



数据库应用系统设计 实验报告

-----图书馆管理信息系统

计算机科学与技术学院

计算机科学与技术专业

2020 级 4 班

姓名：隋春雨

学号：202000130198

任课教师：梁文革

助教：马桢

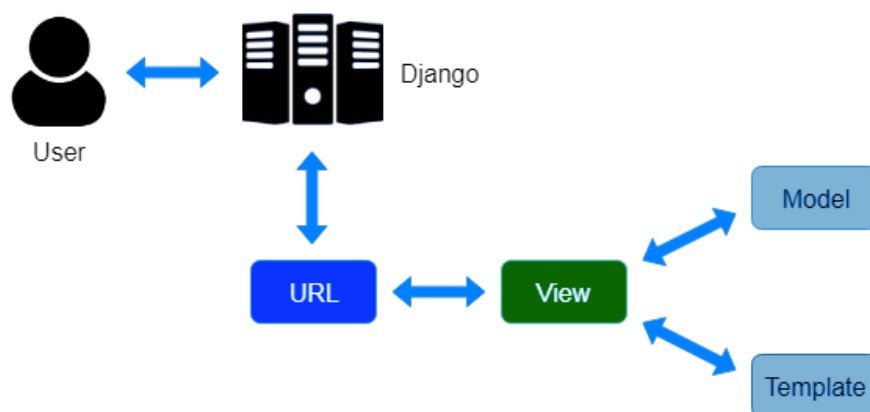
目录

一、项目简介	2
1.1 题目	2
1.2 开发工具	2
1.3 DBMS	3
1.4 操作系统	3
二、数据库规划	4
2.1 任务陈述	4
2.2 任务目标	5
三、系统定义	5
3.1 系统边界	5
3.2 用户视图	7
四、需求分析	8
4.1 用户需求说明.....	9
4.1.1 数据需求.....	9
4.1.2 事务需求.....	10
4.2 系统需求说明.....	11
五、数据库逻辑设计	13
5.1 ER 图	13
5.2 数据字典	13
5.3 关系表	16
六、数据库物理设计	16
6.1 索引	16
6.2 视图	16
6.3 安全机制	17
6.4 其他	17
七、应用程序设计	17
7.1 功能模块	17
7.2 界面设计	17
7.3 事务设计	17
八、测试和运行	17
九、总结	17
附. 参考文献	17

一、项目简介

1.1 题目

选择图书馆管理信息系统，使用 **python** 的 **Django** 框架，结合 **MTV** 模型进行了本项目的建立，本系统功能分为读者信息管理模块、图书信息管理模块、借阅记录管理模块、图书借阅管理模块、图书归还管理模块、管理员模块并及时记录各个信息的变更。在本系统中，考虑到面向对象的不同，将分为两类用户，普通用户（师生）、超级用户（管理员）。读者想要借阅哪一本图书可以使用图书信息管理模块进行查询（支持模糊搜索），查询到了之后可以通过图书借阅管理模块进行借阅操作，同时，读者借阅信息模块中留下相应的记录。对于**超级用户**，可以进行各个信息模块的增、删、改、查操作，并进行特殊情况的处理（比如删除违约记录等）



1.2 开发工具

1.2.1 IDE

Pycharm: Pycharm 拥有一般 IDE 具备的功能，比如说：调试、语法高亮、project 管理、代码跳转、智能提示、自动完成、单元测试、版本控制。

另外，Pycharm 还提供了一些很好的功能用于 Django 开发，同时支持 Google App Engine，同时还支持 ironPython。



1.2.2 框架

Python web 开发具有很多框架，轻量级的比如 flask Bottle 重量级的比如 Django，我选择的是 Django, 因为 Django 集成了安全认证，URL Routing，模板引擎，ORM 以及数据库 Scheme 映射。并且最最重要的是，它的用户群体很广泛，不用担心自己写到一半遇到了一个错误，百度一搜，一个相关信息都没有



1.3 DBMS

SQLite:

- SQLite 是一个非常轻量级的数据库。因此在电脑，手机，相机，家用电子设备等设备的嵌入式软件是非常好的选择。
- SQLite 的数据存储非常简单高效。当您需要存储文件存档时，SQLite 可以生成较小数据量的存档，并且包含常规 ZIP 存档的大量元数据。
- SQLite 可以用作临时数据集，以对应用程序中的一些数据进行一些处理。
- 在 SQLite 数据库中，数据查询非常简单。您可以将数据加载到 SQLite 内存数据库中，并随时提取数据。可以按照您想要的方式提取数据。

