**实 验 报 告 一**

**课程名称：Spring Boot应用技术开发**    
**实验项目 基于Spring Boot开发应用**

**实验仪器：**

**学 院： 计算机学院**

**专 业： 软件工程**

**班级姓名： 软工1803/石荣兴、孙怡轩、田力涛**

**学 号： 2018011377**

**2018011379**

**2018011381**

**日 期： 2021-04-19**

**指导教师： 郝保水、陈军峰**

**成 绩：**

# 实验目的

基于MVC模式，使用Java、Spring Boot、Vue、HTML、Javscript和MySQL协同实现一个小型Java Web应用系统。

1. 正确理解和使用Sprint Boot技术以及应用开发的基本流程
2. 能够根据业务需求设计出数据库表结构，并基于MyBatis构建数据访问层
3. 正确地配置和使用MySQL数据库
4. 在Spring Boot中正确地理解和使用系统的分层设计原则
5. 能够自主进行资料检索和学习，通过对现有技术和方案的对比分析确定最终的实验方案
6. 通过资料查找和学习了解技术进展
7. 能够将新知识和技术应用于实验过程

# 实验要求

1. 项目应用背景：

构建一个可以直接应用于校园的二手交易市场，允许同学在线发布二手商品；在后台审核通过之后，可以在线购买下单，锁定商品。同时约定线下交易的时间和地点，通过线下完成整体的交易环节。同时可以针对购买的商品，可以在线进行评论，对于商品和买家进行评论反馈。

该系统由二手交易和后台管理审核系统两个部分组成。后台管理审核，主要是负责对于用户人员、线上发布商品的审核管理功能。 二手交易系统主要是提供在线商品的浏览和搜索，并提供用户下单购买的主要流程；同时提供用户注册，提供用户个人信息和购买历史记录等信息。

**其中必选满足的功能点：**

* 用户的注册和登录流程（注册要求有随机基于图片的验证码）



用户信息包括：id、姓名、手机号、邮箱、城市、性别、银行账号（16数字）

待管理员审核之后，方可正式生效登录使用。

商家注册过程，需要提供营业执照（图片上传）、银行账号、姓名、性别、手机号、身份证（图片上传），审核成功之后，方可正式使用。

区分不同的用户角色（普通用户/商家/管理员）

普通用户： 浏览、购买商品、已退货/在途退货商品

商家： 发布商品、浏览、下架商品

管理员： 审核商品、审核用户注册、管理用户列表（修改/删除/查看）

* 商家在线发布/下架商品（多张图片、价格和数量），以商家的XXX店展示商品。

支持商家的店铺维度展示商品、商品的库存、历史销量和购买者评价等

发布商品信息：商品名称、类别、价格（原价/折扣）、尺寸大小、照片、使用说明、是否允许议价、商品数量、新旧程度（全新、九成新、八成新等）。

商品支持购买后的评价（基于文字和五星级的评价），评价可以商品查看。

购买用户可以对商家的服务态度进行评价（

已经发布/已经出售/已经下架商品的浏览

商家钱包中，在商品售卖之后24小时，方可收到货款；用户下单的24小时之内，此金额放在系统的中间账户中，24小时交易成功后，需要根据商家等级进行收取交易费用；

商家等级分为5级，由管理员负责设置指定。

|  |  |  |
| --- | --- | --- |
| 商家等级 | 费率 | 备注 |
| 1 | 0.1% |  |
| 2 | 0.2% |  |
| 3 | 0.5% |  |
| 4 | 0.75% |  |
| 5 | 1% |  |

* 登录用户可以使用购物车，支持一键下单购物车的商品(后台无需对接支付接口)

购物车中可以暂存用户的商品列表，支持多个商家的商品选购

一键下单（无需对接支付接口），用户账号会被扣除对应的商品货款。

支持用户在24小时内退货，在用户发起退货之后，在商家审核之后方可生效；发起退货申请时间，超过购买时间24小时之后不能退货。

* 用户有钱包功能和积分功能

管理员可以针对特定用户进行充值操作（即为特定用户增加钱包的钱数）

用户购买商品之后，1元消费额即可转换为1个积分。

* 管理员需要审核用户的注册、商家发布商品；收取交易的费用，根据商家的满意度和交易额，动态调整商家的等级
* 商城需要具备基于商品名称的搜索功能，可以查出多个商家的商品列表，基于价格、销售量和好评率进行排序展示，用户可以设定对应的排序规则
* 商家可以对于用户的购买/退货商品进行评价，评价会转换为购买者的商家好评率。

**可选功能：**

* 商城首页的轮播图（不同商家的推荐位以及广告宣传商品）
* 针对不良商家的限时封禁功能和下架功能等惩罚功能

限时封禁： 不允许新发布商品多长时间

下架： 全部商品下架，店铺暂时关闭

* 支持恶意购买者的拉黑功能，被商家/平台拉黑的购买者，将不能在特定商家和整个平台进行购买商品。

技术设计上的参考建议（可以不参考）：

User表（普通用户和管理员）、商品表，商品状态： 创建、审核、发布和锁定、交易完成、退货、照片/文字描述/价格等、用户购买记录、权限信息、用户评价、商品评价信息。用户积分、用户信息变更记录表。

# 项目管理

3.1 人员功能

人员分配：

前端：孙怡轩，田力涛

Vue，JavaScript：孙怡轩

CSS：田力涛

后端：石荣兴

3.2 项目开发流程

例如： 需求定义-需求分析-系统架构设计-技术选型-系统设计-任务拆分（估时）- 编码-联合调试-测试-发布等阶段。

**3.2.1需求分析阶段**

由于此次试验老师有给出明确的项目需求，所以我们主要对题目中给出的功能需求进行了小幅度的分析以及调整，更多的需求分析则放在了UI设计以及细节实现上。我们小组通过在网上调研淘宝，京东，天猫等现有的电商平台的UI设计以及功能细节进行对比分析，得到了一个比较符合我们小组情况的项目结构。

