



**QG工作室项目报告**

学 院 计算机学院

专 业 计算机类

班 级 2

组 别 后台

姓 名 杨旭东

学 号 3124004109

2024年 4月26 日

广东工业大学计算机学院制

目录

[1. 项目简介 1](#_Toc30766)

[2. 设计思路 1](#_Toc24252)

[3. 详细功能设计 1](#_Toc20914)

[4. 程序测试 1](#_Toc7265)

[5. 项目亮点 1](#_Toc19276)

[6. 心得体会 1](#_Toc25747)

# 项目简介

1.项目背景

随着校园生活的便利，学生们常常会有一些物品需要出售或交换。一个安全、方便的二手交易平台可以帮助学生间快速交换不再需要的物品，降低浪费，提高资源利用率。本项目旨在开发一个校园二手交易平台，让学生能够发布、浏览、购买或出售二手物品，包含商品管理、交易管理、用户互动等多功能。

2.项目基本功能

基础功能：

用户系统（注册登录、个人信息管理、信用评分）

商品管理（发布、编辑、搜索、库存安全）

交易流程（订单状态机、支付模拟、评价系统）

管理员模块（用户封禁、商品审核、日志监控）

高级功能：

实时私聊和系统通知（WebSocket实现）

支付密码与验证码（防御性设计）

单例系统统计（systeminfo表动态更新）

# 设计思路

**一、项目背景与选题**

在面对最终考核的项目选择时，我曾考虑过需求技术栈难度最大的五子棋项目，但因加分项，比如训练AI的那个加分项难以完成，最终选择了更加稳定的交易市场项目。虽然交易市场项目是三个项目里业务量和代码量最大的，但好在业务逻辑清晰，且与实际应用场景紧密结合，能够充分发挥我的技术优势，确保项目顺利完成。

**二、项目开发过程(按照时间顺序编写)**

**（一）项目初始化与基础搭建**

1. **需求分析与项目规划**
   * 花费了四小时对交易市场的业务需求进行深入分析，明确了用户、商品、订单等核心模块的业务逻辑，并设计了相应的 SQL 数据库表结构。
   * 在项目构建过程中，发现需要使用 baseServlet，因此花费了四小时学习相关知识，包括反射和注解，成功将其引入项目中，为后续的开发提供了基础支持。
2. **工具类与框架搭建**
   * 考虑到使用 ORM 框架可能过于复杂，我退而求其次，开发了一个 CRUD 通用工具类并引入项目。通过改进工具类，添加了将 xxx\_xx 的命名转换为 xxxXxx 的方法，解决了字段名和实体类变量不匹配的问题，大大提高了开发效率。
   * 引入了 JUnit 进行单元测试，并规范了单元测试文件的存放路径。

**（二）核心功能开发**

1. **数据处理与实体类设计**
   * 在编写实体类时，针对数据类型的选择进行了优化，部分数据采用 double 类型以满足业务需求。同时，针对头像和时间的处理，我采用了 ISO 8601 标准格式生成时间，解决了前后端时间兼容的问题。
2. **Mapper 层开发**
   * 花费了八小时完成了 Mapper 模块的开发，并进行了单元测试。对于数据的删除功能，采用“字段状态”的概念来代替，模拟数据库朔源以及账号存在状态，确保了数据的完整性和安全性。
3. **WebSocket 实现**
   * 花费了两小时部署了 WebSocket 依赖，并学习了相关原理，成功实现了 WebSocket 功能，为后续的私聊功能提供了技术支持。
4. **单例类与系统变量管理**
   * 开发了一个单例类和对应的 Mapper，用于对系统变量（systeminfo）进行增删改查操作。通过调整单例类的初始化机制，确保其在使用之前总是已经被数据库初始化，从而保证数据安全。

