



**QG工作室项目报告**

学 院 计算机学院

专 业 软件工程

班 级 2018 级（1）班

组 别 网络组

姓 名 陈起廷

学 号 3118004951

2018 年 5月

广东工业大学计算机学院制

目录

[一. 项目简介 5](#_Toc22032_WPSOffice_Level1)

[1.1项目名称 5](#_Toc24772_WPSOffice_Level2)

[1.2项目背景](#_Toc19092_WPSOffice_Level2) [5](#_Toc19092_WPSOffice_Level2)

[1.3项目介绍](#_Toc26806_WPSOffice_Level2) [5](#_Toc26806_WPSOffice_Level2)

[1.4项目功能](#_Toc25995_WPSOffice_Level2) [5](#_Toc25995_WPSOffice_Level2)

[1.5运行环境](#_Toc27533_WPSOffice_Level2) [5](#_Toc27533_WPSOffice_Level2)

[二. 项目设计 6](#_Toc24772_WPSOffice_Level1)

[2.1模式设计 6](#_Toc7544_WPSOffice_Level2)

[2.2流程设计](#_Toc20372_WPSOffice_Level2) [6](#_Toc20372_WPSOffice_Level2)

[2.3 模块设计](#_Toc26267_WPSOffice_Level2) [7](#_Toc26267_WPSOffice_Level2)

[2.3.1用户模块](#_Toc24772_WPSOffice_Level3) [7](#_Toc24772_WPSOffice_Level3)

[2.3.2购物车模块](#_Toc19092_WPSOffice_Level3) [8](#_Toc19092_WPSOffice_Level3)

[2.3.3商品模块](#_Toc26806_WPSOffice_Level3) [8](#_Toc26806_WPSOffice_Level3)

[2.3.4订单模块](#_Toc25995_WPSOffice_Level3) [8](#_Toc25995_WPSOffice_Level3)

[2.4架构设计 9](#_Toc19108_WPSOffice_Level2)

[2.5 数据库ER图](#_Toc20376_WPSOffice_Level2) [9](#_Toc20376_WPSOffice_Level2)

[2.6 数据库详细](#_Toc26689_WPSOffice_Level2) [11](#_Toc26689_WPSOffice_Level2)

[三. 详细功能设计](#_Toc19092_WPSOffice_Level1) [11](#_Toc19092_WPSOffice_Level1)

[3.1 登录、注册、找回密码模块](#_Toc14251_WPSOffice_Level2) [11](#_Toc14251_WPSOffice_Level2)

[前台设计](#_Toc27533_WPSOffice_Level3) [11](#_Toc27533_WPSOffice_Level3)

[后台设计 1](#_Toc7544_WPSOffice_Level3)2

[模式基本图：](#_Toc20372_WPSOffice_Level3) [12](#_Toc20372_WPSOffice_Level3)

[3.2 商品模块](#_Toc3436_WPSOffice_Level2) [14](#_Toc3436_WPSOffice_Level2)

[前台设计](#_Toc26267_WPSOffice_Level3) [14](#_Toc26267_WPSOffice_Level3)

[后台设计](#_Toc20552_WPSOffice_Level3) [15](#_Toc20552_WPSOffice_Level3)

[3.3 订单模块](#_Toc20385_WPSOffice_Level2) [16](#_Toc20385_WPSOffice_Level2)

[前台设计](#_Toc879_WPSOffice_Level3) [16](#_Toc879_WPSOffice_Level3)

[后台设计](#_Toc27088_WPSOffice_Level3) [17](#_Toc27088_WPSOffice_Level3)

[3.4 购物车模块 1](#_Toc20385_WPSOffice_Level2)8

[前台设计](#_Toc6555_WPSOffice_Level3) [18](#_Toc6555_WPSOffice_Level3)

[后台设计](#_Toc31263_WPSOffice_Level3) [19](#_Toc31263_WPSOffice_Level3)

[3.5 管理员模块](#_Toc20552_WPSOffice_Level2) [20](#_Toc20552_WPSOffice_Level2)

[前台设计](#_Toc32469_WPSOffice_Level3) [20](#_Toc32469_WPSOffice_Level3)

[后台设计](#_Toc23584_WPSOffice_Level3) [20](#_Toc23584_WPSOffice_Level3)

[四． 程序测试](#_Toc26806_WPSOffice_Level1) [21](#_Toc26806_WPSOffice_Level1)

[4.1 登录、注册和忘记密码](#_Toc879_WPSOffice_Level2) [21](#_Toc879_WPSOffice_Level2)

[4.2 用户操作](#_Toc27088_WPSOffice_Level2) [22](#_Toc27088_WPSOffice_Level2)

[4.3 管理查看](#_Toc6555_WPSOffice_Level2) [25](#_Toc6555_WPSOffice_Level2)

[五 .程序亮点](#_Toc25995_WPSOffice_Level1) [26](#_Toc25995_WPSOffice_Level1)

[5.1 数据库操作优化](#_Toc31263_WPSOffice_Level2) [26](#_Toc31263_WPSOffice_Level2)

[5.2 数据展示](#_Toc32469_WPSOffice_Level2) [26](#_Toc32469_WPSOffice_Level2)

[5.3 用户体验](#_Toc23584_WPSOffice_Level2) [27](#_Toc23584_WPSOffice_Level2)

[六 .程序难点](#_Toc27533_WPSOffice_Level1) [27](#_Toc27533_WPSOffice_Level1)

[6.1 BUG的查找与解决](#_Toc10095_WPSOffice_Level2) [27](#_Toc10095_WPSOffice_Level2)