**3.2.2系统架构设计**

前台的电商项目系统结构主要参考了现有的网络电商平台的可见设计。而后台管理员系统的架构设计则是主要参考了学校教务系统的信息查询模块的设计。将表的筛选权交给管理员用户，使其可以对用户，商店，商品等进行自定义检索以及管理。

**3.2.3技术选型**

在技术选型上，由于我们小组使用的电脑为两个Windows，一个苹果，所以最终我们决定在硬件的选择上以Windows系统为主。而在软件的选型上，编写工具我们统一选择了idea这块开发工具，因为其不仅可以进行后端编写，同时还可以进行前端的编写及设计。而在项目的编写框架上，我们使用的方案就是题目中给出的springboot+mybatis+vue的组合。同时在对图片进行存储时我们采用的是远程图床的存储方式，主要利用的平台时腾讯云的cos对象存储技术。

**3.2.4系统设计**

在系统设计上，我们主要分为了两大模块，一个是管理员模块，一个是前台商城模块。二者分别对应了用户和管理员两种用户角色。在最终的后端接口以及前端代码编写上进行统一编码，相互照应。

**3.2.5任务拆分**

在对任务进行拆分时，我们主要根据人员拆分成为了两大阶段——后端编写阶段，和前端编写阶段。具体的时间分配如下：3月30号前完成后端项目的编写【包含数据库设计，后端逻辑代码编写等工作】。4月到6月间进行前端项目的编写【包含vue代码的编写，js代码的编写，css代码的设计与编写】。如上的安排都在后面的实践中如期完成。

**3.2.6编码**

在实行编码时我们小组主要采用阶段式编码，即先编写后端接口，在完成接口后再让前端人员根据接口文档进行前端编写。

**3.2.7联合调试**

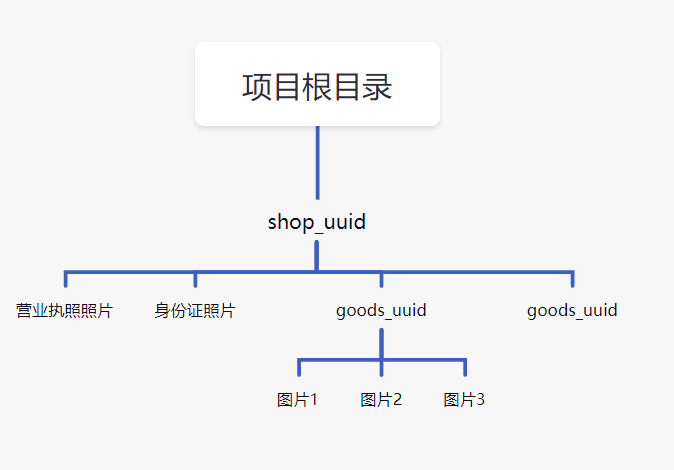
在联合调试阶段主要是后端人员向前端人员解释接口文档以及，前端对接口进行反馈，让后端人员按照前端人员需求对接口进行调整变更。

**3.2.8发布**

该阶段主要是将编写完成的系统上线服务器，并成功运行。经测试后成功发布。

3.3 技术关键节点

**数据存储结构设计：**



技术栈的选取

**技术栈选择**

**后端**

**1.springboot**

**2.mybatis**

**3.mysql**

**4.maven**

**前端**

**1.vue**

**2.element**

**3.css**

**4.axios**

# 实验过程

1. 数据库设计（MySQL）
2. 创建数据库，名为同组人员任一同学的学号。

数据库名为： 2018011377

1. 系统用到的数据表设计

这里对于数据库表的设计，我们小组采用的是腾讯文档进行编写的，所以这里给出具体文档的连接地址：

【腾讯文档】软件开发实践-二手交易平台

<https://docs.qq.com/sheet/DTXROeXZyU0R6b0Zq>

1. 系统功能说明。

本次实验实现的系统具体功能可分为以下几点：

**管理员模块**

1.对用户进行检索，管理，审批。

2.对商品进行检索，审批。

3.对店铺进行检索，管理。

4.对订单进行检索

**商城模块**

普通用户

1.对商品进行浏览

2.对店铺进行访问

3.对商品进行购买，添加到购物车操作

4.对已购买的商品进行评价，评论，退货

5.进行钱包充值

6.对商品，或店铺进行检索按需检索

7.查询已购买的商品订单

8.进行用户注册

9.更新密码，地址

商家用户

1.进行商家注册

2.进行商品的发布

3.对商铺进行管理，改名等操作

4.对商品进行调整【数量，价格，图片，描述等等】

5.对买家进行评论回复

6.更新密码

1. 程序资源（清单）说明。以表格的方式说明各程序文件的作用，github的代码地址。

后端Github地址：<https://github.com/srx-2000/transaction>

前端Github地址：<https://github.com/huoyanlieniao/vue-shop/tree/master>

**后端资源清单**

**由于后端总文件数达到了64个之多，所以这里对相应的资源进行了分类，并对分类进行说明。**

|  |  |
| --- | --- |
| **Controller类** | **主要负责对相应的接口进行转发以及service层的调用** |
| **Entities类** | **实体类，主要是对表中数据进行实体映射，同时其中还包含了一个DTO（对象传输类）主要用于对象在web间的传输。** |
| **Enum类** | **枚举类，主要作用就是对项目中的各种常用到的实例进行列举。如：返回码，商店等级【以及对应的税率】，商品种类等。** |
| **Exception类** | **异常类，主要作用就是自定义了一些在后端可能会出现问题的异常：如反射异常，存储路径异常等等** |
| **Main.Mapper类** | **映射类，主要作用就是定义相应的mapper接口，供mybatis进行映射。** |
| **Service类** | **服务层，主要就是对从controller层映射过来的逻辑业务进行处理，以及对一些事务操作进行处理。** |
| **Util类** | **主要用来存放一些常用的工具类：如Md5加密工具，图床搭建工具，分页器工具，文件存储工具，验证码工具等等。** |
| **SrxApplication** | **Springboot项目的启动类** |
| **Resources.Mapper** | **Sql语句编写的xml映射文件，里面主要存放了项目中可能会用到的sql语句。与Main.Mapper相映射。** |
| **application.yml** | **该文件主要用来对项目端口，数据库连接，文件传输大小，mybatis映射扫描等配置属性进行配置。** |