**（三）业务逻辑实现**

1. **Service 层开发**
   * 将必做功能和选做功能的业务逻辑相对独立，分别在项目开发的前期，中期，后期开发了不同的service:前期开发了用户，商品，订单，评论的增删改查，中期开发了会话，消息和举报的增删改查以及与系统单例类业务的整合，后期开发了日志，系统通知和用户商品的信誉评价体系。
2. **Servlet 层与前端开发**
   * 开发了登录注册模块，并解决了验证码认证出 bug 的问题。通过 session 技术获取当前用户信息，实现了个人用户数据更新板块，引入了正则表达式进行数据校验，确保了数据的准确性和安全性。
   * 实现了个人商品管理功能，包括订单加载、评论加载、新增商品、更新商品以及查询特定订单评论等，虽然代码量较大，但通过合理的逻辑设计确保了功能的实现。
   * 开发了用户搜索、查看和购买商品的业务，并通过给相关业务上线程锁并随后细化了锁的粒度，保证了订单的安全性。
   * 实现了管理员功能，包括查看系统数据、查询用户信息、封禁和解封禁用户、查看和处理用户举报等功能，提升了系统的管理效率和安全性。

**（四）功能优化与改进**

1. **用户体验优化**
   * 优化了聊天室的 CSS 样式，使其更加美观，提升了用户界面的友好性。
   * 在前端实现了根据用户和商品的“信誉除以流量”值的高度来判断并显示商品和用户质量的功能，使信息展示更加直观。
2. **功能修复与改进**
   * 修复了验证码无法生成和存储的 bug，将验证码 Servlet 整合到 WebServlet 中，确保验证码功能正常使用。
   * 修复了支付密码相关的 bug，并重新制作并成功实现了邮箱发送验证码功能。
3. **系统安全与性能优化**
   * 对被评价页面进行了改进，优化了页面布局和信息展示。
   * 加强了 Cookie 的安全性，防止用户数据泄露和恶意篡改。
   * 实现了商品到指定时间自动上架的功能，提高了商品管理的自动化程度。

# 详细功能设计

1. **数据库**

制作了十个表和对应的实体类，以下是表及其字段信息

**Admin表：'管理员个人信息';**

'索引','昵称','账号','密码','电话号码','邮箱','头像','账号在线和存在状态''角色'

**User表**'用户个人信息';

'索引',‘昵称','账号','密码','电话号码',‘邮箱', '头像','角色','信誉',‘账号状态','支付密码'

'流量';

**Product表'商品信息';**

'索引','商品名','商品描述','价格','库存','分类','商品图片','发布时间','发布状态' '发布者','评分','流量';

**Deal表'交易信息';**

create table deal(

'索引' '支付方式' '交易状态', '交易时间','交易者', '交易商品', '购买数量',

**Comments表**'评论';

'索引', '评分', '评论时间', '评论内容','交易号',

**Log表'日志';**

'索引', '操作内容','操作种类', '操作者','操作时间'

**Session表'会话记录';**

create table session(

'索引', '创建时间', '先发起会话的人', '先手角色', '后发消息的人', '后手角色'

**Sentence表'私聊语句';**

'索引', '发表时间', '语句内容', '聊天者', '聊天者角色''隶属会话';

**Report表'举报记录';**

'索引', '举报时间', '举报种类', '举报内容','举报结果''举报者'; '举报对象';

**Systeminfo表'系统对象';**

'单例系统ID','单例系统标识''管理员数量','用户数量', '商品数量', '吞吐金额总量以w为单位', '举报数量', '会话数量', '订单数量', '评论数量'

**表之间的关系：**

admin和user表是账户的不同种类，而user可以生成不限的product，在product里衍生出不限的deal，deal衍生出对应的唯一的comment。

User可以生成不限的session，一个session衍生不限的sentence

User可以生成report，Admin可以处理report

以上所有表的业务都会经过systeminfo的统计

1. **基础业务实现讲解**

以上表通过CRUD通用工具类和数据库连接池建立mapper层，实现数据的底层的增删改查。然而大部分表都不设删除的代码，取代以一个叫做status的字段来模拟数据的状态，作为实现用户和商品封禁和解封，订单的送货和退货的业务层面的支撑

**接下来写service和servlet层，下面重点介绍相对复杂的业务**

**注册**

前端传入角色，用户名和密码，后台根据一个在service层的叫selectSingle的函数，可以根据账号，邮箱，电话或者ID搜索唯一的账号，查找是否有存在账户，若否则创建账户，同时systeminfo的对应用户+1，log增加一条注册日志，自动创建账户和系统之间的会话，系统给用户发送欢迎消息

