

图书管理系统设计文档

小组成员：王嘉康 韩璟瑄 陈灯月



2021-5-18

# 绪论

图书管理系统的主要功能是实现图书馆图书的借阅和归还的管理自动化，图书新增及销毁的及时化，用户及图书信息的更新，围绕这些主要功能，本系统涉及到以下核心功能：图书管理，用户管理。  
 该系统设计的主要目标是．设计一个图书馆管理系统，该系统主要功能分为图书管理、用户管理两大部分。在图书管理模块中要求能对书进行增删改查等日常管理功能；在用户管理部分能完成对角色的增删改查。

# 系统设计

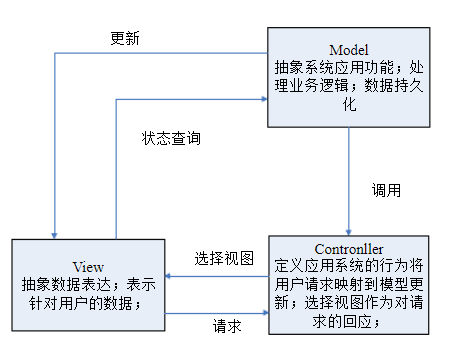
## 2.1 指导原则

（1)利用软件开发现有软硬件环境，及先进的管理系统开发方案，从而到充分利用现在资源提高系统开发水平和应用效果的目的。

（2）系统应该符合软件工程开发的理论，开发方法等开发依据。  
 （3）系统应满足图书馆管理工作的需要，并达到操作过程中的直观，方便，实用，安全，准确等要求。  
 （4）系统应具备数据库维护功能，及时根据用户需求进行数据库的各种操作。  
 （5）系统采用原型，实用模块化程序设计方法，便于系统功能的各种组合和修改，以及系统的测试与维护。  
 （6）图书馆管理系统的设计应适合校园的发展。

## 2.2 设计模式

设计模式是面向对象的程序设计人员用来解决编程问题的一种形式化表示。本系统开发采用目前一种广泛流行的软件设计模式 MVC (Model—view—Controller)应用程序结构被用来分析分布式应用程序的特征。这种抽象结构能  
有助于将应用程序分割成若干逻辑部件，使程序设计变得更加容易。把一个应用的输入、处理、输出流程按照Model、View、Controller的方式进行分离，这样一个应用被分成三个层一一模型层、视图层、控制层。



MVC模式的出现，很好的解决了传统开发WEB应用方式中存在的问题。M代表模型(Model)，包含完成任务所需要的所有的行为和数据；v代表视图(View)界面，现实模型提供的数据：c代表控制器（Controller），它将模型映射到界面中，处理用户的输入并相应请求。其模型的关系如图所示。在MVC模型中，三层各尽其职、相互独立，各层内部的改变不会影响到其它层，从而降低了数据表达、数据描述和应该操作的耦合度，也能更好的实现开发中的分工，加速工程进度。

## 2.3 图书管理系统的功能范围

图书管理系统包含三个模块分别是：系统登录、图书管理、用户管理。系统登录模块一般用户和管理员都必须登录才能进入系统，用户登录时在后台判断用户的权限类型，分为普通用户和管理员，普通用户可以对书籍及个人信息的增删改查，管理员可以对图书和所有人的信息进行操作。

## 2.4 图书管理系统的功能细节描述

### 2.4.1 用户登录

|  |  |
| --- | --- |
| **参考界面** |  |
| **流程描述** |  |

### 2.4.2 用户注册

|  |  |
| --- | --- |
| **参考界面** |  |
| **描述** | 邮箱不允许重复；  所有人注册都只能是普通用户，而管理员只有一开始内置的一个。  只有管理员才可以修改别人的用户权限。 |