1.4 操作系统

Microsoft Windows 10



Microsoft Windows

版本 20H2 (操作系统内部版本 19042.1348)

© Microsoft Corporation。保留所有权利。

Windows 10 家庭中文版 操作系统及其用户界面受美国和其他国家/地区的商标法和其他待颁布或已颁布的知识产权法保护。

二、数据库规划

2.1 任务陈述

图书馆管理系统是一个学校不可缺少的部分,它的性能的高低与可靠性高低直接关系到广大师生的借书效率与学习效果。随着信息时代的普及,师生们学习动力提高,对图书的需求也随之增加,自然对图书馆管理系统的要求也随之升高。面对庞大的信息量,需要有一个有效的、性能优秀的住院管理系统来提高图书馆管理员管理工作的效率。一个完善的图书馆管理系统能够极大地提高广大师生以及图书馆管理员信息检索的效率,具有检索迅速、可靠性高、成本低、性能优秀等特点

结合所用框架 Django 的 MVT 模型,和本题的要求,本系统功能分为读者信息管理模块、图书信息管理模块、借阅记录管理模块、图书借阅管理模块、图书归还管理模块、管理员模块并及时记录各个信息的变更。在本系统中,考虑到面向对象的不同,将分为两类用户,普通用户(师生)、超级用户(管理员)。读者想要借阅哪一本图书可以使用图书信息管理模块进行查询(支持模糊搜索),查询到了之后可以通过图书借阅管理模块进行借阅操作,同时,读者借阅信息模块中留下相应的记录。对于超级用户,可以进行各个信息模块的增、删、改、查操作,并进行特殊情况的处理(比如删除违约记录等)。

2.2 任务目标

模块	功能	具体描述
读者信息管理模块	注册	注册个人信息
	修改信息	修改个人信息
	查看信息	查看个人信息
	读者挂失	标记原来的卡废除，补办新卡
	缴纳罚款	缴纳违约金
图书信息管理模块	搜索	搜索图书
	查看信息	查看图书
	分类浏览	按照 tag 查看图书
	热门借阅	查看最近 30 天借阅量最多的书
图书借阅模块	借阅图书	借阅图书
	预约图书	在别的图书馆预约图书
	更新图书信息	更新图书信息
	更新读者信息	更新读者信息，增加借阅量
	在借阅记录管理模块增加该记录	在借阅管理记录中增加记录本条
图书归还模块	归还图书	归还了某本图书
	更新图书信息	原来的图书标记为可借
	更新读者信息	读者标记已经归还图书
	在借阅记录管理模块中删除该条记录	在借阅记录管理中删除该条记录
管理员模块	查看日志表	查看发生的所有事情
	对图书的增、删、改、查	对图书信息的增、删、改、查
	对某借阅记录的增、删、改、查	对图书的增、删、改、查，防止意外情况的发生
	查看数据库信息	可以查看数据库中所有的数据

ER 草图:

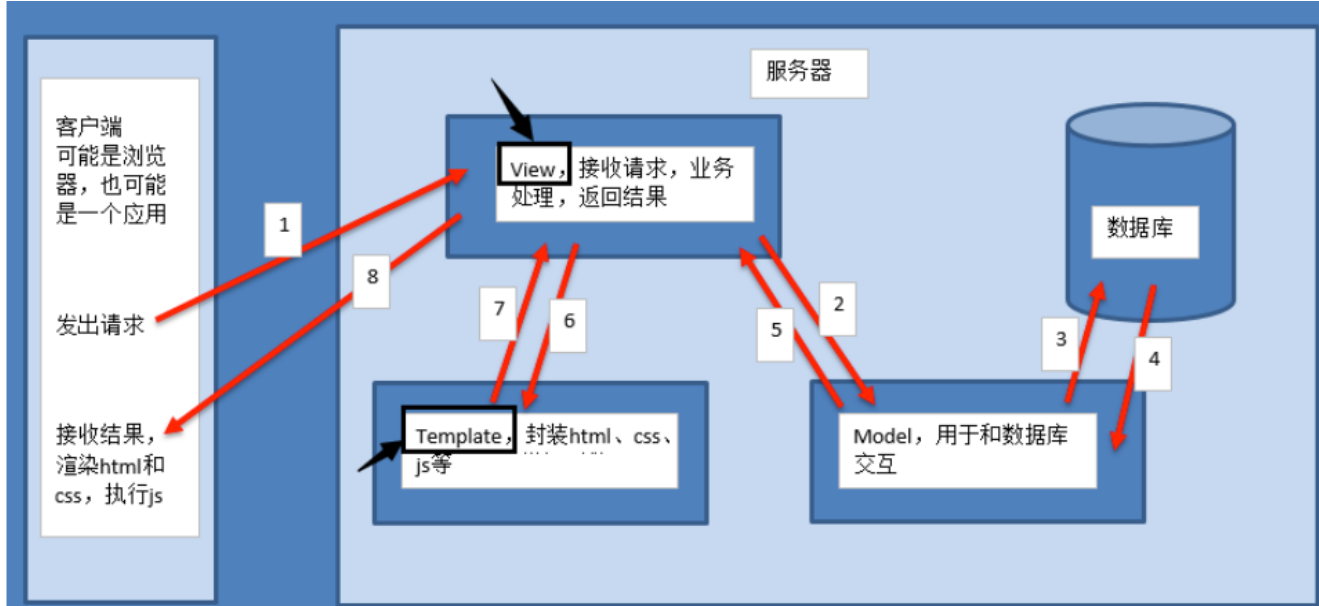
三、系统定义

3.1 系统边界

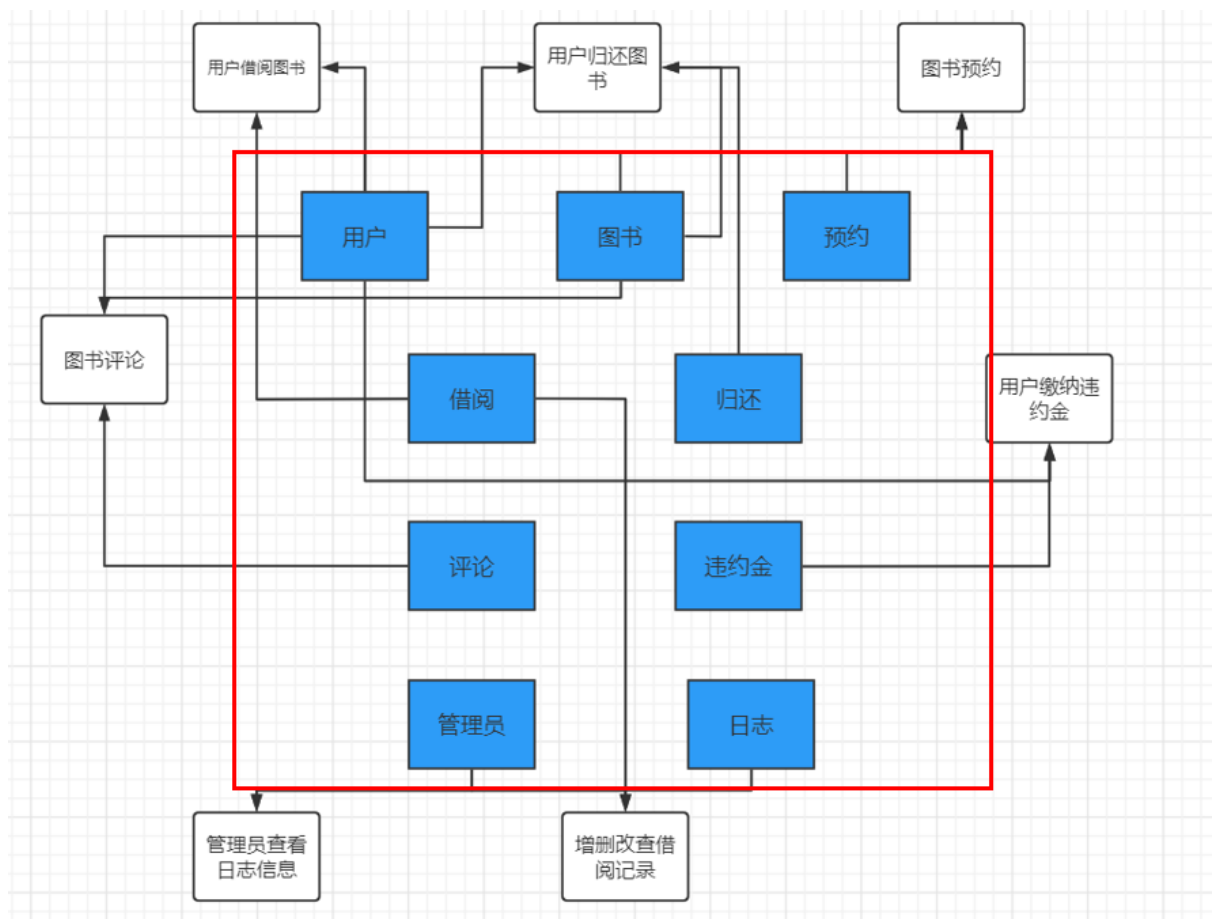
系统边界描述数据库系统和企业信息系统的其他部分的接口，是信息系统内部构成

元素与外部有联系实体之间的信息关系的描述与分割。它并不需要在它们之间划一条物理边界，而只需要弄清它们之间信息输入与输出的分割。

结合 Django 的 MVT 模型与系统边界的定义，我们只需要确定，M 与 V 的边界即可



本数据库系统包括图书信息、普通用户信息、管理员信息、借阅信息、归还信息、违约金信息、日志信息、预约信息、订阅信息等十个实体信息