[6.2 数据库表的设计与关联](#_Toc27552_WPSOffice_Level2) [28](#_Toc27552_WPSOffice_Level2)

[七. 体会总结](#_Toc7544_WPSOffice_Level1) [28](#_Toc7544_WPSOffice_Level1)

[7.1 记录过程](#_Toc32160_WPSOffice_Level2) [28](#_Toc32160_WPSOffice_Level2)

[7.2 技术收获](#_Toc31003_WPSOffice_Level2) [29](#_Toc31003_WPSOffice_Level2)

[7.3 总结心得收获和感言](#_Toc19111_WPSOffice_Level2) [29](#_Toc19111_WPSOffice_Level2)

# 项目简介

## 1.1项目名称

QG二手市场

## 1.2项目背景

开发者参加QG工作室的考核而做出来的项目，项目兼小型，旨在提高开发者的代码能力

## 1.3项目介绍

QG二手市场可以给用户提供一个良好的网络环境，对于自身不需要的商品用户有权利放到QG二手市场去售卖，用户也可以在QG二手市场上购买商品。

## 1.4项目功能

本项目具有用户登录、用户注册、游客游览、查询二手商品、售卖二手商品、管理订单的功能、购物车记录商品等功能。

用户分为普通用户和管理员，普通用户上传的二手商品需要被管理员审核通过才能售卖，管理员有权决定用户能否售卖商品。

用户与用户之前通过商品购买实现交互，用户可以查看特定用户的所有可售商品，买提交订单时需要卖家审核，卖家有权决定售卖此商品给其他用户。

购物车记录此用户添加的商品，用户可以通过进入购物车购买已经添加的商品。

## 1.5运行环境

该项目基于JAVA和MySQL技术开发，运行服务器为tomcat-8.5版本

MySQL版本为8.0。

要导入额外的jar包：

commons-fileupload-1.3.1.jar、commons-io-2.4.jar、jstl.jar

、mysql-connector-java-8.0.15.jar 、poi-4.1.0.jar、standard.jar

# 项目设计

## 2.1模式设计

本项目采用MVC的设计模式，其中DAO层连接数据库调用数据库的数据，DAO层数据返回给Service层进行数据组装和业务逻辑处理，Controller层负责把前台的数据传到Service层进行逻辑处理后进行返回信息，再把数据呈现给View层，即客户端。

MVC设计模式允许试图层和业务层分离，这样就允许更改视图层代码而不用编译模型和控制器。MVC使开发和维护用户接口的技术含量降低。

MVC模式的Model层对应于数据库操作，可以有效地调用数据库的数据并进行处理，以得到想要的功能。MVC的模型图如下：

## 无标题

## 2.2流程设计

1. 先设计数据库表，按需求构建四张表 用户表、购物车表、订单表、商品表。
2. 对应的表在MVC的DAO层写相应的增删查改，购物车要实现添加、删除、删除所有的购物车信息、更新购物车的信息、查询个人用户的所有购物车信息反馈回给前台；
3. 订单表要实现添加、删除、查询所有订单，卖家与买家之间的修改、对订单状态的改变，订单管理要实现 完成收货、取消交易等功能。
4. 用户表要实现添加、查询，普通用户可以被管理员禁止销售商品，用户要和订单、商品之间实现1：N的关系，一个用户可以有多个订单，也可以有多个商品；用户要登录，可以通过数据库的查询和验证实现登录。
5. 商品表，商品可以被用户上传，管理员要对商品进行审核，商品中要有管理员可以修改的审核字段，该字段决定该商品是否能出售。当用户被管理员禁止售卖商品后，该用户所有的商品都不可出售。
6. 用service层对Dao层数据库操作取出来的数据进行组装，逻辑判断，返回给控制层

控制层当要返回数据的同时要返回提示给客户端，前端视图层用jsp代码写出来展示数据呈现给用户看。

程序流程设计图：

QG二手商城展示页面

Sevice层写逻辑判断代码，分为四个逻辑层

用户逻辑层、商品逻辑层、购物车逻辑层、订单逻辑层

查询所有页面的时候对数据进行分页处理返回给控制器层

编码过滤器

控制层写Servlet，获取前端传来的数据，把数据传到后台进行处理

用户数据库操作：登录、注册、查询用户信息

商品数据库操作：添加商品、删除商品。多种类型查询商品

订单数据库操作：购买商品创建订单、按用户查询订单输出文件xlsx格式允许用户导出订单、卖家修改订单、买家修改订单

购物车数据库操作：添加商品进购物车、删除、查询所有购物车信息

## 2.3 模块设计

### 2.3.1用户模块

1. **用户注册**实现，前台控制格式，比如前台控制的密码格式为“10-15个字母和数字混合”、“邮箱格式要正确”、账号格式“10-15个纯数字”、电话号码、以及姓名等。若该账号已存在数据库中，提示信息到前台让用户换个账号注册。
2. **用户登录**，前台控制账号密码的输入格式，把正确的格式传到数据库查出来，若匹配不上，则用户账号或者密码错误；
3. **忘记密码**和**修改密码**为同一个功能，都需要用户去验证邮箱修改新的密码，忘记密码是未登录的操作，修改密码是登录后的操作，所有修改密码要提示用户去重新登录。
4. 用户可以修改自己的电话号码、邮箱地址、姓名。
5. 用户可以上传照片，利用servlet的文件上传就可以做到。
6. **管理员**和**普通用户**存在着1对多的关系。管理员可以对用户的商品和用户本身的权限进行管理和限制。
7. 游客模式主要在前台控制，通过判断是否登录来决定游客可以看到页面的数据，游客的功能很局限，只能看到商品和查询其他用户的信息。