**技术难点和解决方案**

注册需要用户输入验证码防止被脚本自动注册来攻击服务器，但我生成不了验证码，研究之后发现我的验证码工具类的方法出现了没有关闭字节输出流的异常，又研究了很久引进了使输出流关闭的语句解决了这个问题

前端没办法刷新验证码，发现是浏览器缓存导致没办法正常刷新的问题，所以我在刷新验证码的方法那里给异步加载的地址加了个时间戳让图片强制刷新

**登录**

前端可选择账号登录，邮箱登录或者电话登录，按下登录按钮后传输对应数据到后台，如果后台用selectSingle函数搜索到用户就判断前端传输的‘password‘是否契合账号，如果是则判断前端传输的‘remember参数是否为true，如果是则‘将账号名和角色输入到浏览器的cookie里，然后确认登录后，后台获取session将用户名和角色作为参数输入作为登录后业务的一个凭证，log加一条登录日志

**在登录界面加载cookie**

如果浏览器的cookie有数据，那么获取cookie的账号并使用selectSingle函数进行搜索，如果存在这么个账号则向前端传用户名，邮箱和电话，密码不进行传输。

**找回密码**

前端输入邮箱，按下获取验证码按钮将使该按钮冷却一分钟，同时后台获取邮箱号，并以我的qq邮箱作为发送者，给对方发送验证码。对方得到验证码后可以输入新密码，确认重设密码后在后台判断用户输入的验证码是否符合系统发送的验证码，若是则操作成功

**技术难点以及解决方案**

在引进发送邮箱的函数时发现得开启发送者的邮箱的对应的stmp服务，对此我学习了怎么管理我邮箱的stmp服务，而且发现outlook邮箱正在淘汰相关的服务，真是麻烦，所以我所以了qq邮箱。

**加载账号，用户和管理员的权限管理**

在登录后进入一个页面，自动加载LoadAccount的方法：用登录后在session写入的用户名数据搜索账号，将搜索到的账号传到前端。

然后编写管理员过滤器和用户过滤器，对应的过滤器触发条件是用户进入前台或后台并触发了对应的servlet方法，而过滤器会获取当前session的角色，如果角色和进入的页面不符则拒绝调用对应的servlet，实现了用户权限隔离的功能，而如果用户没登录则session没有数据，那么将加载不了登录后的页面

如果没有登录或者对应权限的但进入登录后界面，那么可以点击网页的右上角重新登录

**进入登录界面/退出登录**

登录后右上角有个按钮，专门进入这个页面，如果你已经登录了但你又点击了这个按钮，那么后台帮您把当前的session销毁掉实现登出的业务

**更新账户，商品及其头像**

以用户为例，此处不再赘述商品更新

普通用户在没有完善个人信息时是没办法购买物品和进入聊天室的，意味着他们不能使用网站的大部分功能。所以用户有必要完善个人信息。

用户在更新个人信息时首先要输入自己的密码来表明自己就是本用户，确认身份后你可以改你的头像，登录密码，支付密码，昵称，邮箱和电话，其中昵称，邮箱和电话是必填项，其他两个是选填项，如果你没有支付密码你将无法购买物品，没有昵称你将无法进入聊天室

**技术难点和解决方案**

更新头像时要存储头像，怎么存储呢？用户有专门的存储头像的字段，当上传头像时在项目根目录的file文件夹里生成一个以**UUID**（通用唯一识别码）为名的图片，存储头像的字段存储该图片的名字。然而我为实践这个方法花了很大功夫，比如我实验了很久才意识到不能在前端的img块显示绝对路径的图片，而且当tomcat部署和未部署时根目录会随之改变。

**发布商品**

当发布商品时得填满商品的所有数据，成功添加商品后systeminfo的商品数+1，log记录商品日志

商品可以设置马上发布和延期发布，当延期发布时可设置未来的某个时间发布商品，商品会在时间到达时实时发布，届时系统会通知商家

**难点和解决方案**

怎么实时发布延期发布的商品呢？我引进了quartz框架，编写job类来搜索未发布商品，获取数据并进行时间判定，如果到了指定上架时间将更新商品到发布状态，然后写触发器，设定一分钟触发一次job类