，关系信息包括：用户借阅图书、用户归还图书、用户缴纳违约金、用户预约图书、管理员查看日志、管理员增、删、改、查借阅记录等即为：



3.2 用户视图

普通用户视图:

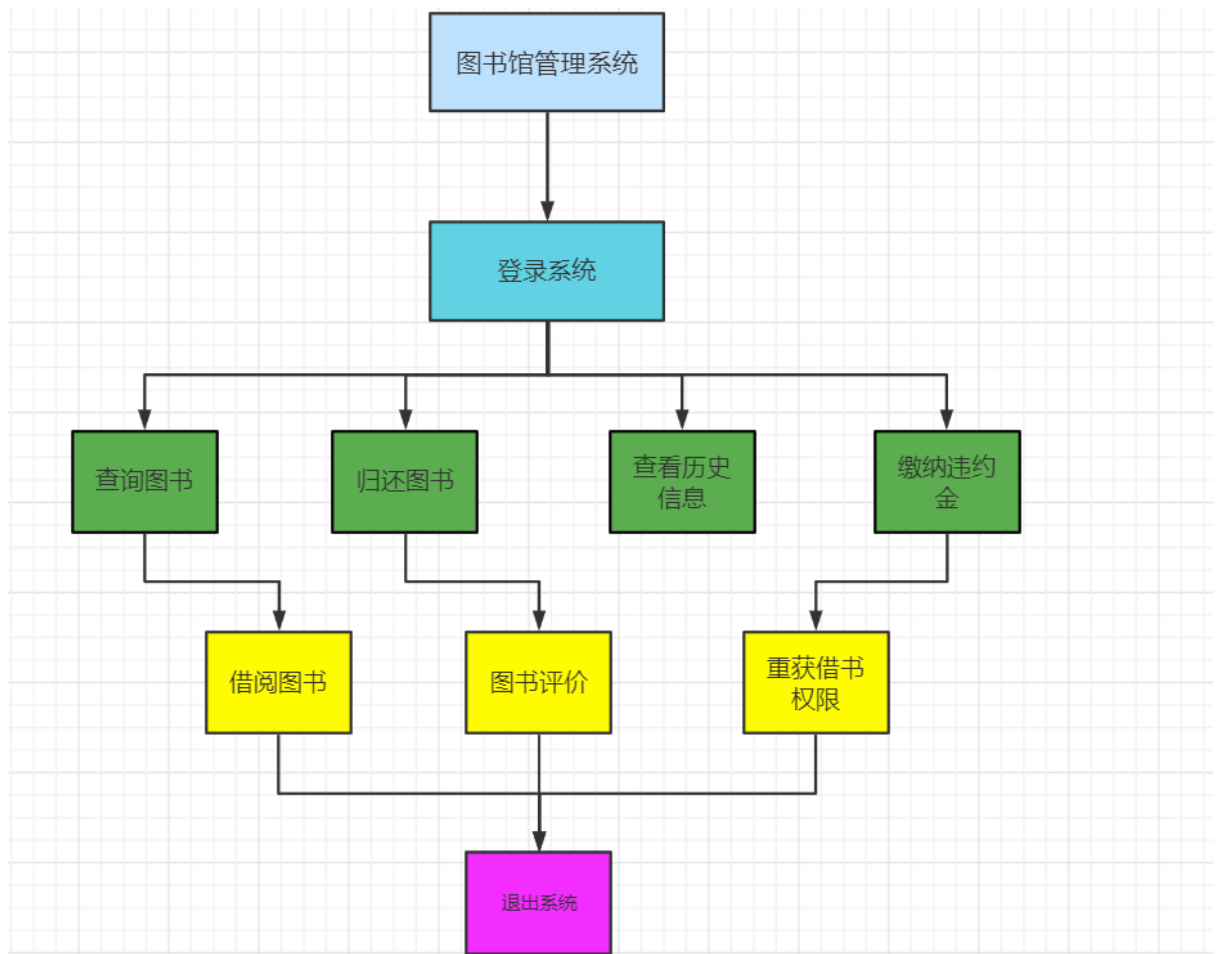
1. 注册管理:
新生注册账户
2. 登录管理:
登录自己账户
3. 查询管理:
查询某一本书是否存在及其存在的位置
4. 借阅管理:
借阅图书
5. 热门借阅管理:
查看近 1 个月借阅量最大的 20 本书
6. 分类浏览管理:
按照 tag 分类浏览图书
7. 预约管理:
预约别的图书馆的图书
8. 荐购管理:
推荐购买图书
9. 评价管理:
评价借阅的图书, 分享收获