**前端资源清单**

|  |  |
| --- | --- |
| **Api文件夹** | **与后端的接口进行数据传递。按照业务逻辑，分为deal、good、root、shop、shopCar、user、wallet，七个接口使用axios的get与post方法。默认baseUrl为”/api”** |
| **Assets文件夹** | **css所需图片** |
| **Router文件夹** | **vue路由，里面又两种，不需要验证的页面包括，首页、商品页、登录页、注册页、商家注册页、商品详情页、商店页。需要验证登录的页面包括购物车页、订单页、订单处理页、用户详情页、商家处理页、管理员处理页。** |
| **App.vue** | **主页面，通过router-link来进行页面的切换。** |
| **Mian.ts** | **Vue项目main文件，创建vue并引用element、router等工具。配置全局拦截器用语拦截未登录用户。** |
| **Vue.config.js** | **用于解决前端跨域问题。默认baseurl为“/api”** |
| **Business.vue** | **商家页面。商家的所有操作均在此页面。管理员对具体商家操作会跳转到此界面进行相关操作。根据不同用户的权限进行功能隐藏。** |
| **Deal.vue** | **下订单页面。用户通过商品详情页下订单跳转到此页面。可在此页面进行商品数量调整及个人账户充值操作。下单后会跳转到首页。** |
| **DealView.vue** | **用户的所有订单页面。本页面显示用户的所有订单情况。可以根据订单进行签收、退货、结束操作。对订单的评价可直接进行，对商品的评价跳转到商品页进行评价。** |
| **GoodDetail.vue** | **商品详情页。展示商品的所有信息，可跳转的商店界面。可直接下单或加入购物车。可查询商品评价及自身购买后评价商品。** |
| **GoodsView.vue** | **所有商品的页面。作为商城的首页进行展示。可在本页面跳转到商家或者具体商店。可以进行全局搜索商品或商店。** |
| **Login.vue** | **登录页面。所有账户的登录页面。登录需要用户名、密码及验证码。根据用户的用户类型登陆后跳转到不同的界面。** |
| **Register.vue** | **普通用户注册界面。根据检验规则进行数据填充注册。** |
| **RegiserBusiness.vue** | **商家用户注册界面。和普通用户相同，增添上传身份证正反面及营业执照功能。** |
| **Root.vue** | **管理员界面。管理员可以审核、拉黑、通过其他用户。可以进入具体商家或普通用户界面进行查看。可以查看所有的订单。** |
| **Shop.vue** | **商店界面。展示商家该商店的所有商品，可以在本界面进行商店内搜索。** |
| **ShopCar.vue** | **购物车界面。展示用户购物车的内容，可以进行多个商品多个数量的购买下单** |
| **UserDetail.vue** | **用户详情页。商家和普通用户都进入此页面查看自身的信息。可以进行密码更改与地址更改及对自身账户充值。** |

1. 说明如何使用Spring Boot分析和设计功能
2. Web Controller层的分析、设计与开发
   1. 方案说明

**设计**

在web controller的设计上我主要分为了以下几大模块：

评论模块

订单模块

商品模块

店铺模块

购物车模块

用户模块

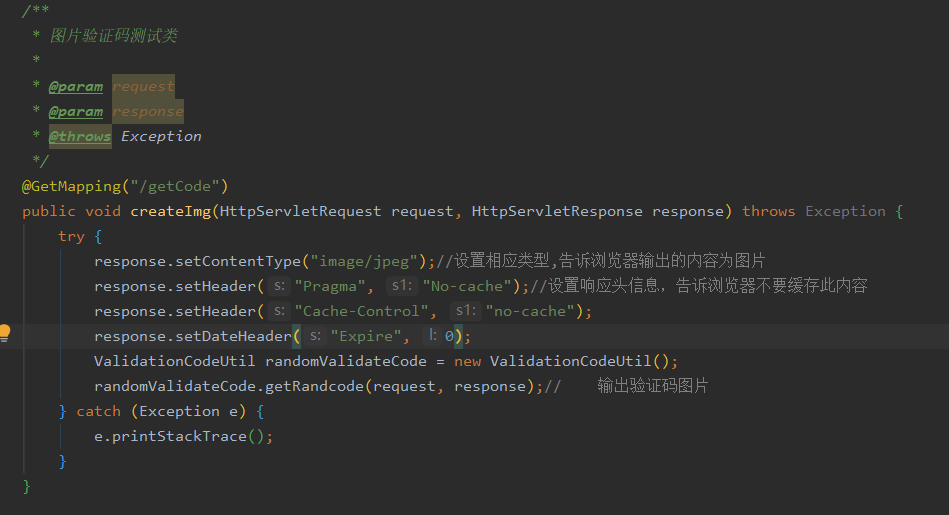
并以模块对相应的接口进行了url上的映射划分，在相应的模块上加入了相应的前缀。

**开发**

由于本次实验后将续可能不会有系统维护等方面的的考虑，所以在开发中主要将各个小模块融入到大模块中，尽量的减少文件的数量，做到高内聚。如将用户的Wallet模块融入到了用户模块下。管理员模块的所有接口与商场模块的所有接口合在一起进行编写。这样可能会导致系统违背了开闭原则，但作为一次实验也未尝不可。