### 2.3.2购物车模块

1. 用户在查询商品时可以添加商品到购物车，前台把这个添加的信息数据传给后台，拿到数据就可以进行添加。特殊情况为：用户本来已经添加该商品到购物车了，再次去商品详情页面把商品添加到购物车的时候，此次我思考得来的结果是单独增加购物车中该商品的购物车数量而不是再去添加一个商品到购物车。
2. 用户可以在购物车删除商品，但不修改购物车的信息。
3. 用户可以在购物车购买商品，当用户把上传的商品删除了，其他用户的购物车也要删除该商品。

### 2.3.3商品模块

1. 用户可以添加商品，商品与用户之前存在N：1的比例关系。
2. 用户可以删除自己添加的商品。
3. 客户端可以根据商品类型、商品id、用户id从服务端查询商品，最后显示在前端

### 2.3.4订单模块

1. 卖家可以修改订单（在发货时填入一个发货地址），买家也可以修改订单（当订单还处于发货前的状态时，可以修改订单中的电话、收货地址、相关申请信息），卖家和买家都有取消订单的权利，但买家想买该商品时必须征求得到卖家的同意（即需要通过卖家审核）。
2. 当用户存在订单时，用户可以导出自己的订单，这里可以用servlet的文件下载。

## 2.4架构设计

**该项目使用的MVC设计模型，但整体项目中我选择用了四层架构**

①**View**层，主要用到技术有jsp，css，html，简单的javaScript，展示该呈现的东西给用户。

②**Controller**层，主要用到的技术是Servlet，通过请求转发或重定向，对前台的数据进行获取和返回。控制层把数据处理交到给了Service层，控制层只是控制和页面的跳转，不对数据进行业务逻辑的处理，相当于汽车中转站。

③**Service**层，收到控制层的数据处理请求，Service层从Dao层获取数据，处理业务逻辑，订单的管理（收货成功、取消交易、待付款、待发货、发货）这些功能都由Service判断后再把对应的数据存储到数据库中，但客户端用户传来的数据不符合时，Service层不会把数据存进数据库里。Service层要对Dao层传来的基本增删查改的数据进行数据组装。用户购买商品时，要同时把这个购买申请添加到订单中，并把订单数据存到数据库中，在服务端卖家同意售卖后，该商品的销量要增加对应的数量，同时商品的库存数量也要相对应地减少。用户付款时需要账户里存在足够的金额支付商品，这都需要经过Service层去判断是否合理。

④**Dao**层，对数据库表的操作，对应的数据库表要构建对应的数据模型从而形成JavaBean，方便Dao层读取数据时封装好成一个对应的对象传递给Service层进行数据组装。

## 2.5 数据库ER图

**用户E-R图：**

用户

**账号**

密码

销售权限

头像

金钱

电话号码

身份

邮箱

姓名

性别

星评

**购物车E-R图：**

购物车

**加入时间**

数量

用户

商品

## 

**订单E-R图：**

收货地址

**订单号**

订单

发货地址

买家

卖家

订单总价

商品

商品数量

订单状态

联系方式

创建时间

**商品E-R图:**

商品

**商品号**

商家

商品权限

商品名

类型

价格

图片

销量

库存量

商品信息

## **整体ER图：**



**commodity**

Id: varchar(20) Name: varchar(30)

Message: varchar(500) Sum: int(5)

Price: double(10, 2) Photo: varchar(100) Buy\_Sum: int(5) Type: varchar(10) Owner: varchar(20)

Condition1: varchar(10)

**user**

Id: varchar(20) Password: varchar(255) Sex: varchar(5)

Email: varchar(30) Name: varchar(100) Phone: varchar(12) Priority: varchar(20) Star: int(2)

Money: double(10, 2) Photo: varchar(100) Sell: varchar(20)

**order1**

Id: varchar(30) CreateTime: datetime(0) Condition1: varchar(10) Owner: varchar(20) Buyer: varchar(20)

SendAddress: varchar(50) FromAddress: varchar(50) Phone: varchar(12) Relate\_Message: varchar(255) Price: double(10, 2)

GoodId: varchar(20) GoodSum: int(3)

**shopcart**

Commodity\_Id: varchar(20) User\_Id: varchar(20)

Sum: int(4)

Date: datetime(0)

## 2.6 数据库详细

1. 用户表存在 用户账号、密码、身份、销售权限、姓名、电话号码、邮箱地址、性别、

账户余额、头像等属性，主键为**用户账号**，具有有唯一性。用户表和购物车表、订单表、

商品表都存在着1:N的关系。因为用户可以上传商品、可以作为卖家或买家生成订单、添

加商品到购物车。

2. 购物车表主要由 用户账号、商品、商品数量 、加入商品的时间，**加入商品的时间**作为主键，具有唯一性。购物车和商品存在N：1的联系，一个商品可以被多人添加进购物车。

3. 订单表主要由 订单号、订单时间、订单完成状态、卖家、买家、发货地址、收货地址、收货人的电话号码、买家申请订单的相关信息、订单总价、商品ID、商品数量组成。其中**订单号**为主键、**卖家和买家**均为参考用户表的外键，其中订单完成状态按功能划分有 **审核中、取消交易、交易完成、已发货、待付款**等字段共同组成这一状态。买家和卖家根据身份对订单状态的操作也是不一样的。

4. 商品表主要由 商品号、商品照片、商家、商品名、商品信息、销量、库存量、商品权限组成。其中**商品号**为主键，商品的上传者**商家**为参考用户表的外键，根据逻辑处理关系，没有商家就没有商品。

