*Java框架编程课程设计报告*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *作业题目* | 图书管理系统 | | | |
| *专业及班级* | 计科二班 | | *指导教师* |  |
| *姓名* | A | | *学号* |  |
| *组中其他成员姓名* | | B | | |
| *组员分工* | | A：spring boot后端，mysql数据库设计与搭建  B：react前端，jwt登录注册权限认证，Restful重构 | | |
| 1. **实现的主要系统功能（按评分标准书写）**   1． 数据库  保存图书，用户，借阅记录等信息  2． 用户操作  1. 登录  2. 注册  3．浏览主页信息  4. 浏览图书信息  5. 借阅图书  6. 查看个人借阅记录  7. 归还图书  3. 管理员操作（包含普通用户操作）  1. 查看图书与订单统计数据  2. 查看图书库存情况  3. 增加，删除，下架，上架，查询图书  4. 删除，封禁，解封用户  5. 删除，强制归还借书记录  6. 查看Druid数据源后台  4. 额外功能  1. 非法输入检测与防止  2. 访问与操作权限认证，伪造请求防御  3. 数据库操作监控，sql注入防御  6. 多线程运算，多用户并行操作优化  7. 数据库引索优化与列字段精简  8. 磁盘读写与内存交互速度优化  9. jar包内嵌tomcat一键部署运行，便于虚拟机部署  10. jwt 认证实现无状态用户鉴权  11． 无状态 (stateless) REST API 设计，便于分布式集群部署 | | | | |
| ***二．*类的设计**  **1. 数据库表结构**    books表存放图书信息，字段从上到下分别对应：图书编号，书名，作者，出版社，价格，总库存量，已借出量，现有库存  libusers表存放用户信息，字段从上到下分别对应：用户编号，用户名，密码，用户权限  orders表存放借阅记录信息，字段从上到下分别对应：记录ID，该记录中被借阅的图书编号，该记录创建者的用户编号，借阅时间，归还时间，是否被归还的状态标记  orders表含有两个外键约束，作为books表和libusers表的字表，分别约束bookID（对应books表中图书编号）和userId（对应libusers表中用户编号） | | | | |
| **2． Java代码结构**     1. domain包保存项目所需的JavaBean，分别是图书，用户，和借阅记录 2. repository对应MVC结构项目中的dao层，但相比传统dao层写入了sql语句，整合了Mapper。三个类文件分别与三个JavaBean对应。 3. services包对应MVC结构项目中的service层，主要实现业务逻辑，由controller调用并返回处理结果。三个类文件分别与三个JavaBean对应。 4. controller包对应MVC结构项目的controller，负责接收前端的请求，并返回service的处理结果。在本次作业中，这些controller用于提供Restful Api，接收前端的json文件，并返回json结果。实现前后端分离。三个类文件分别与三个JavaBean对应。 5. application.properties文件为Spring Boot项目配置文件，主要用于写入数据库配置信息，mybatis配置信息，druid数据源配置信息，热部署配置信息 6. LibraryManagerApplication类为项目启动类，包含main函数。另有：对不同路径的访问权限的设置信息，分离普通用户与管理员操作。 7. RuntimeConstants类用于设置运行时常量，此处主要用于设定密钥 8. Filter包中有管理员权限验证与过滤器，另有大致相同的普通用户权限验证与过滤器。使用JWT token实现加密解密认证等。保证安全。 9. AuthException类用于封装用户认证过程中出现的错误信息   **·三、算法设计（请用流程图说明程序流程）**   1. **操作**   ***Diagram  Description automatically generated*** | | | | |
| 1. **运行**   ***Diagram  Description automatically generated*** | | | | |
| **四、程序纠错过程（对从程序初次运行到运行成功过程中，所显示的错误信息与改正使用的方法）**  1. 前端向后端请求获得 HTTP 状态码 401 或 403  未在header 中提供 JWT 令牌，或令牌无效。可能是没有登录、或者后端业务逻辑错误（例如没有添加或者添加了错误的 claim）导致签发了无法被认证的 JWT  2. 前端向后端请求获得 HTTP 状态码 5xx  服务器内部错误，视返回的 response 中的具体 exception 而定。例如 SQLException 可能是 repository 层与数据库操作的问题，WeakKeyException 可能是设定的加密密钥强度太低，等等   1. 初稿使用Spring Security，导致postman测试时返回401.和登录界面的html代码。初步怀疑是csrf功能未关闭导致。关闭csrf后问题仍未解决。查阅文档得知此处需要重写Spring Security默认登录接口相关，过于复杂。为了引入JWT，放弃了Spring Security。   4. 图书或者用户的删除操作，有时会发生SQLException。分析错误信息得知，因为借阅记录表存在外键约束，不能随意对父表执行删除。否则会导致子表中出现大量null。因此我们在删除操作中加了安全保护。之后新增了上架下架，封禁解封等功能，实现类似删除的效果的同时，不在物理上消除数据库数据。 | | | | |
