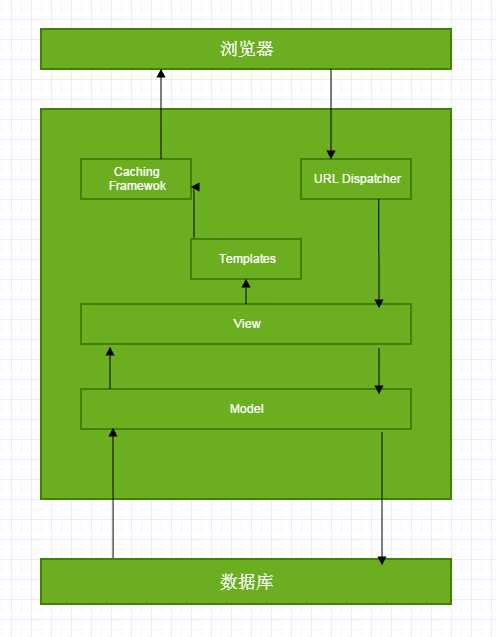
* pytharm：作为IDE
* django Web架构：即MVC开发框架，把代码的定义和数据访问的方法（模型M）与请求逻辑 （控制器C）还有用户接口（视图V）分开来。其中数据库用mysql实现，template主要用Materialize实现前端框架。
* 框架运行逻辑：用户在浏览器中输入url后回车, 浏览器会对url进行检查, 首先判断协议,如果是http就按照 Web 来处理, 然互调用DNS查询, 将域名转换为IP地址, 然后经过网络传输到达对应Web服务器, 服务器对url进行解析后, 调用View中的逻辑, 通过Model与数据库的进行交互，并处理数据，然后将数据发到Template进行渲染, 然后发送到浏览器中, 浏览器以合适的方式呈现给用户。

Figure django Web框架

* 1. 架构搭建（网上有教程）
* 安装所需内容；
* 在pytharm新建项目，通过命令行创建app和超级管理员；
* 在mysql中建好表，通过命令行把mysql的表结构导入到models.py中；
* 建好static、template文件夹，分别用于放静态文件（图片、css、js等）及html文件；
* 在pytharm连接数据库，并在settings.py中设置好DATABASES参数；
* 在settings.py中设置时区、LOGIN\_URL、static文件路径等参数。
  1. 系统角色
* Super User: 超级管理员，通过AuthUser中的is\_superuser属性来确定，拥有前台所有权限，能进入django后台管理所有数据，还能设置Staff来负责处理用户需求；
* Reader:可以查看自己的借阅、书评、预约记录，可以查找、点评、借阅、预约图书，可以查看书评，可以删除自己的评论；
* Staff:员工，通过AuthUser中的is\_staff属性确定。拥有前台所有权限，不能进入后台，负责处理用户需求，确认图书预约、归还图书，增加、删除书籍，管理评论等。
  1. 场景及功能流程

1. 主页

拥有到其他全部页面的链接，分布在导航栏图标、logo、主页按钮、侧栏导航栏中。其中，导航栏有已登陆及未登陆两种状态，分别显示不同的图标链接。主页上还有两个按钮链接，页底有3个内容链接。点击logo图片或文字会跳转回主页。当屏幕变窄时，会出现侧框导航栏，点开图标会在左边出现全部链接。这些主要通过设置url来实现跳转。

1. 图书馆

可以查找书籍，并可以点进图书标题，来查看图书信息、存储信息、借阅、预约、查看点评、进行点评，以及管理员可以增加图书、删除图书、删除点评。

1. 借阅表

管理员可以查看全部借阅信息，并负责归还书籍。读者只能查看自己的借阅信息，每本书可以续借一次。借一次书可用30天，续借后可延期30天。还可以点进图书标题，来查看图书信息。

1. 点评表

管理员可以查看并管理全部点评。读者可以看到、管理自己的点评，另外，在图书列表中可以查看其他人对任一本书的点评。还可以点进图书标题，来查看已点评的图书信息。

1. 预约表

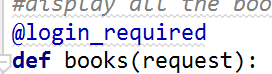
读者只能看到自己的预约，预约书籍可以使书籍等待运送到目标图书馆，而预约后只能等管理员确认。当书籍到达目标图书馆时，管理员可以确认预约，同时书籍所在图书馆变更。