# 详细功能设计

## 登录、注册、找回密码模块

前台设计

1. 数据格式要限制好。账号限制格式为10-15位纯数字组合。密码格式限制为10-15位数字字母混合组合。注册新用户时，电话格式、邮箱格式、名字格式都要受到限制，表单设计两次输入密码，在前台通过jquery拿到两个密码的数据，当密码一致时，密码验证才算结束。以确保后台拿到的数据是正确的。找回密码功能同样如此，找回密码我的设计思路是让用户输入账号、新密码、二次密码、账号邮箱。
2. 前台通过AJAX异步处理和jquery的部分高效功能对后台的判断信息进行反馈并将处理结果反馈给用户。
3. 表单设计要美观，通过css对form表单的input框进行样式调整，增加用户的系统体验。

### 后台设计

1. 后台使用Servlet对前端传来的数据进行获取，并把数据一层层地往下传，传

到数据库中，登录时，数据库对传来的账号和密码进行sql语句查找，MD5算是一

种通用的加密算法，通过读取每个字母获取到的Hash值对密码进行重加载，其中密

码经过MD5进行加密后才能与数据库中已存在的用户密码符合。若存在，返回true，并且把该用户的全部信息组装成一个User封装实体类放进Session域确认User已登录，否则返回false。

1. 用户注册时通过前端传来的信息并传到Service层，Service层需要先调用Dao

层查找数据并对用户进行判断是否已经存在。若用户已存在，则反馈该信息给用户，

若用户不存在于数据库中，注册信息和用MD5加密后的账号密码可以存进数 据库中，

并在servlet中把信息放进request对象中，调用request的请求转发方法请求转发

到登录界面。

1. 找回密码功能主要体现在数据库对前台传来的账号和邮箱信息进行验证，通过

select的where语法进行查找。若存在此人，则把新的用户密码转换成MD5加密

后的密码存放到数据库中，若不存在此人，则返回错误信息给客户端。

### 模式基本图：

前台展示

格式验证，邮箱、个人账号校对、信息匹配则找回密码成功

后台逻辑判断、账号是否存在

数据格式验证

后台验证密码、账号存在、一致

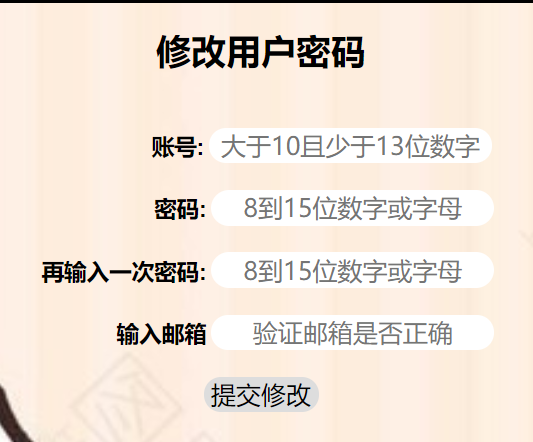
前台验证密码、账号格式

忘记密码

注册

登录





## 商品模块

前台设计

1. 前台提供一个界面（首页）进行后台传回来的数据进行数据展示，分别展示出商

品的照片、价格、销量给用户看，前端需要控制好页面，当没有商品时要怎么提示，

有商品时如何进行分页提示，进行商品分页的时候把上一页或者下一页的信息传递给

后台。

1. 前端通过提供不同的form表单，输入不同的数据对商品进行查询，比如设计一个

form表单去模糊查询具有相关名字的商品，通过类型选择查询到相关类型的商品，通过查找用户账号得到该用户售卖的所有商品。

1. 当用户点击商品图片时，用户可以去到商品的详情页进行加入购物车和购买
2. 用户购买商品时，需要对应的form表单把数据传给后台，商品和订单模块其实

是连接起来的，当用户确认购买**商品把收货地址、联系方式、商品数量和相关的申**

**请信息**都传都后台进行验证。把商品的某些信息传到数据库里去

1. 前台通过判断一个商品javabean的某个属性，可以禁止用户购买自己上传的商

品。

1. 前台写一个个人商品页面，当点击时可以去到后台，把用户上传的所有商品都进

行展示出来（禁止发售、允许发售、审核中的商品都可以展示出来）。用户可以在个

人商品页面中对商品进行删除和添加商品操作（添加商品需要权限，若无此权限，则用户无法添加商品），用户添加商品所填信息同样是通过form表单提交到后台。

后台设计

1. 后台拿到前端传来的请求，得到商品分页的页码和每页显示数据的大小，在Service层调用工具类的查询所有把总数据量查出来，按查询的具体请求（按类型、按商品名、按商家、按销量）查出来，把**页码、每页显示数据的大小、最大页数、商品Javabean组成的List集合、总数据量**封装成一个Page辅助类，在servlet层将该数据放到request域中返回给前台。
2. 主要逻辑处理在service中，不同的form表单提交过来时调用的servlet不一样，Dao层主要负责的就是把客户端想要的数据查出来并封装成一个个JavaBean，再把这一个个javabean放到集合中去，返回集合，就可以让前台进行相对应的页面展示了。该过程sql语句是比较繁杂的，对不同的商品查询就有不一样的sql语句。
3. 前端跳转到商品详情页也要通过数据库的查询，后台拿到商品的id可以去到数据库里把该商品的所有信息进行查询并封装成一个对象返回信息给前台看。
4. 具有销售商品权限的用户可以添加商品，否则无法添加商品，这一关键点通过前台控制。添加商品传过来的数据分别有**照片、商品信息、商品库存、商品名字、商家、商品价格**，商品号按当前时间的System.currenttimemills()得到的当前时间将long类型转换成String类型存放到数据中，商品的初始销量为**0**，且商品的初始权限为**“审核中”，**商品是否上架由该系统的管理员决定。
5. 用户删除商品时把商品id一直传到调用数据库数据的Dao层，在Dao层中用sql语句delete **\* from table where id=?**进行删除操作，并把是否删除成功的信息传回到前端，让用户拥有良好的体验。
6. 查询商品需要控制好商品权限，展示给所有人看到的商品的商品权限都未“**允许发售”**，“**审核中**”的商品只有用户自己和管理员能看到，**禁止销售**的商品只有用户自己的能看到，管理员能看到的商品只有**审核中**和**允许发售**的商品。对查询商品呈现给用户而言，不一样的查询都有不一样的字段，主要通过sql语句进行判断从而让查询出来的数据准确无误。