* 1. 关键代码说明。

在controller层大多数都是转发代码，所以如果说关键代码的话，我可能会说一下验证码生成代码：



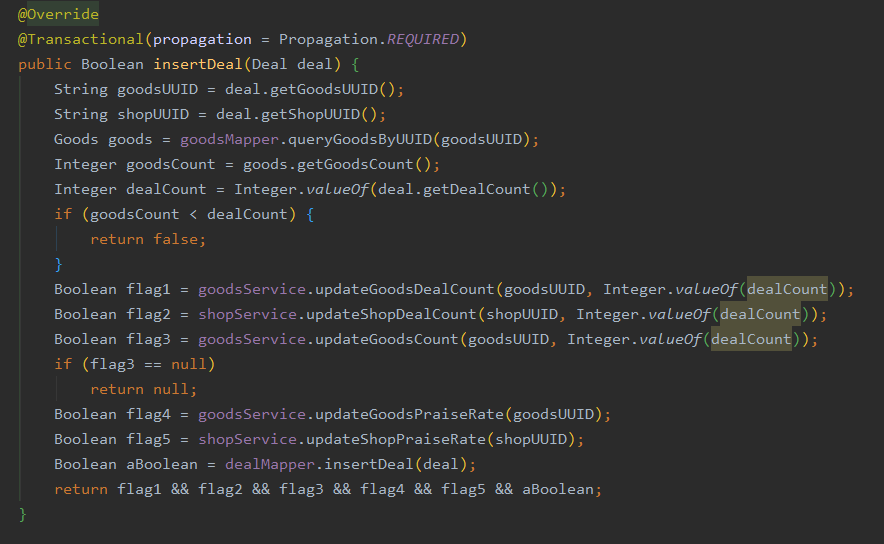
此处的代码主要作用就是在前端请求该接口的时候给出一个验证码图片，其本质的生成代码由我项目中的ValidationCodeUtil类进行控制，而controller层主要是对响应体进行的一些设置。与一般的restcontroller不同，该controller返回值设定的是void是因为其主要的返回内容已被放到了HttpServletResponse中，我们只需要设定相应的响应头即可。

1. Service层的分析、设计与开发
   1. 方案说明

在service层的设计上也主要跟随了controller层的设计思路，将系统分为几大模块。

* 1. 关键代码说明。

关键代码可能就是进行多表操作时加上的事务操作吧，具体代码如下：



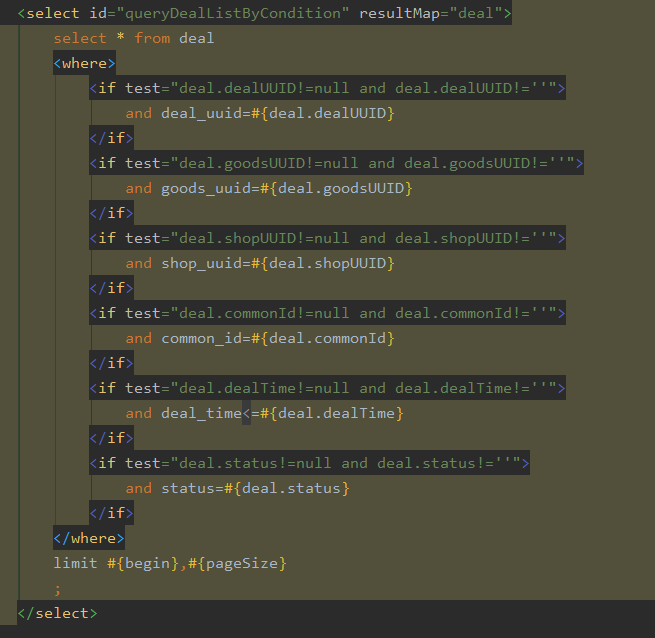
主要是利用springboot中的transactional注释中的required传播模式对事务进行传播。使多表操作时不至于因为中途出现错误而导致数据不一致。

1. DAO层功能的分析、设计与开发
   1. 方案说明

在dao层由于使用的是mybatis框架所以主要就是对其中的sql的编写。由于在最开始的设计中就有想过去写一个分页+筛选器的东西，所以在dao层的sql编写时我有加入if判断从而对service层传入的对象的属性进行筛选，最终的效果就是用户传入一个实体对象，而系统则会自动筛选对象中不为空的字段进行查询并返回结果。

* 1. 关键代码说明

关键代码就是上面所说的if-test的使用：



利用上面所说的方法，对实体对象中的属性进行判断，并进行sql拼接的操作。最终返回结果。

1. 说明如何使用vue构建商城界面
   1. 前后端数据传递

通过axios进行数据传递，因为需要和后端进行大量数据传递，所以需要单独进行js编写与使用。根据对接口的不同条件的限制，根据前端需要进行参数传入与编写，最后根据具体功能进行声明。