10. 查看信息管理：
查看自己的历史借阅
11. 更新信息管理：
更新自己的身份信息，并同步到服务器端
12. 挂失借书卡管理：
挂失补办新卡
13. 挂失图书管理：
挂失图书
14. 违章缴款管理：
缴纳违约金
15. 归还图书管理：
归还图书

管理员视图：

1. 查看日志管理：
查看日志
2. 增加图书管理：
增加新的图书并同步到服务器
3. 删除图书管理：
删除图书
4. 删除违约记录管理：
删除某一条特定的指定违约记录
5. 查找图书管理：
查找图书
6. 查找用户管理：
查找指定的用户
7. 导入信息管理：
对于转入的转专业同学，将其信息导入到服务器
8. 预购信息管理：
是否同意读者请求的购买服务
9. 权限赋予管理：
将某些权限赋给特定的人
10. 权限收回管理：
将赋予的权限

四、需求分析



4.1 用户需求说明

4.1.1 数据需求

1. 用户的个人信息：
学号（主码）、姓名、性别、借阅图书数量、身份证号、入学日期、离校日期、违约金数额
2. 图书信息：
名称、编号（主码）、作者、索书号、条码号、出版日期、馆藏地、书刊状态、评价信息
3. 借阅记录信息：
借阅人学号、借阅书编号、借阅日期
4. 归还记录信息：
借阅人学号、借阅书编号、归还日期
5. 违约金记录信息：

- 借阅人学号、违约金额
- 6. 权限赋予信息：
被赋予的人的学号、权限
- 7. 日志记录信息：
操作的记录、操作的时间、操作的人员
- 8. 推购信息：
推荐人的学号、推荐书的 ISBN 码
- 9. 预约信息：
学号、预约书的 ISBN 码