具体流程图

开始

商品查询

是否存在商品

否

返回无信息

是

终止

将数据封装返回



## 3.3 订单模块

### 前台设计

1. 前端通过导航条设置入口，入口便是通过a标签跳转到后台把用户订单全部查询出来。前台主要通过table呈现数据，订单有多个状态，在前台主要通过select标签供用户选择修改，作为订单的买家和卖家看到的select标签的option呈现出来的东西不一样，在写jsp页面时主要通过jstl的<c:if>标签进行用户是**卖家**还是**买家**的判断。
2. 关于用户修改订单的操作，卖家和买家都可以对订单的订单状态修改，但具有一定的逻辑控制，因为卖家有权不售卖商品给买家，当订单的状态为“**审核中**”的时候，卖家可以**取消交易**，也可以接收审核让**买家付款，**具体逻辑靠后台判断返回给前台显示。
3. 订单可以导出，前台用a标签进行具体操作，让后台对所需的文件进行分析再返回给用户去下载。

后台设计

1. 后台对前端传来的数据和用户的id对后台数据进行匹配，若该用户存在着订单

信息，则对数据库中的信息组装成一个个订单的实体类再放入集合中返回给前台并

展示给用户观看订单。

1. 订单的状态修改操作进行后，后台负责逻辑分析，如果订单卖家已经同意售卖，

那么订单买家可以选择取消交易、付款；如果订单买家已经付款，那么订单卖家只能进行发货并填写发货地址的操作；买家在订单处于已发货的状态后，买家可以进行收货成功的修改操作；其他情况均无法修改订单的状态。买家在订单未发货前可以修改订单的信息，例如修改订单的发货地址、订单相关信息、订单的收货人号码。

1. 一个用户的订单可以被导出。在后台进行的操作是得到该用户的id，对用户id

进行分析并放到数据库中进行核对，把该用户的全部订单信息放到ResultSet集合中进行获取并用插件生成Excel表格文件，再在控制器中对订单进行response的回写返回文件供用户下载。



具体流程图

开始

修改订单状态

导出订单

修改订单信息

买家 否

是否有订单数据

按对应状态进行修改订单状态，成功则返回修改成功、失败则返回修改失败

判断身份

判断身份

卖家 是

在发货前可修改地址、申请信息、电话

在发货前修改地址

前修改发货身份

生成文件并返回文件流

、

终止

终止

终止

注意：订单状态顺序分为：审核中<待付款<待发货<取消交易<已发货<交易完成

其中买家可以在订单发货前取消交易。卖家可以在待付款前取消交易。

即卖家一旦通过审核，他就对该商品不具有决策的能力，主动权重新回到买家的手里。

## 3.4 购物车模块

前台设计

1. 购物车模块主要显示在购物车页面，购物车可以在某商品详情页被添加，添加购物车信息时只要填写数量信息便可，商品ID和用户被隐藏在form表单中一起被提交到后台进行数据登记。
2. 在购物车页面买家可以删除商品，当需要购买时，可以对商品进行数量改变，但不能超过商品的库存值。如果商品当时处于禁止发售的情况时，购物车页面中该商品不可被购买（显示已经下架），商品被商家删除后，购物车对应的商品会不显示在页面当中。
3. 同样的商品被添加到购物车中，在购物车中只会在原来购物车商品存在的商品数量上增加新的添加数量。

### 后台设计

1. 添加商品到购物车时，获取form表单的信息，得到**商品号、用户账号、数量，**并自动**生成创建时间**组装该购物车实体类传到Service层，在Service层调用dao层的查询方法查询该购物车商品是否存在于购物车表中，若存在，则添加购物车原来的数量，若不存在，则把该实体类购物车传到dao层进行数据实体类拆分，添加到数据库的购物车表。
2. 查询个人购物车时，数据库将所有存在的购物车信息联合商品共同查询出来呈现给用户，数据组装时需要形成一个购物车展示的javabean，其中包括两个属性：订单实体类和购物车实体类。将该实体类放进list集合里进行数据返回给前段进行展示。
3. 在购物车进行购买商品时，对商品进行单实现购买，需要填入地址信息等内容。

