

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 |
| 2 | 实现方式 | 3 |
| 3 | 技术重点 | 5 |
| 4 | 其他特色 | 7 |
| 5 | 实验改进 | 8 |
| 6 | 实验感想 | 8 |

1 数据库设计

1.1 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；
-

1.2 ER 图

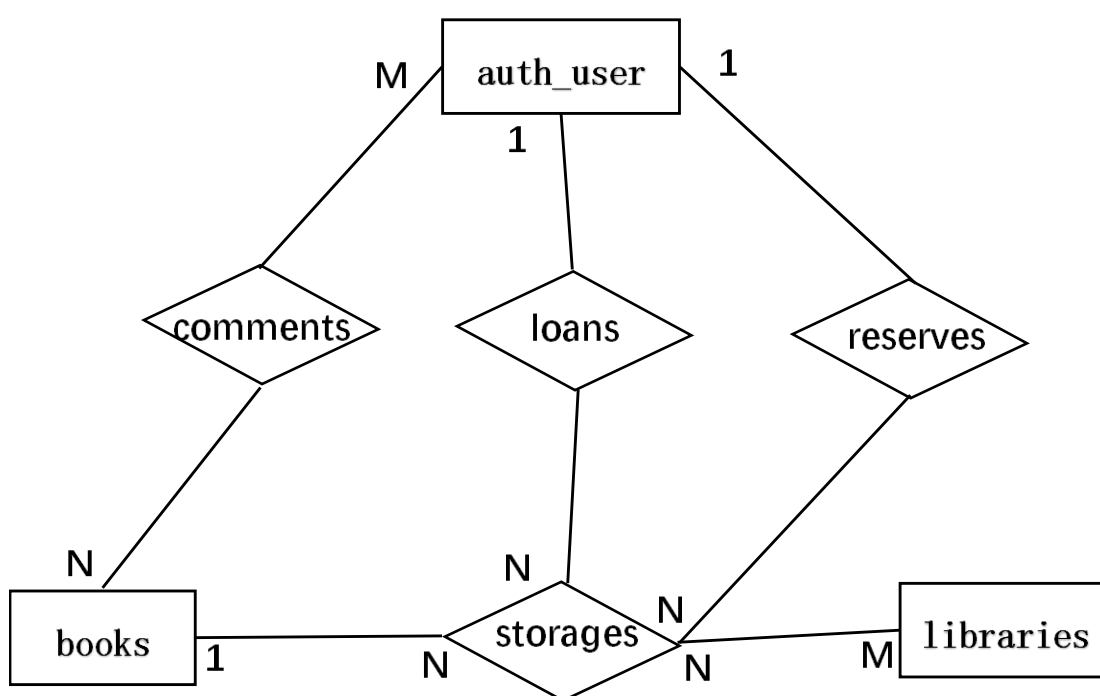


Figure 1 ER 图

1.3 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 | | | |
| libraries | lno | lna | laddr | | | | | |

| | | | | | | | | |
|-----------------|---------|-------|-----------|----------|----------|-------|-------------|----------|
| se (记录图书馆信息) | | me | | | | | | |
| comments (记录书评) | comno | isbn | text | id | | | | |
| loans (借阅记录) | loan no | 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 | | | | | |

1.4 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。

2 实现方式

2.1 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，然后调用 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。