4.1.2 事务需求

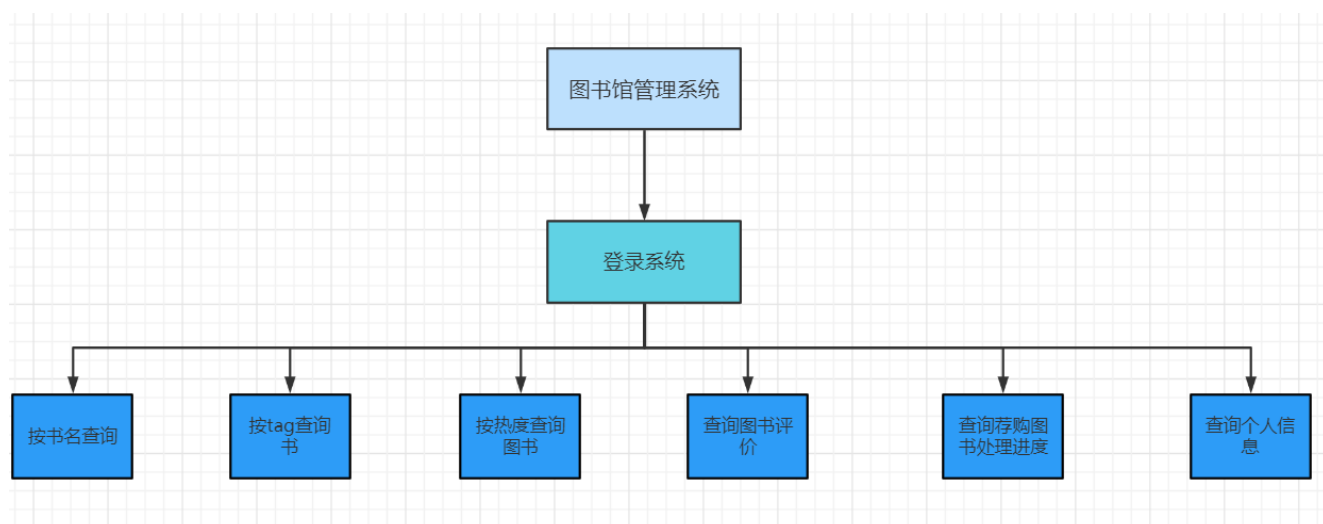
一、数据录入

1. 用户的个人信息的录入
学号（主码）、姓名、性别、借阅图书数量、身份证号、入学日期、离校日期、违约金数额
2. 用户个人信息的修改
学号（主码）、姓名、性别、借阅图书数量、身份证号、入学日期、离校日期、违约金数额
3. 图书信息的录入
名称、编号（主码）、作者、索书号、条码号、出版日期、馆藏地、书刊状态、评价信息
4. 图书信息的修改
学号（主码）、姓名、性别、借阅图书数量、身份证号、入学日期、离校日期、违约金数额
5. 借阅信息的录入
借阅人学号、借阅书编号、借阅日期
6. 归还信息的录入
借阅人学号、借阅书编号、归还日期
7. 预约信息的录入
学号、预约书的 ISBN 码
8. 荐购信息的录入
推荐人的学号、推荐书的 ISBN 码
9. 违约金记录的修改
违约人的学号、修改后的金额
10. 权限赋予的录入
推荐人的学号、推荐书的 ISBN 码

二、数据的更新与删除

1. 学生信息的更新（比如新加入了一些学生）
2. 学生信息的删除
3. 图书信息的增加
4. 图书信息的更改
5. 图书信息的删除
6. 图书信息的删除
7. 借阅信息的增加
8. 借阅信息的减少
9. 借阅信息的删除
10. 归还信息的增加
11. 归还信息的减少
12. 归还信息的删除
13. 权限的赋予
14. 权限的删除

三、数据的搜索与查看



4.2 系统需求说明

一个好的系统满足可靠性、健壮性、时间复杂度与空间复杂度优秀，该图书馆管理系统理论上可以满足十万级别的图书信息、个人信息存储，并且在高峰期也有优秀的查询、借阅处理速度，同时有效的处理各种异常（使用 `try...except`），健壮性良好。

1. 初始数据库大小

图书数量：10000 本，其中不同 tag 的书数量不尽相同

普通用户数量：4000 人

超级用户数量：1 人

2. 网络需求

必须能够支持至少 100 名用户同时查询、进行借阅操作，需要考虑这么大数量并发访问的许可需求

好的网络需求直接决定用户体验

3. 性能

高峰期：每天的上午、下午

- (1) 单个记录查询时间少于 1 秒，高峰期少于 5 秒
- (2) 多个记录查询时间少于 5 秒，高峰期少于 10 秒
- (3) 更新/保存记录时间少于 1 秒，高峰期少于 5 秒