* 1. 跨域

前后端跨域问题，在config中进行声明。设置默认baseurl及其他参数，将其引入即可解决跨域问题。

* 1. 路由

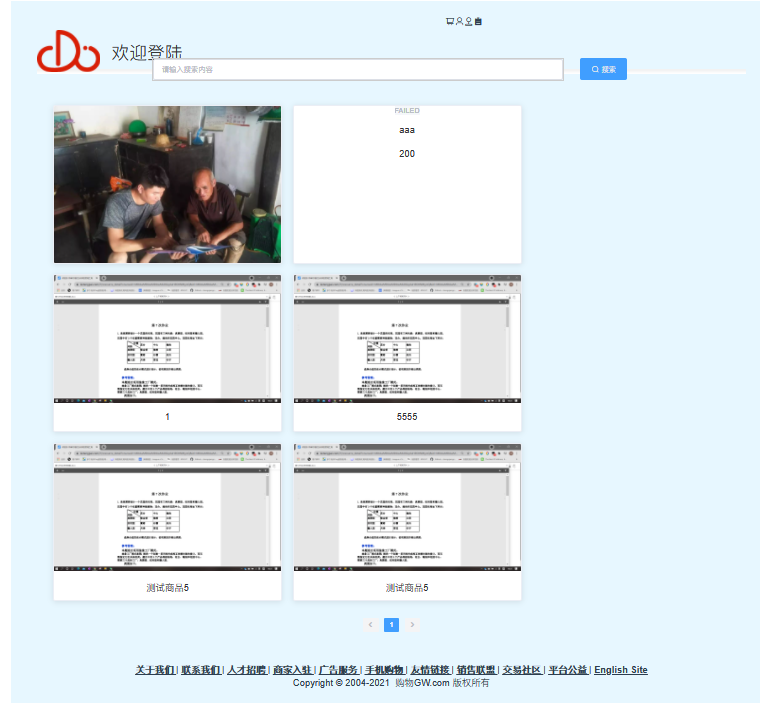
根据前端页面及实际功能需要对不同页面进行拦截，在main中配置即可。

* 1. 详细页面

根据商城功能，进行页面划分。登录页只需要根据不同用户权限进行跳转即可，故使用一个页面。普通用户注册与商家用户注册因为商家用户需要上传自身的营业执照及身份证及不同接口参数，所以分成两个页面。用户详情页商家与普通用户信息基本相同，公用一个页面。商品下单页面、商品详情页面、商品展示页面、订单展示页面购物车页面，是普通用户的操作较为分散，单独列出。商家操作页面及管理员界面操作较为集中，在一个页面中进行操作。

* 1. 页面详细

页面通过不同数据进行展示操作。在beforeCreate中进行数据获取及处理。部分页面需要根据用户的权限进行权限验证，对一些功能进行隐藏或展示及其他操作。

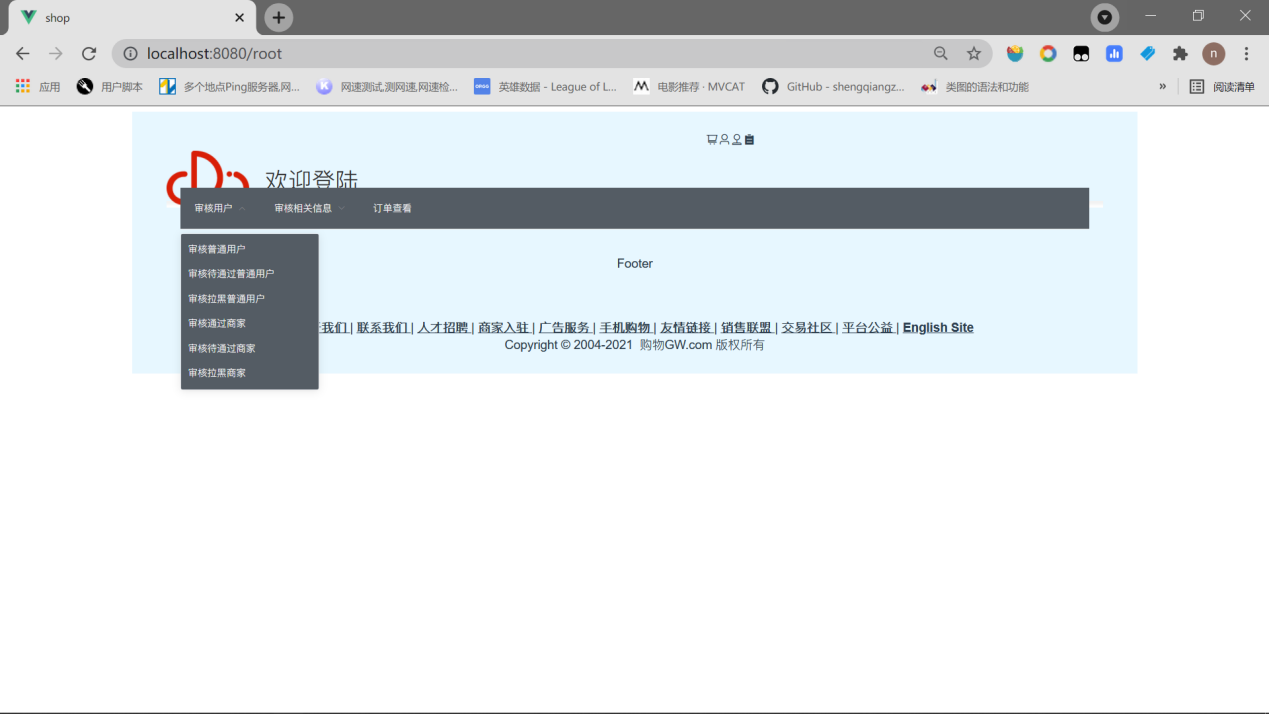
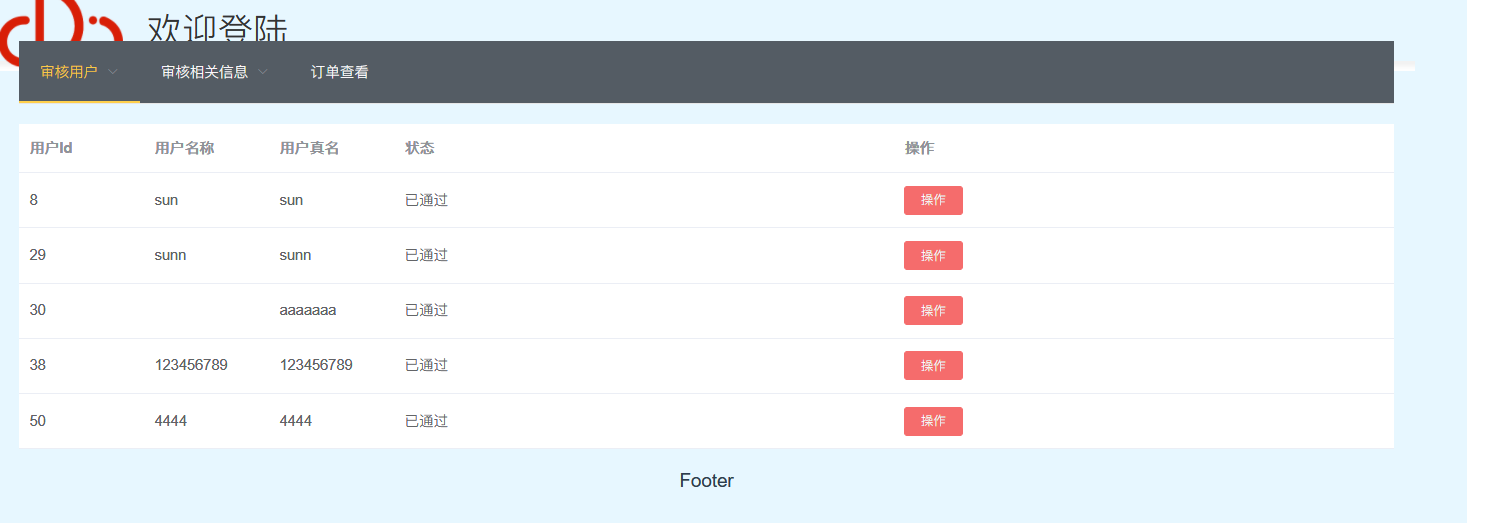
1. 运行截图