**用户下单**

当用户选中并下单时他得填要购买的数量和支付密码，经过检验后生成订单，商品和商家的总流量和总评分+1，systeminfo的订单数+1，库存数-productnum总流动资金加上price\*productnum/10000，log记录订单日志。系统自动通知卖家订单通知

**难点：**下单，评论和撤评不是最难的业务，但是是最复杂的业务，因为这三个业务与商品和商家的评价息息相关，构成了商品和商家的评价体系，代码量相当大。稍不注意这个体系的数值计算就会出错

**用户评论和撤评**

当用户评论时生成一条评论。我制作了一个select函数实现了一个订单只能对应一条评论，防止有人刷分。评论分为评论内容和评分，评分有-2分，0分，2分。当评论时systeminfo的评论数+1，系统通知被评论者评论通知，商家和用户的总评分会根据评论者的评分影响，log记录评分日志

当评论被撤销时，数据库删除评论，systeminfo的评论数-1，商家和用户的总评分会根据评论者的评分反影响，

**更新订单状态**

商家可以让订单状态从“待发货”到“在发货”到“待收货”

用户在订单“待收货”后收货然后变成‘已收货’

用户对订单不满意时可选择退货使订单状态‘已退货’，systeminfo订单数-1，商品库存+productnum，总流动资金还原，如果已经评论了将撤回评论，商家和商品的评分和流量会被影响。

系统会为订单状态的改变给商家或用户发消息

**举报**

用户进行举报是可选择举报商家或商品，然后写举报内容，提交后将举报交给管理员审核，用户自动给系统发举报消息，systeminfo举报数+1

**封禁**

管理员可在后台的查询商品界面和查询用户界面分别手动对商品或用户进行封禁

管理员可在后台的举报审核界面自动进行封禁，通过批准或反对举报，处理后系统给举报者和被举报者发消息，若管理员批准举报，系统自动对用户或商品进行惩罚

**创建会话**

怎么创建和另一个用户的对话呢？我的答案是：在购买商品的界面时有一个“和商家私聊”的按钮，点击后携带自己和对方的消息组成新url，跳转到聊天室，解析url，如果当前用户和商家之间没有对话就创建对话，创建对话后当前用户自动给对方发送欢迎消息。

同理对方也会自动对你发起会话。

创建会话后systeminfo的会话数+1

**发信息和接收消息**

用户进入聊天室后系统广播给所有在聊天室内的用户提示该用户进入聊天室的信息

用户接收其他用户的消息而且用户在聊天室里的时候将会在广播块收到新消息通知

用户可以手动给系统管理员外的用户发消息，发信息后自己和对方都会刷新会话的消息列表，同时数据库添加新消息使离线用户也能接收消息

**难点和解决方案**

怎么做到实时发送消息，我学习并引进了webSocket依赖并进行测试

怎么做到区分系统信息，新消息通知和私聊信息？我从前端给后台传递信息时给信息进行包装让后台识别是什么类型的信息，然后执行不同的消息推送逻辑

当websocket对象生成时获取当前用户的httpsession，将httpsession的参数传入websession实现识别用户的功能，进而实现私聊功能，将消息推送给指定用户。特别注意的是当对方是系统管理员时将特别地执行另一套逻辑来实现私聊效果，因为这个系统管理员是虚拟出来的，并不是真正的用户。