4. 安全性

数据库必须有口令保护（比如 `github` 上的 `token`）

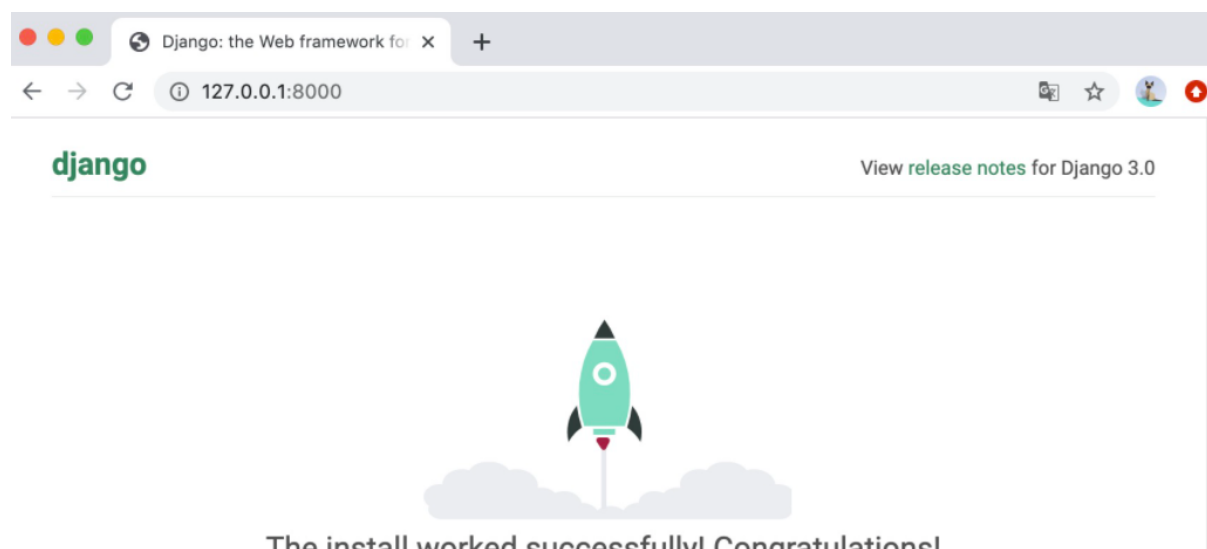
不同的用户拥有不同的访问权限，不能越级访问

5. 备份和恢复

每天 24 点备份

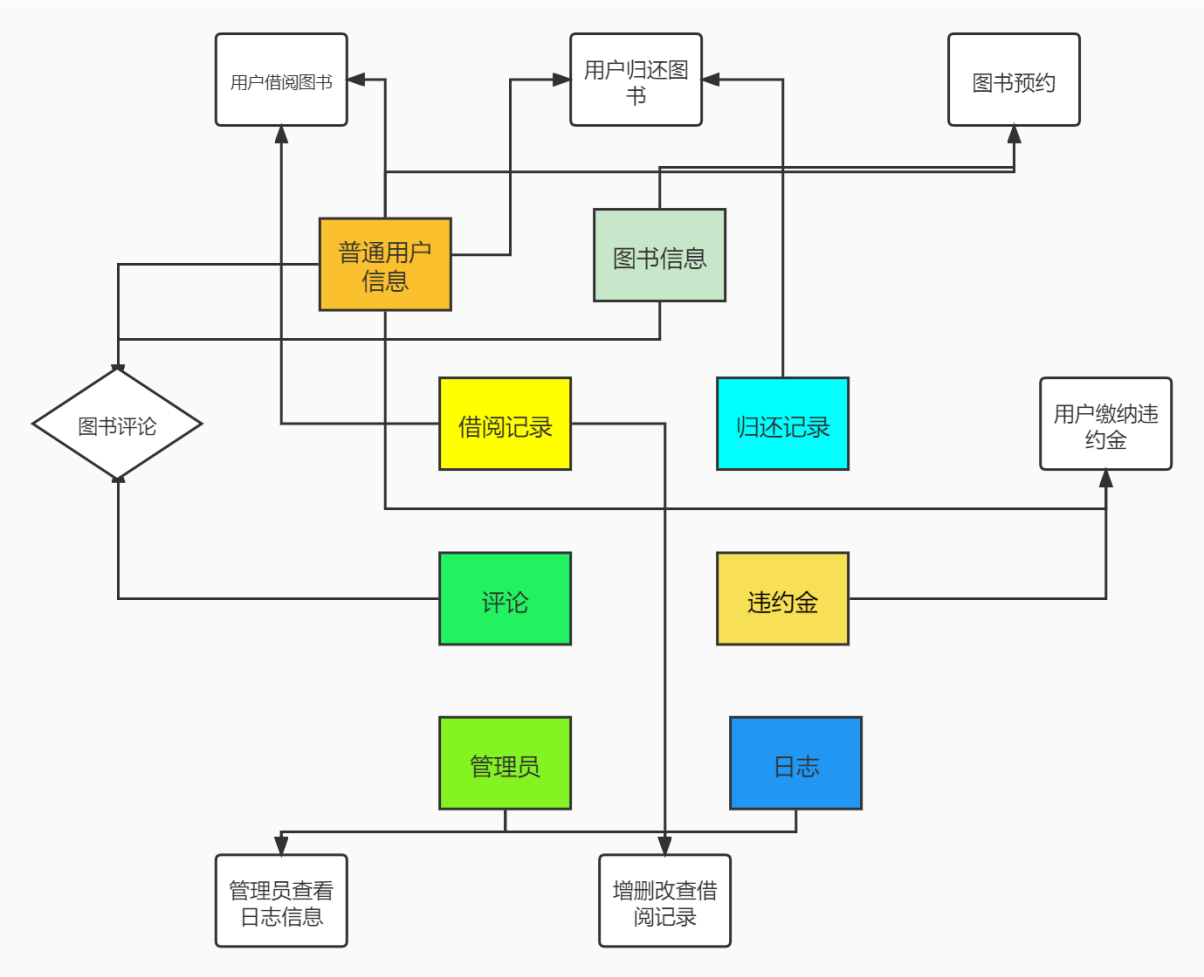
6. 界面友好性

使用 web 端，界面良好



五、数据库逻辑设计

5.1 ER 图



5.2 数据字典

1. 普通用户实体表（如学生）：

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
name	varchar	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
deparment	varchar	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
gender	varchar	1		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
my_book	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
book_borrowed	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
missed_books	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
enrollment_date	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>		
leave_date	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Liquidated_damages	int	11		<input type="checkbox"/>	<input type="checkbox"/>		

```

1. create table general_user
2. (
3.   id int PRIMARY key,
4.   name varchar(20) not null,
5.   deparment varchar(20) not null,
6.   gender varchar(1) not null,
7.   my_book varchar(20),
8.   book_borrowed int not null DEFAULT 0,
9.   missed_books varchar(20),
10.  enrollment_date DATE not null,
11.  leave_date date not null ,
12.  Liquidated_damages int
13.);

```

2. 图书实体表:

名	类型	长度	小数点	不是 null	虚拟	键	注释
id	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	
name	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
author	varchar	20		<input type="checkbox"/>	<input type="checkbox"/>		
ISBN	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