购物车的所有信息生成订单，后台对购买商品的id和用户进行删除该用户购物车里商品的操作。

基本流程图

开始

展示商品

删除

添加

数据：商品号、用户账号

·

数据：加入时间、商品数量、商品号、用户账号

用户

该用户的购物车是否存在该商品

数据库where语法进行删除

查询购物车表该用户的购物车信息组装成list集合并返回

否

是

直接添加

添加商品数量

终止

终止

终止



## **3.5 管理员模块**

前台设计

1. 前台控制显示管理员才能显示的标签，用户身份不是管理员不能看见普通用户

看到的标签显示。

1. 管理员有两个页面，一个是管理用户页面，管理员只能对用户是否具有销售商

品权限该字段进行修改；一个是管理商品页面，对审核中的商品和已经上架的商品进行修改商品状态字段。

后台设计

1. 后台对商品的id进行审核时，传进来的是商品的id，只要从数据库中找到商品并修改其商品权限状态就可以了。
2. 后台对用户的管理，当用户本身处于**可销售**商品状态时，管理员修改其权限为**不可销售**商品，该用户本身所上传的商品全部都要下架，后台servlet层调用多次dao的方法进行修改。当用户本身处于**不可销售**状态时，管理员修改其权限为**可销售**商品，该用户所上传的所有商品都可再次被管理员审核，即将所有该用户的商品修改其商品权限为“**审核中**”。

## 具体流程图

开始

管理商品

管理用户

修改商品权限

修改用户销售权限

修改其所有商品权限为审核中，用户销售权限为是

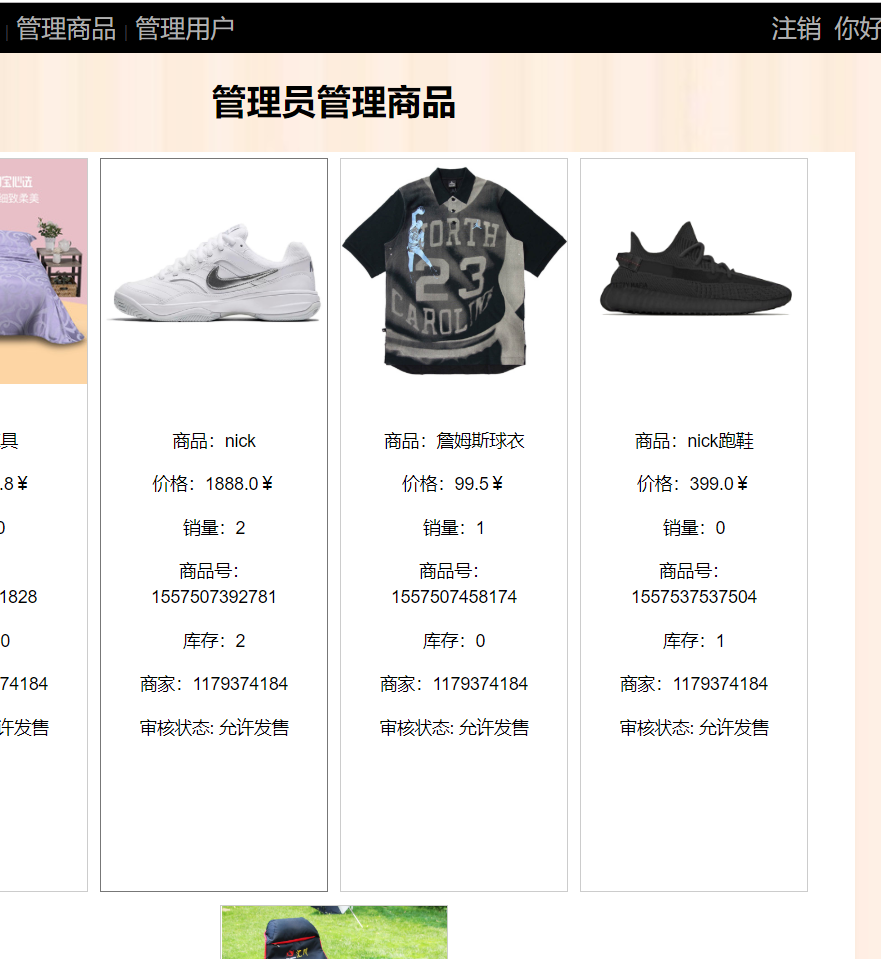
判断用户初始销售权限

修改其所有商品权限为禁止发售，用户销售权限为否

是 是 否

终止

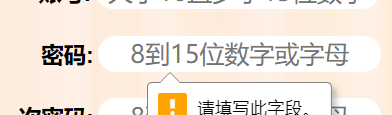
终止



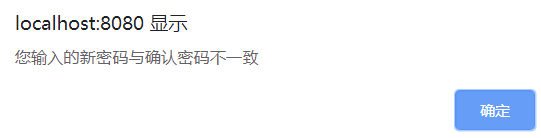
# 程序测试

## 4.1 登录、注册和忘记密码

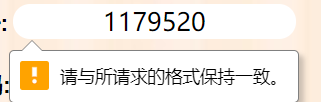
1. 当输入为空时不允许提交数据到后台。



2. 当注册和忘记密码输入二次密码不符合时不允许提交数据到后台。



1. 输入数据格式不对不允许提交数据到后台。

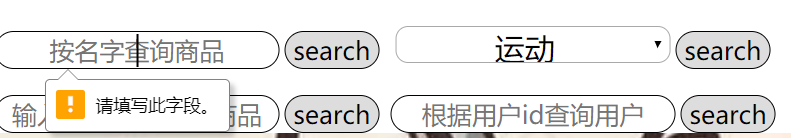


1. 后台进行数据校验后返回信息

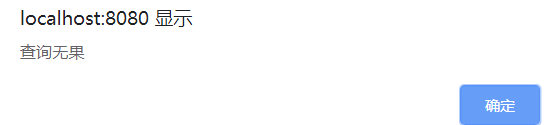


4.2 用户操作

1. 商品多查询设置，数据为空不允许查询。



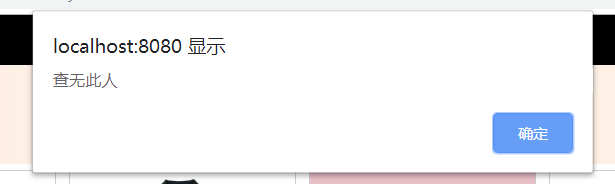
1. 查询成功则展示商品，查询不成功则返回查询信息并返回首页。



1. 点击自己上传的商品进到详情页面后，不能对商品进行购买



1. 用户可以查看其他用户的信息，但需要输入其他人的用户账号。 若有该用户则展示该用户的信息，否则显示查无此人，为防止数据空指针传递，前台设定输入值不为空，同样设置的是input的required模式提交表单。