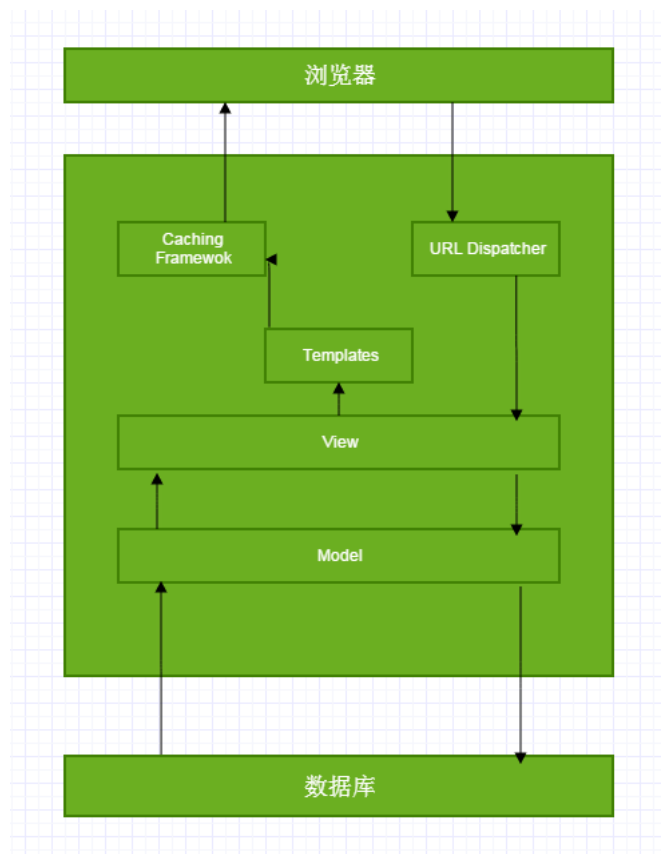


Figure 2 django Web 框架

2.2 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。

2.3 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。

2.4 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。

2. 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。

3. 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。

4. 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。

5. 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。

6. 用户管理

有登陆、注销、注册、修改信息、修改密码等功能。

3 技术重点

3.1 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；

```
#display all the doc
@login_required
def books(request):
```

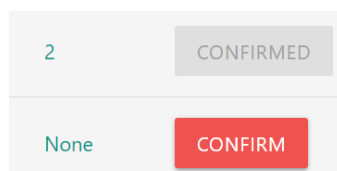
- **权限限制:** 有些操作只有 `staff`、`superuser` 才能做，因此在这些操作前都要判断 `request.user.is_staff` 属性是否为 `True`，是的情况下才能允许进行操作；

```
all=[]
if request.user.is_staff:
    login_model = login_obj
```

- **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。

```
<td>
{% if storage.available == 0 %}
  <a id="download-button" class="btn waves-effect waves-light red lighten-1 disabled">Loaned</a>
{% elif storage.reservable == 0 %}
  <a id="download-button" class="btn waves-effect waves-light red lighten-1 disabled">Reserved</a>
{% else %}
  <a href="{% url 'library:reservebook' storage.stono %}" id="download-button" class="btn waves-eff
{% endif %}
</td>
```



3.2 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。

| Reserve | Loan | Delete |
|----------|----------|----------|
| LOANED | LOANED | LOANED |
| LOANED | LOANED | LOANED |
| RESERVED | RESERVED | RESERVED |

| Reserve | Loan | Delete |
|---------|------|--------|
| RESERVE | LOAN | DELETE |

3.3 各本书的库存数量及已借出数目

```
for book in books:
    storages = models.Storages.objects.filter(isbn=book.isbn)
    loaned = models.Storages.objects.filter()
    loaned_count = 0
    for storage in storages:
        loaned = models.Loans.objects.filter(stono=storage.stono, return_date__isnull=True,
                                              admin_id__isnull=True)
        loaned_count += loaned.count()
    # new dictionary
    dic = {'isbn': book.isbn, 'title': book.title, 'authors': book.authors,
          'publisher': book.publisher, 'price': book.price,
          'count': storages.count(), 'loaned': loaned_count}
    all.append(dic)
```

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。

3.4 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。

```
if form.is_valid():
    text=form.cleaned_data['text']
    books = models.Books.objects.filter(Q(isbn__icontains=text)|
                                         Q(title__icontains=text)|
                                         Q(authors__icontains=text)|
                                         Q(publisher__icontains=text))
```

3.5 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。

| | | | |
|---------|--------------|------|----------|
| RENEWED | May 21, 2018 | 2 | RETURNED |
| RENEWED | None | None | RETURN |
| RENEW | None | None | RETURN |

3.6 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。

4 其他特色

4.1 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。

4.2 简洁优雅界面

- 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

4.3 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

5 实验改进

5.1 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；
-

5.2 界面

- 可以添加更多有趣的视觉效果及部件

6 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。