1. **松散业务实现讲解**

**保证订单安全**

当卖家更新商品，买家购买商品时可能多线程并行，使商品库存到负数

为了防止这种情况，我给商品更新的方法和下单的代码块都上了线程锁

锁对象是系统单例类，虽然锁粒度相对大，但我找不到更好的对象作为锁了，而且这个锁是更加安全的，毕竟整个系统就一个系统单例类

**日志记录**

每当用户有特定操作时记录用户的IP，用户名，操作种类，操作内容和操作时间。

**难点和解决方案**

这个业务是很松散的，所以要在很多业务里插入效果语句记录日志

因为要记录用户的IP，所以我引进了个IP工具类，它的原理是解析request对象来获取IP

**系统和用户之间的自动通信**

编写了两个方法，分别实现了系统给用户发信息和用户给系统发信息的功能

同时用户进行一些操作时可能会触发这两个方法

**用户和商品评分系统**

当下单时卖家和被购买商品的总流量和总评分+1

当评论时卖家和被购买商品的总评分变化

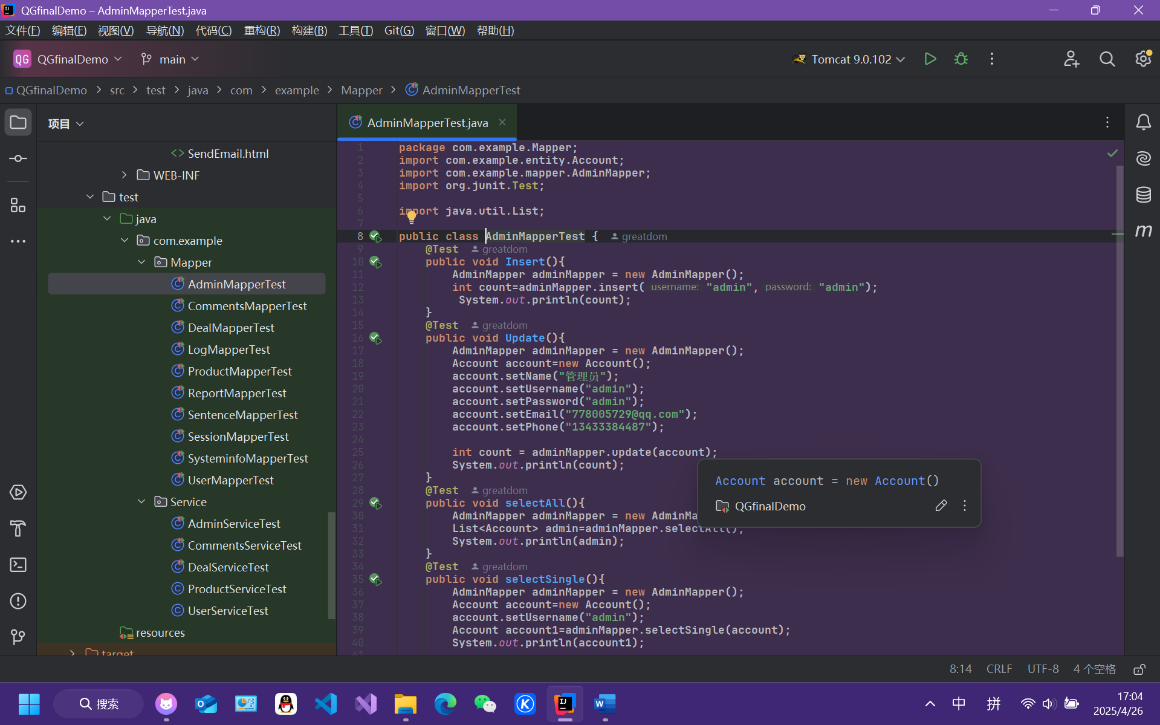
当评论被撤销时卖家和被购买商品的总评分还原

理论上，当用户和商品从未被评论时，他们的总评分/总流量的值等于1，

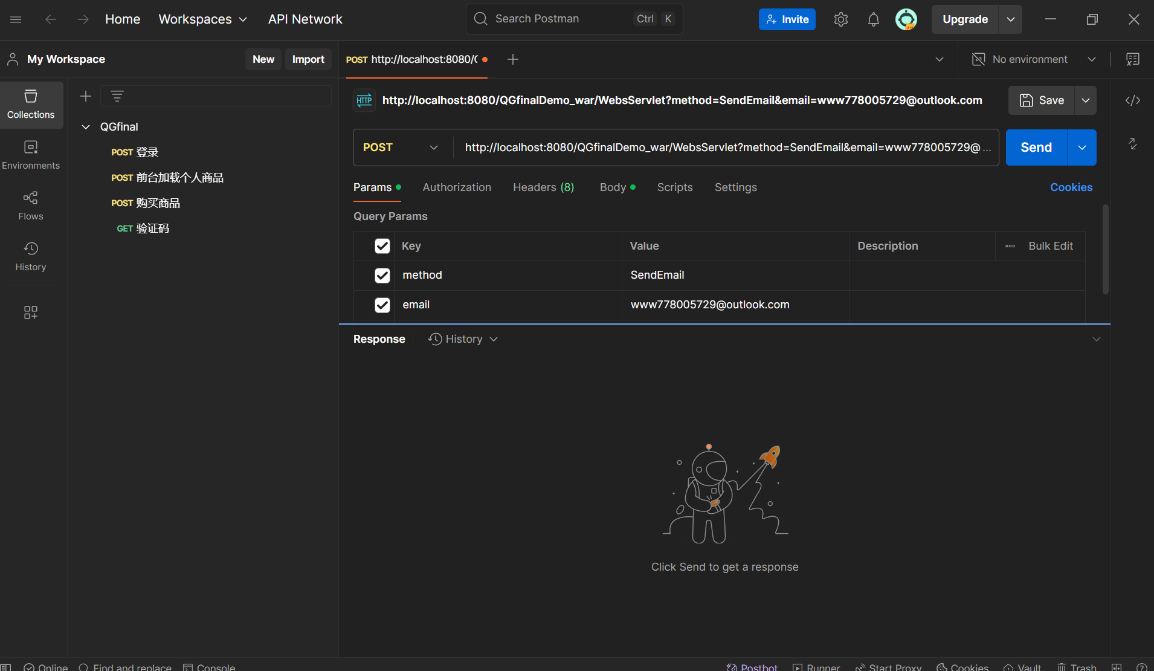
整个值越高那么他们的信誉越好，反之越差。

# 程序测试和前端页面一览

在github仓库里有基于junit库编写的mapper层和service层的测试，经过测试已经修复了bug