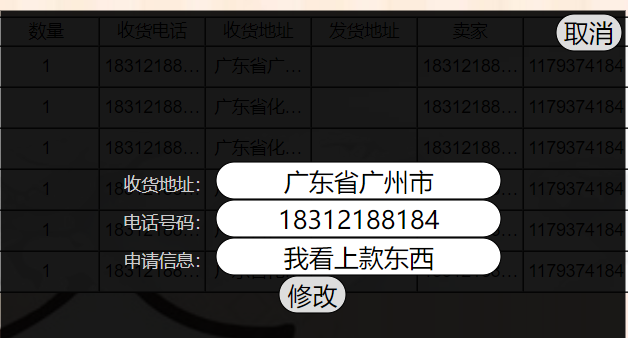




1. 卖家和买家可在订单不同的状态下进行修改订单，详细操作以上有解释，此处进行简单展示，但订单需要修改的状态不是所需状态时，后台会禁止用户进行该操作。

卖家可以在发货的时候填上发货地址，买家可以在发货前修改订单的收货地址。导出订单操作。

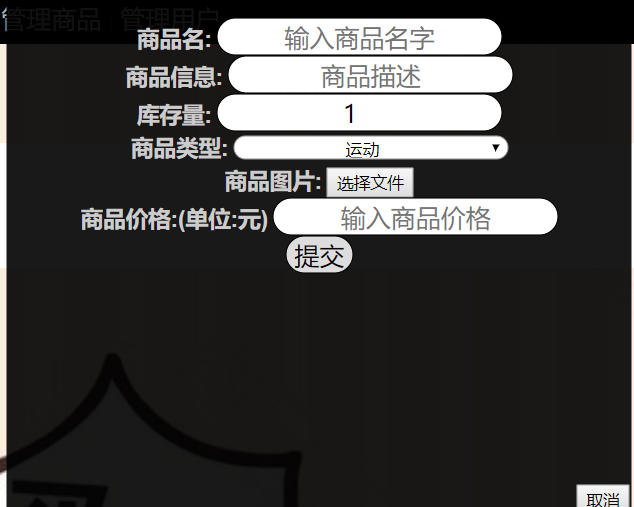


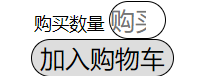
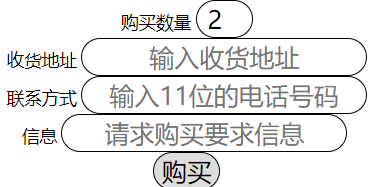






1. 用户上传商品、添加商品进购物车、购买商品都需要填一些具体的资料，且资料均不允许空白。上传商品的资料包括 商品名、商品信息、商品的库存量、商品类型、商品图片和商品单价；添加商品进购物车的资料只要商品的数量便可；购买商品的资料包括 购买数量、收货地址、联系方式、请求购买的信息。



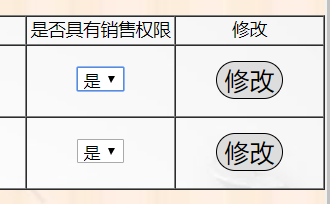


## 4.3 管理查看

1. 管理员对处于允许发售和审核中状态的商品进行查看并可以对其进行修改操作。

2. 管理员对所有普通用户的信息可以进行查看，并修改其用户的销售权限，当修改用户销售权限为**否**时，该用户所有商品都无法在页面上被查询出来，当修改用户销售权限为**是**时，该用户所有商品进行二次上传，需要管理员通过审核。



**修改后查不到此人的商品**



**再次修改后该用户的商品在管理员管理商品页面可被查询到**



## 五 .程序亮点

## 5.1 数据库操作优化

1. 编写通用的工具类代码对数据库进行增删改查。在数据库操作工具类中，分别定

义连接数据库的相关属性，内部定义了update方法，createPreparedStatament方法，

Query方法，close方法，total方法，通过Dao层传入sql语句以及通配符的代理值，

通过调用内部方法得到相关的信息或数据返回给Dao层，这大大减少了Dao层的代

码冗杂度。

2. 合理地关闭了已调用的资源，不会导致程序资源产生浪费的现象的发生。

3. 数据库采取了PreparadStatament的方式对sql语句进行预编译限制了网页的sql注入攻击，提高了后台数据的安全性。

5.2 数据展示

1. 良好的逻辑展示良好的数据，用户可以看到应有的数据展示，比如当鼠标放到商

品上时，该商品的描述信息会显示出来供用户观看，用户可以查看别人信息，也可以查看到自己的信息。

2. 用户查看自己的订单时，完整的订单数据展示在页面中，让用户一目了然。当用

户导出订单下载到本地电脑磁盘时，该文件以Excel格式展示出该用户的所有订单的

数据，数据显示整齐明了。

3. 商品页面商品排列整齐，初始首页商品价格从高排到低，且用户在许多页面中都能进行翻页观看商品数据、订单数据和购物车数据。

5.3 用户体验

1. 良好的产品需要良好的用户体验，该产品理应提高用户的体验，比如前端提示用

户输入正确的数据，后台数据处理进行消息返回通知，相关页面的跳转控制。

2. 当用户查询商品、查询用户、删除商品、删除购物车、添加商品、添加商品到购物车、对订单状态进行修改、购买商品时都会产生相对应的处理结果显示，让用户知晓该操作是否执行成功以及得到的结果是什么。