### 2.4.3 图书列表页

|  |  |
| --- | --- |
| **参考界面** | 用户界面：    管理员页面： |
| **描述** | * 用户操作描述： * 图书查看：查看详细信息。 * 图书购买：点击购买添加至购物车。 * 个人中心：查看个人资料。 * 清空购物车：清空已选中书籍和总价格。 * 确认购买：点击对话框的确认即可完成购买。 * 管理员操作描述： * 图书编辑：点击进入图书编辑页面。 * 图书删除：删除指定书籍。 * 图书添加：点击进入图书添加页面。 * 个人中心：可对所有用户进行管理。 |

### 2.4.4 添加图书页

|  |  |
| --- | --- |
| **图例** |  |
| **描述** | 书名、作者、内容均为文本字段；  点击添加按钮后完成新增图书，可返回图书列表页进行查看；  点击选择上传文件按钮，可从本地选择需要上传的文件。 |

### 2.4.5 编辑图书页

|  |  |
| --- | --- |
| **图例** |  |
| **描述** | 书名、作者、内容均为文本字段。  若本书已有文件，则可对应下载文件和删除文件。  若本书没有对应文件，则可上传本地文件。 |

**2.4.6 个人中心页**

|  |  |
| --- | --- |
| **图例** | 图1    图2    图3 |
| **描述** | 权限只有ROLE\_ADMIN和ROLE\_USER。  图1为ROLE\_USER的个人信息页面，密码使用强哈希加密，双击密码对应位置即可修改（如图3），点击修改按钮提交至服务器。  图2为ROLE\_ADMIN的个人信息页面，可以查看所有的人信息，可修改所有人的密码和角色，也可添加新用户。 |

1. **关键技术与实现**

**3.1 JWT:**

在spring security登录成功之后，跳转到该函数，生成JWT，返回前端。



之后的前端的每次访问，都要经过下面的filter，来判断jwt是否正确或者过期。



**3.2 REST:**

本项目rest接口设计如下：

**3.2.1 ROLE\_UESR权限的crud接口**

@RequestMapping(value = "/user/user", method = RequestMethod.***PUT***)

@RequestMapping(value = "/user/user", method = RequestMethod.***GET***)

@RequestMapping(value = "/user/user", method = RequestMethod.***DELETE***)

分别对应修改、获取和删除个人信息。

**3.2.2 ROLE\_ADMIN权限的crud接口**

@RequestMapping(value = "/users", method = RequestMethod.***GET***)

@RequestMapping(value = "/users", method = RequestMethod.***POST***)

@RequestMapping(value = "/user/{email}", method = RequestMethod.***DELETE***)

@RequestMapping(value = "/user/{email}", method = RequestMethod.***PUT***)

分别对应查询所有用户信息、添加新的用户、删除指定用户、修改指定用户。

**3.2.3 book的crud接口**

@RequestMapping(value = {"/books"}, method = RequestMethod.***POST***)

@RequestMapping(value = "/book/{bid}", method = RequestMethod.***DELETE***)

@RequestMapping(value = "/book/{bid}", method = RequestMethod.***PUT***)

@RequestMapping(value = "/book/{bid}/upload", method = RequestMethod.***POST***)

@RequestMapping(value = "/book/{bid}/delete", method = RequestMethod.***DELETE***)

@RequestMapping(value = "/books", method = RequestMethod.***GET***)

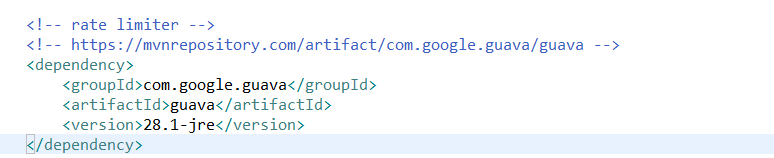
@RequestMapping(value = "/books/{bid}", method = RequestMethod.***GET***)

分别对应添加新的书、删除指定书、更新指定书、指定书上传文件、指定书删除文件、获取所有书籍信息、获取指定书的信息。

**3.3 RATE LIMITING:**

使用Guava

在pom文件中引用



创建自定义的限流





在所有路由下引用注解