在postman里我曾经测试了一些出了bug的接口，发现在postman里发现的bug的反馈更精准



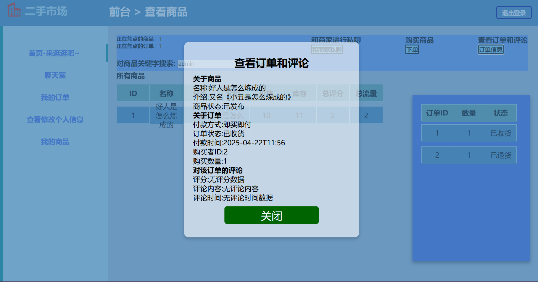
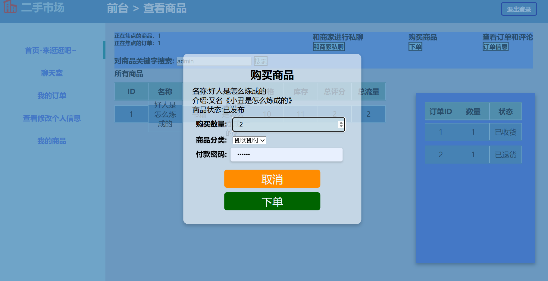
没有登录时是没办法加载信息的



由登录界面可以进入注册界面和找回密码界面，在浏览器有cookie时传入账号，邮箱和电话



用户在首页可以搜索和购买商品，然后可以看商品信息



用户可以查看自己的订单，进行评论，撤评，接收货物，举报



用户可以发布和更改商品，查看订单和对应的评论，更新物流状态



用户可以进行私聊



更改自己的个人信息



后台查看用户



后台查看日志



后台处理举报



后台查看系统数据



# 项目亮点

**1. CRUD通用工具类以及自研数据库连接池的研发和使用（通用加分项）**

* **技术突破**：
  + 通用工具类动态解析实体类与数据库表的字段映射，以及设计出将带下划线字段转换为驼峰命名法的方法，减少了70%重复CRUD代码。只需将自写的sql语句和类参数导入工具类接口即可实现数据的增删改查
  + 自研数据库连接池通过properity文件减低了连接池参数的维护成本，提供设计接口时开发者更方便地调用conn对象。
  + 将数据库连接池的接口应用到CRUD通用工具类，进一步降低代码量，提高了开发效率
* **实际应用**：
  + 在商品发布、订单管理等模块中，简化数据操作逻辑，显著提升开发效率。
  + 极大节省了服务器的调用conn对象的时间，提高服务器性能