## 六 .程序难点

## BUG的查找与解决

1. BUG在该项目中通常出现在Dao层、Service层以及Controller层。
2. Controller层获取表单数据时，要找到key相对应的value，NULLPOINT异常的出现也往往是出现在这里，找不到对应的value，一般是在Controller层写错key了，导致找不到对应的value。**解决方法：**耐心查找对应的key，进行Debug操作，看具体是哪个数据出错导致出现了空指针。
3. Service层一些逻辑判断失误，判断的结果对应不上导致出现了不同的处理结果。

**解决方法：**根据具体的逻辑，一步步排查下来，进行重判断。

1. Dao层出现BUG主要是在sql语句的编写和预编译的通配符分配处理不当、关闭资源不及时或者关闭资源失败，导致资源持续开启造成的资源堵塞从而抛出了异常。

**解决方法：**把sql语句放到mysql数据库中进行检验，若成功，则排除sql语句不正确的原因；若失败，则sql语句有问题，先对sql语句进行检查是否有mysql关键字出现，再对字段以及表名进行分析，一步步找到错误的原因；查看资源是否关闭，根据定义位置对打开资源先进行判空再用close方法进行关闭。

## 数据库表的设计与关联

1. 该项目的数据库一共用了4张表。每个表的字段属性比较多，表与表之间还需要相对应的外键关联，对数据操控的不一样，对数据之间相互的处理也不一样。每个表都需要唯一的主键以确保数据不出现重复。
2. 订单数据需要保持完整性，订单中的商品字段是不设置外键的，卖家和买家参考用户表设置外键关联。购物车表的商品、用户字段设置外键关联，商品被删除则购物车中该商品也被删除。商品表对用户设置外键关联，用户不存在则商品也不存在了。外键关联时的更新和删除时都选择了CASCADE。
3. 表与表之间存在的1对多，1对1关系复杂、表内数据的关系都需要谨慎处理。管理员和用户都处于一个用户表中，且管理员对用户的管理是1对多。用户和商品、

管理员和商品、用户和订单、用户和购物车、商品和购物车都是1对多模式。

# 体会总结

## 记录过程

我记得，那个满是工作室面试的三月份一步步压迫着我们这一群大一新生。师兄们都说：“在进工作室这条路上，最好大家都去走一趟，只是可能结果不如意！”我抱着希望不大去经历一场场考试、面试，我最终被定局在了QG的最后一次面试。再后来，我经过了QG工作室的面试，进入了QG工作室的训练营，认识了一群能一起努力的小伙伴，也认识了带我们翱翔、鞭策我们的林旭师兄。

每一次大组作业，可能是我这两个月来比较痛苦的回忆了，因为要写大组作业我把每周的周六周日都放弃了，该实现的需求我都尽力去实现。很多时候写出来的程序有很多BUG，但我还是不依不饶地去debug，C语言的空指针真是处处都是，程序不崩都不行，只能一步步寻找错误并改正。这可能是我写的程序的时候不能专心地去写。程序员想要不出巨大bug，只能全身心地投入去写好代码。每当想放弃的时候，我都在思考着一句很普通的话：“别人能做好的东西为什么你不能做好”，没有谁甘愿服输，这就是我坚持到现在的原因吧！

这个过程我们学习了很多之前都没怎么接触的技术，比如JAVA、Mysql、Servlet和jsp。后台之路不仅于此，我们能学习的还有很多，只是QG训练营时间比较短，我们需要打好基础去学习。林旭师兄教会我们很多，师兄会指导并督促我们去学习，师兄有时候也会说出很暖心的励志话语。

过程很好，我不后悔。

## 技术收获

短暂的时间里，我在师兄的指导下我打好了JAVA基础，也算是入门了JAVA语言。mysql的增删查改的语法学习，让我了解到如何去操作数据库的表。Java和mysql的连接靠的是java中的jdbc，jdbc可以在sql语句对数据库进行增删查改的过程中获取到相对应的结果,结果和java的集合相互结合，就可以从数据库中拿到自己想要的数据了。jdbc可以连接mysql数据库。

关于JavaEE的开发，我学习了jsp简单的网页设计，学会了一些html标签语言和css装饰以及js的一些用法。对于网页和后台的连接传数据，我学会了使用Servlet来得到jsp页面传来的数据。并在Servlet中new其他的类调用这些类的方法进行数据处理。我了解了服务器的应用，并能较熟悉的使用tamcat服务器对网页进行开发。

## 总结心得收获和感言

我很庆幸在大学生活中有这么一段过程是自己在逼自己去学习。做考核那几天，通常会熬夜熬到3、4点，舍友们没参加工作室可能他们觉得我很苦逼，他们觉得这样可以但没必要，但项目不完成我不会甘心，也许旁人不知你心，所以谓之他言。我很清楚地知道自己想要什么。

我经过这次工作室的考核后，我知道了应该如何去学习后台的知识，相比他人我在程序之路上更多了一份坚持。感谢QG工作室，让我找回了那种奋斗的感觉；感谢林旭师兄和华琛师兄，教会了我们用很多工具并且开阔了我们的视野。每一份精彩都来自于自己的奋斗。谢谢！