| **五、程序运行截图并对截图实现的功能进行简要说明。** | | | | |
| 未登录时的主页 | | | | |
| 登录和注册页面 | | | | |
| 登录后出现借阅按钮供用户操作 | | | | |
| 普通用户可以查看自己的借阅记录 | | | | |
| 管理员可以选择下架图书 | | | | |
| 管理员可以管理借阅记录 | | | | |
| 管理员可以查看所有用户并对用户做出封禁或删除的操作 | | | | |
| 管理员可以登录 druid 后台监控是否有异常操作 | | | | |
| **六、程序主要代码（可只提供程序运行的关键代码）。** | | | | |
| Libuser的JavaBean（省略setter和getter） | | | | |
| Libuser的JavaBean（省略setter和getter） | | | | |
| Order的JavaBean（省略setter和getter） | | | | |
| *在Order中设计的时间格式化函数，并与构造函数，settergetter函数结合。这些函数将被mybatis框架在执行sql语句时自动调用，讲借阅记录中包含的时间信息全自动转化为mysql数据库与JavaDate通用的格式。避免出现时间格式错误导致的乱码。* | | | | |
| *映射控制相关代码。借阅记录信息在数据库中，仅通过外键约束保存借阅的图书编号与用户编号信息。而JavaBean需要全套的Book与Libuser信息，此时就无法使用自动映射，需要手动配置JavaBean中成员变量和mysql列字段对应关系。并在涉及ID的部分嵌套一对一查询，实现多表联查。* | | | | |
| *多线程并行流式操作计算统计数据。* | | | | |
| *Controller举例，@RestController用于设置响应格式为纯json，符合Restful Api接口标准*  *以下主要展示除去基础增删改查外的功能*      *图书借阅与归还操作。@PutMapping用于设置前端请求格式与地址*  *@ResponseStatus用于设置响应的状态码。@RestController用于设置响应格式为纯json* | | | | |
| *图书上架与下架* | | | | |
| *用户封禁与解封* | | | | |
| *获取当前用户的借阅记录* | | | | |
| *用户登录注册与权限验证。* | | | | |
| *管理员权限验证与过滤器，另有大致相同的普通用户权限验证与过滤器。*    *对各路径的访问权限进行设置的代码。* | | | | |
| *16.*    *Druid数据池配置* | | | | |
| *17.*    *秘钥* | | | | |
| *18.*    *Druid后台权限认证，拦截切面aop配置，防sql注入安全设置*  *19.*  *Text  Description automatically generated*  *前端用户认证请求处理部分*  *20.*  *Text  Description automatically generated*  *前端一个表格行组件示例（管理员管理借阅记录的行）*  *21.*  *Text  Description automatically generated*  *前端的一个表格示例（管理管理借阅记录表格）* | | | | |
| ***七、实验心得。***  前端部分：第一次在一个比较完整的项目中使用了一个前端框架（React），体验了一把使用 Reactive MVVM 的快感。也是第一次写比较正经的和 REST API 交互的项目，深切的感受到了这种架构的优劣之处：例如用户验证必须无状态，这对传统的有状态的用户验证方案（例如 Spring Security）其实有很大的不同。前端的其实到处还有比较多的比较不得当的实现的，例如因为自己实在是对 Javascript 了解不精，代码重用远远没有到令人满意的水平。但是最终得到了一个还算能用的前端，而且算是在实践中对 MVVM 的理解加深了不少，算是一些不小的收获吧。  后端部分：本次项目作业是对我本学期以来自学的大量java web知识的大串联。从传统的jsp+servlet,到ssm，到spring boot，一步步稳扎稳打走下来的成就感让我十分满足。同时也让我不断感慨新技术新框架的优势，这一系列知识足够我用到毕业也不过时。从本次作业中我也初步掌握了独立管理和开发维护大型软件项目的能力和对大量代码的差错纠错能力，这会是一次宝贵的实践经验。  最后，感谢B同学指导我学习Rest Api等先进技术标准。感谢老师的悉心指导！ | | | | |