1. 用户管理

有登陆、注销、注册、修改信息、修改密码等功能。

# 技术重点

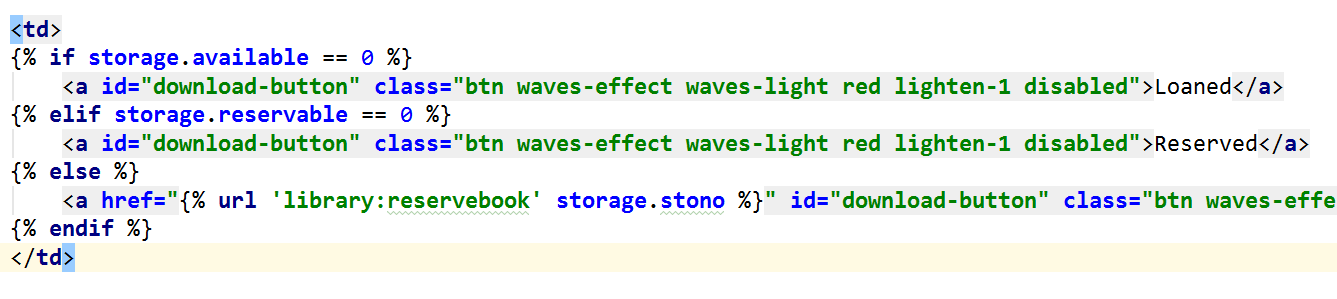
* 1. 非法访问限制
* 登陆限制：在settings.py中设置好LOGIN\_URL，在需要登陆了才能执行的view的def前面加上@login\_required，那么当未登陆的时候访问了这个view，就会跳转到登陆页面要求登陆；

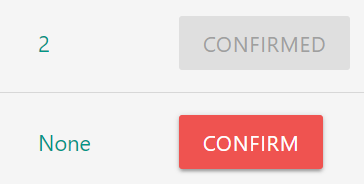


* 权限限制：有些操作只有staff、superuser才能做，因此在这些操作前都要判断request.user.is\_staff属性是否为True，是的情况下才能允许进行操作；

C:\Users\ECHOES\AppData\Local\Temp\1526920820(1).png

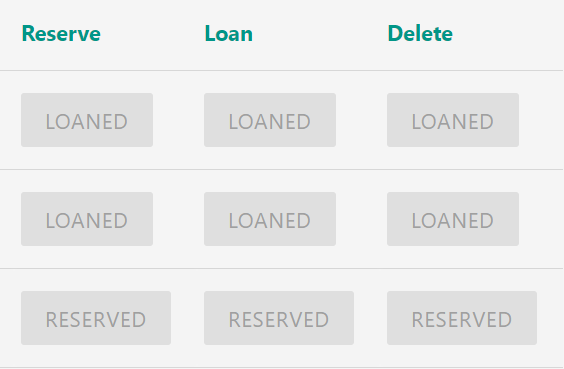
* url限制：当需要添加登陆限制、权限限制的地方都添加好后，就能会有完备的url限制，能有效防止用户通过url来访问权限更高的内容、操作；
* 按钮形态限制：在views中传所需的Query Object到html中，在html中使用Query Object中的数据库数据，如是否借出的信息，来控制借书按钮是正常状态或disabled状态。