商城主界面

可在此界面跳转登录、注册、具体商品、商家、自身详情、订单等界面。可以进行全局所搜商品或商店操作。

 全局搜索商品

登录界面



管理员对用户操作界面

可以审核、拉黑、通过用户



 管理员对商店操作界面

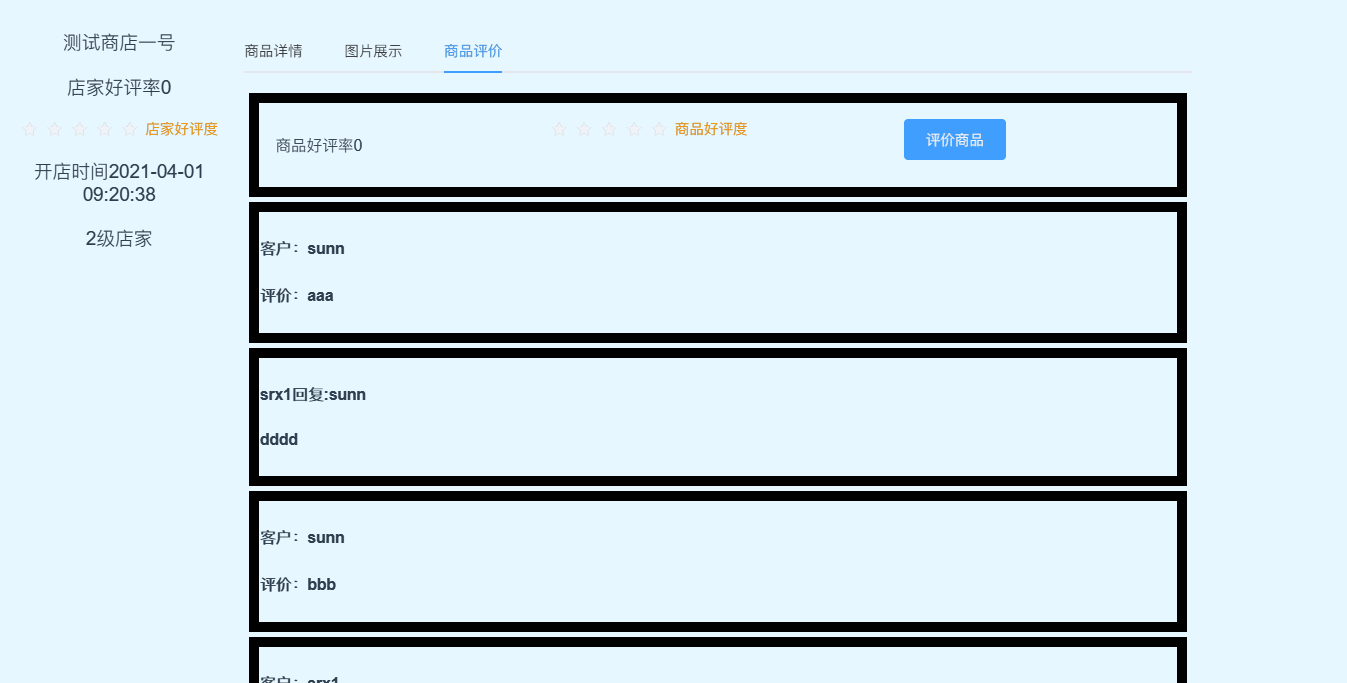
可以对商店进行查看、停业、提升或降低级别操作。如果对具体商家或商品操作需要进入查看界面，跳转到商家界面。