published_date	date			<input checked="" type="checkbox"/>	<input type="checkbox"/>		
collected_area	varchar	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
is_borrowed	int	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>		
score	int	11		<input type="checkbox"/>	<input type="checkbox"/>		

```

1. create table book
2. (
3.   id int PRIMARY key,
4.   name varchar(20),
5.   author varchar(20),
6.   ISBN int not null,
7.   published_date date not null,
8.   collected_area varchar(20) not null,
9.   is_borrowed int not null default 0,
10.  score int

```

```
11.);
```

3. 借阅记录实体表:

名	类型	长度	小数点
general_user_id	int	11	
book_id	int	11	
borrow_date	date		

```
1. create table borrow_data
2. (
3.   general_user_id int,
4.   book_id int,
5.   borrow_date date
6. );
```

4. 归还记录实体表:

名	类型	长度	小数点
general_user_id	int	11	
book_id	int	11	
return_date	date		

```
1. create table return_data
2. (
3.   general_user_id int,
4.   book_id int,
5.   return_date date
6. );
```

5. Log 记录

名	类型	长度	小数点	不是 null
person_id	int	11		<input checked="" type="checkbox"/>
did_what	varchar	100		<input checked="" type="checkbox"/>
_date	date			<input checked="" type="checkbox"/>

```

1. create table log_data
2. (
3.   person_id int not null,
4.   did_what varchar(100) not null,
5.   _date date not null
6. )

```

5.3 关系表

六、数据库物理设计

6.1 索引

6.2 视图

本节可选。

6.3 安全机制

系统安全、数据安全

6.4 其他

本节可选：引入受控冗余等。

七、应用程序设计

7.1 功能模块

7.2 界面设计

可抓几幅画面介绍。

7.3 事务设计

关键或复杂的事务操作的设计，可附相应 SQL 语句进行分析介绍。

八、测试和运行

介绍系统的测试情况，可提供几个测试用例。

可选：生成大数据量的仿真数据，进行性能测试的情况。

九、总结

经验体会；系统的特色、不足（可继续完善的方面）。

附. 参考文献

主要的参考资料、网址。