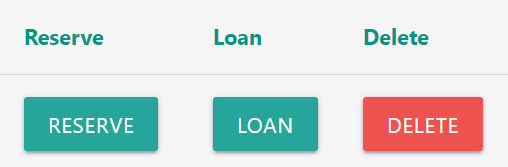




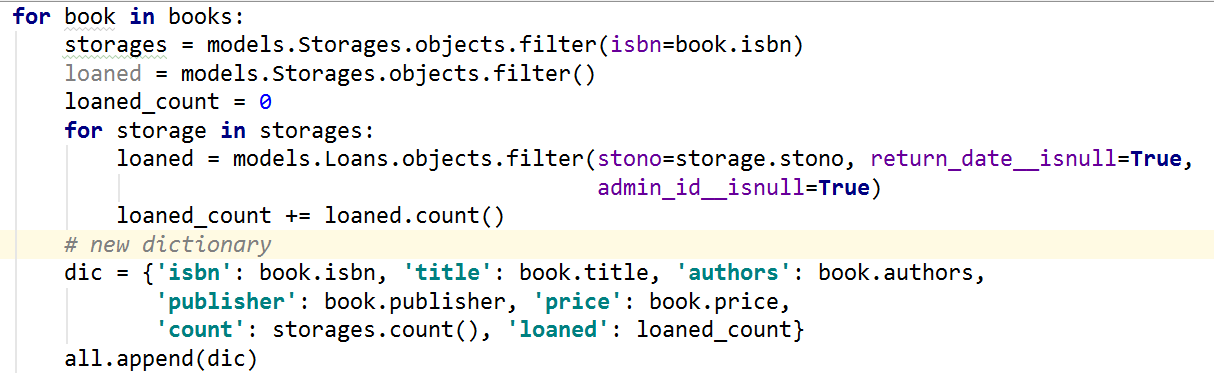
* 1. 预约、借阅、删除图书之间的关系

预约了的书不可以借阅、删除，借阅了的书不可以预约、删除，只有没预约、没借阅的书才可以被删除。





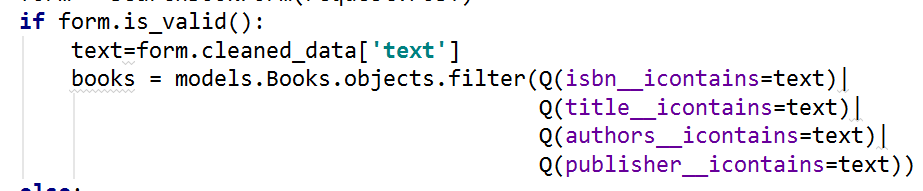
* 1. 各本书的库存数量及已借出数目



通过filter函数来查找这本书的库存的存储条目，用count函数求库存数目；通过filter来选每个存储条目的借出记录，并统计借阅总数。由于查询结果的Query Object不可以修改，所以最后新建一个包含全部内容的字典列表，并通过render函数传到html中使用。

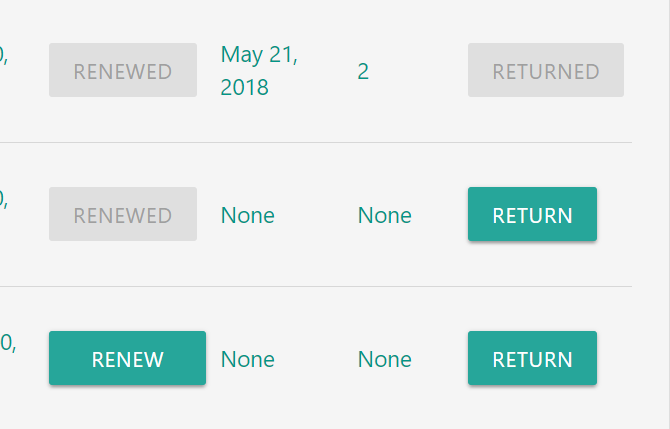
* 1. 查询操作

查询操作可以同时匹配title、isbn、authors、publisher中含有输入的text的书籍。通过使用django自带的Q函数来实现。



* 1. 信息保留

借阅、预约信息即使结束之后，信息仍然保留，但是会记录处理管理员的id和处理日期，那么通过判断管理员id和处理日期是否为空，就可以获知该记录是否结束。



* 1. 表格提交错误信息提示

通过在forms中添加约束，就能在{{form.as\_p}}生成表格的时候自动添加错误信息，会在输入不符合forms中要求的时候，提示输入错误。

# 其他特色

* 1. 有屏幕大小自适应的能力
* 通过诸如<div class="col s12 m12"></div>的方式对网页页面进行栅格化，使得网页中模块的大小会按照浏览器大小的比例缩放；
* 通过Materialize 的<table class="responsive-table highlight">来使得页面中的表格会按照浏览器大小进行自适应变化；
* 添加的sidenav(侧边导航栏)在屏幕变小的时候菜单按钮，点开按钮后出现左侧弹出菜单栏；
* 由于有屏幕自适应的功能，使得此网页能适应各种多媒体使用，如：手机、平板、电脑等。
  1. 简洁优雅的界面
* 通过使用Materialize的模板来设计页面，多花心思就好了；
* 拥有诸如高亮、悬浮等视觉效果
  1. 拥有众多链接，极大地方便用户
* 从用户角度出发，链接应该是容易辨认、明显、完整的。通过Materialize可以完成

# 实验改进

* 1. 功能
* 可以添加逾期罚款、添加好友、对评论进行评论、打分、高级搜索等功能；
* 添加头像、个人主页背景等与照片相关的设计；
* 学习并了解更多的前后端数据交互的方式，例如ajax、vue等；
* ……
  1. 界面
* 可以添加更多有趣的视觉效果及部件

# 实验感想

* 在一段相当短的时间里独立从0开始学习写Web，虽然采用的是django的亲民方式，但是也极大地提高了网页编程的能力；
* 学习到基本的Web框架，以及基本的前后端数据交互方式；
* 加深了对数据库设计的认识，意识到数据库设计在日常生活中的重要性；
* 提高了编程的条件控制能力。