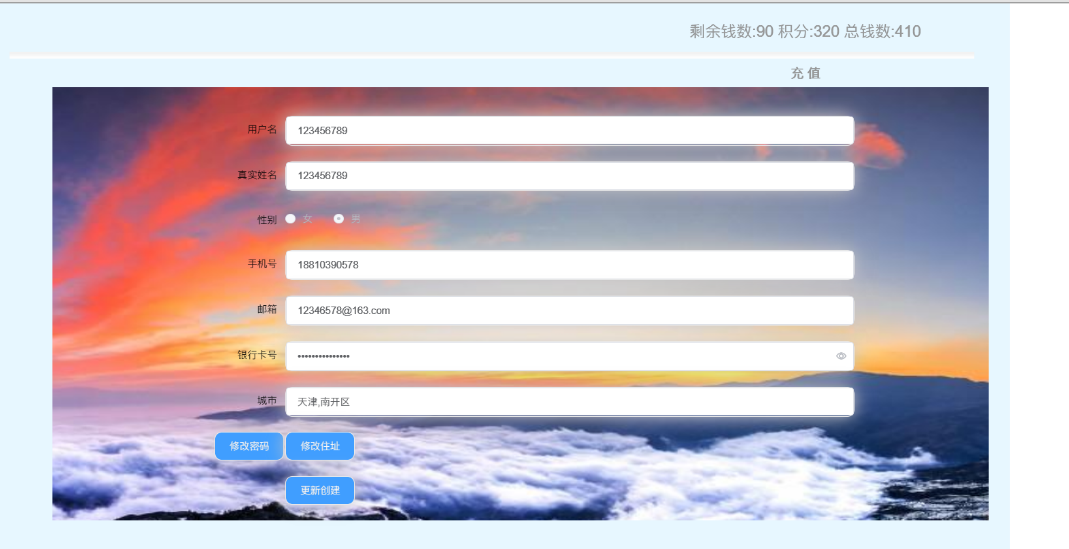
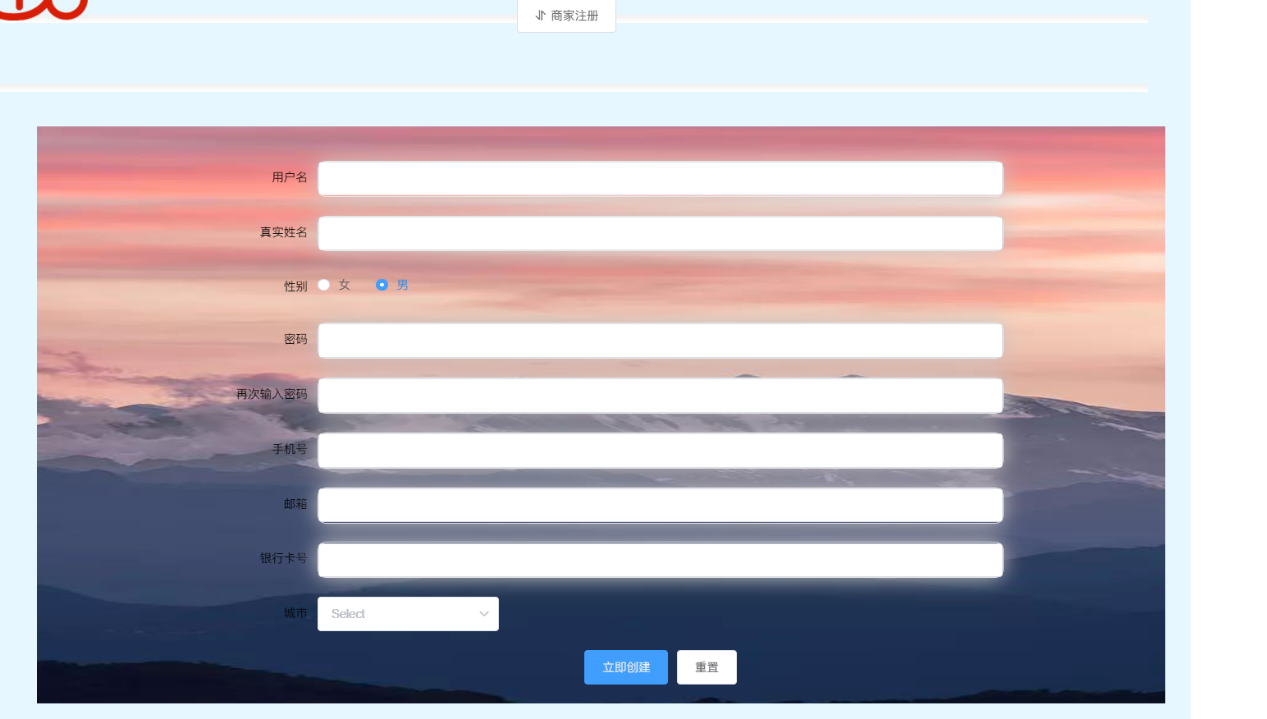
订单查看界面

查看所有的用户订单

商品详情页

 商品评价页

商品下单页面

 个人详情页面

 用户注册页面

购物车界面



用户订单页面

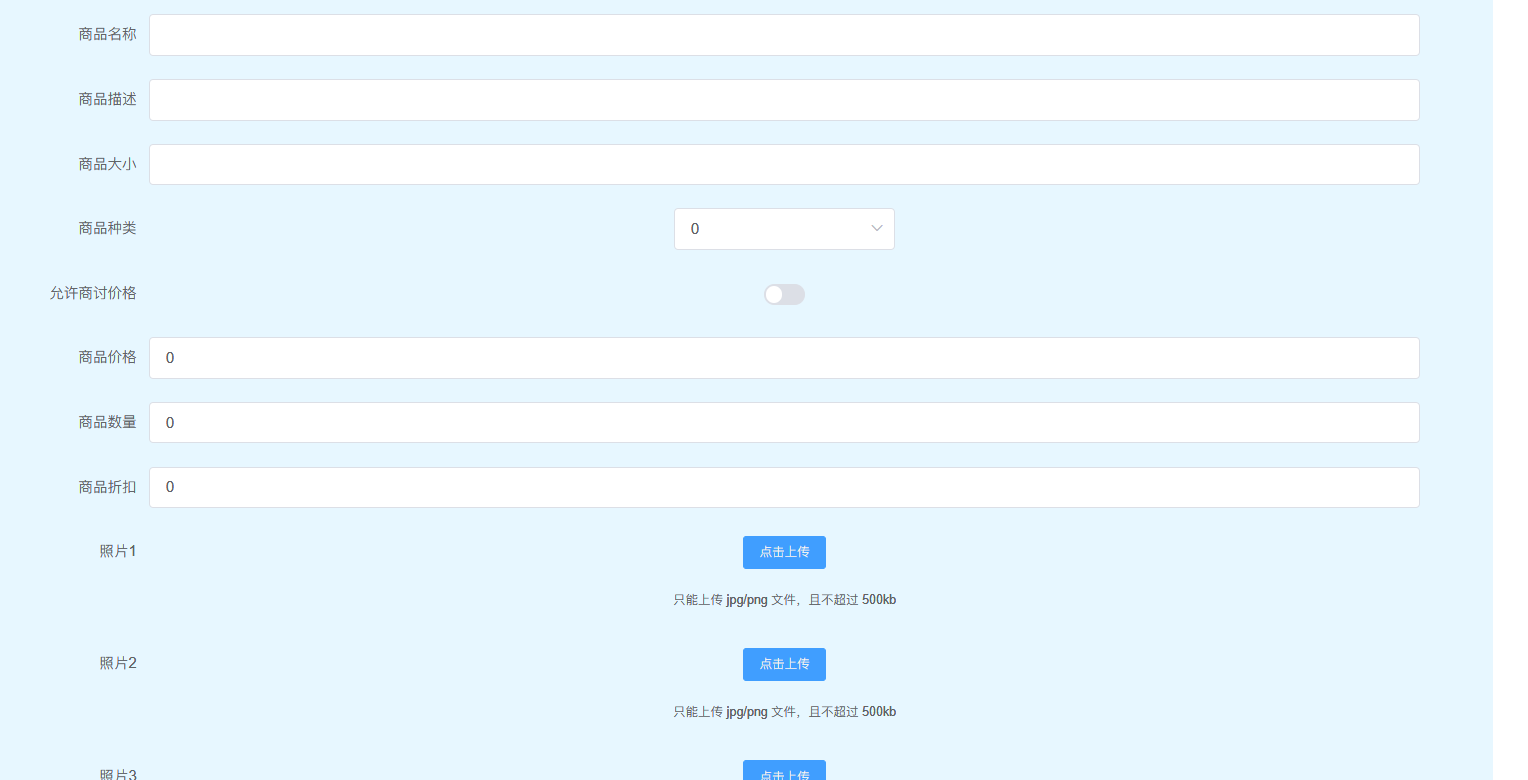
 商家商品操作页面

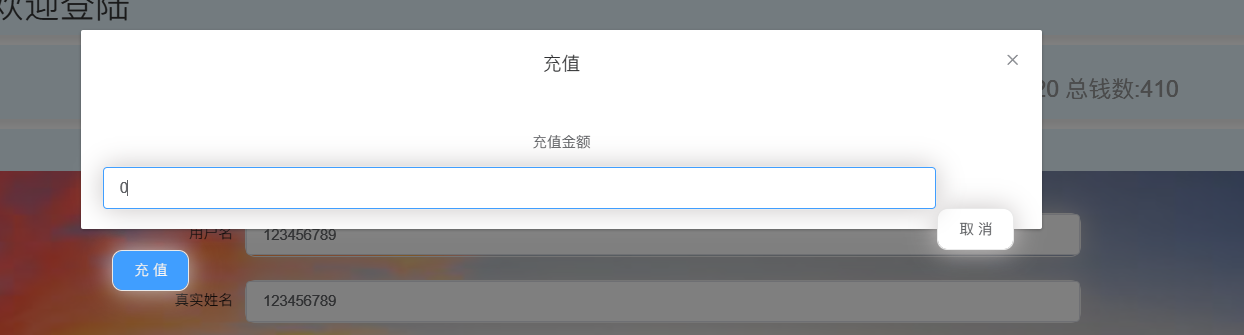
 商品调整页面



 商家信息页面

商家订单页面

 添加商品页面

 商家评论回复界面

充值界面

# 五、实验总结

1 实验中碰到的问题总结

**后端开发中遇到的问题主要有以下几点：**

1. 图片如何存储
2. 验证法应该如何生成
3. 如何进行验证码的校验
4. 后端接口的自动定时执行
5. 数据库设计的可扩展性
6. 对传输对象以及响应码的定义
7. 如何进行全类别自动分页
8. 文件转存

**前端开发中遇到的问题主要有以下几点：**

1. 前端跨域
2. 数据获取及处理
3. 全局拦截器配置
4. Element使用
5. 前端数据格式处理

上述问题后经过网上查找或自行设计以全部解决。

2 Spring Boot使用感想

作为后端的主要开发，我在体验springboot框架之前其实是由两次更早的开发经历的，第一次是大二的javaweb，在其中使用的是java的原生servlet容器。当时最深的开发印象就是繁琐，且不够优雅。后来在暑假期间出于好奇自己进行了springMVC，spring以及mybatis的学习。后使用这个ssm框架自己开发了一套论坛系统。而对于这套框架的使用感受最大的就是配置上的繁琐。尤其是springMVC使用xml配置相关文件时，真的是痛不欲生。同时由于ssm框架其实并没有整合tomcat服务器，所以其还需要自行对服务器进行配置。但在此次实验中使用了springboot框架之后，上述的相关问题可谓是一扫而空。做到了所谓的开箱即用，甚至连服务器都不需要进行配置了。在获得真真正正的提高了编程的幸福感的同时也不由得感叹这些框架开发者的伟大。

**3 Vue使用感想**

Vue是现在主流页面。之前并没有系统的使用过vue进行编写项目。Vue的数据双向绑定更新摆脱了之前的繁琐。通过axios方便的请求数据及绑定，不需要考虑其他复杂的反复构建导入等操作，同一界面数据替换不需要多写重复的界面。通过router来实现不同页面的跳转，简单方便。Vue的使用不需要过多的考虑html页面代码，更多的考虑功能实现。Element的使用，将常用的功能及html进行封装，可以直接引用，通过微调可以实现自己的功能。

现在网上很多的框架及现成的工具可以帮助提升编写，避免了重复造轮子。

4 实验内容总结

**后端**

作为后端开发本次实验可以说收获颇丰，首先是对商城业务流程有了一定的了解，同时对管理员用户的功能以及权限有了更深的认识。在数据库的设计上也完成了一次突破——即数据库在设计完成后仅更改了2次。还有对一些细节上的实现：用户登录时的验证码，传输对象的设计，自定义图床的搭建，文件的转存，反射类的应用，自定义分页器，筛选器的编写等等。但在本次实验中还是有一些改进的空间，如：设计模式的应用，对系统扩展性考虑的不足，业务相对简单，业务逻辑可能存在纰漏导致数据上的问题等等。希望以上的问题可以在以后的实践中尽量的减少。

**前端**

本次前端开发，体会到了前后端分离的开发模式。对商城相关页面功能进行了分析，对不同功能的详细操作进行了了解。使用element，简单的调整开发，对商城页面功能进行了开发。并且利用css、html、javascript传统方式进行了一定程度上的功能、样式实现。