**2. WebSocket实现用户间通信，并结合了举报与举报处理功能，系统根据用户行为自动通知用户功能（选做项实现）**

* **功能亮点**：
  + 实现用户间点对点聊天，支持离线消息存储（**session**，**sentence**表）与登录后推送。
  + 用户可以对商家或商品进行举报，管理员在后台可以处理举报，如果举报被批准，那么被举报对象自动被惩罚
  + 实现用户和系统管理员之间的交互，用户的部分操作(如举报和得到举报结果，评论撤评，下单退单等相关操作)会让系统**自动**给用户发消息或用户自动给系统反馈
  + 在聊天室里当接收到其他普通用户的新消息时收到新消息通知
  + 通过用户ID绑定Session，解决多浏览器会话冲突问题，确保消息精准推送。
  + 在前端或后端之间传递消息时自动对信息进行格式包装以区分广播，新消息通知与私聊，避免会话混淆。

**3. 动态系统统计模块（需求未提及的创新功能）**

* **功能设计**：
  + 单例类**SystemInfo**实时统计平台数据（用户数、交易额、举报量等），并持久化至**systeminfo**表。
  + 管理员可在后台查询系统数据
  + 调用单例类的初始化安全问题，为此具体问题具体分析，我设计新的算法保证调用安全
  + 单例类与数据库交互的实现，由于单例类不兼容我的CRUD通用工具类的通过反射调用类的算法只能为其手搓Mapper。

**4. 日志管理与操作溯源（通用加分项）**

* **功能实现**：
  + 操作**日志**全记录（**log**表），涵盖用户登录、商品发布、订单支付等行为。
  + 调用自研**IP工具类**以得到操作者IP，调用自研时间工具类获取时间，调用常量类获取日志种类，得到更全面科学的日志
  + 管理员可查看日志

**5. 时间格式兼容前后端，创新性实现头像存储，并实现商品定时上架（选做项实现）**

* + 统一采用ISO 8601标准时间格式（如**2025-04-22T11:56**），后端**LocalDateTime**解析，前端格式化显示。
  + 在数据库中，用户有专门存储经过处理的图片名的字段，在项目的根目录里成立一个文件夹存储图片，在前端的一些界面显示图片，使用户更直观
  + 前端可设置是否延期发布商品，若是则设置在未来的某个时间上架
  + 后台引进Quartz定时任务框架，服务器开启时用Listener自动调用任务。每隔一分钟检查未发布商品是否满足发布条件，若已经到达发布时间则发布商品，同时系统在聊天室自动给发布者发消息
  + 解决前后端时间解析差异，避免因时区或格式导致的业务逻辑错误。

**6. 代码质量与工程规范（通用加分项）**

* **实践成果**：
  + **JUnit**单元测试覆盖Mapper和Service层核心逻辑，确保业务健壮性。
  + 前端的数据传入后台前经过正则表达式检验
  + 用**postman**测试出bug的接口，提高了测试效率
  + Maven多模块管理 + Git小粒度提交，代码可维护性显著提升。
* **规范落地**：
  + 统一响应封装（**Result**类） + 常量类管理，代码风格高度一致。

**7. 验证码，拦截器（通用加分项）**

* + 后台通过引进验证码工具类生成随机验证码字符串和验证码图片，并通过servlet发送验证码图片给前台，用session存储字符串
  + 前台显示验证码图片，用户输入验证码后通过异步请求，调用session进行数据匹配达到验证码的目的
  + 设置了身份验证拦截器，拦截器读取登录后用户信息的session，如果用户调用的异步请求和用户的角色不相符则不允许请求，实现了用户权限的区分

**8.科学的支付流程，通过邮箱发送验证码找回密码（选做项实现）**

* + 设置了购买数量，当购买商品时根据需求购买；设置了支付密码，保证了用户钱包的安全性
  + 引进了邮箱发送验证码的工具类，但发送验证码就要打开邮箱的stmp服务，实践证明outlook邮箱的基本认证正在逐步淘汰，但qq邮箱可以实现这样的服务，所以使用我的qq邮箱作为验证码的发布者，同时打开了我的qq邮箱的stmp服务
  + 前台设置了点击发送验证码按钮后得冷却60秒，防止用户频繁请求验证码

# 心得体会

开发项目好难啊，从一开始到收尾都要面临不同的问题。刚开始要面临分析业务和规划项目问题，中期要面临学习新技术，整合功能，调试和改bug的问题，后期要面对海量代码，稍不留神就会思路错乱的问题。真是麻烦呢。你知道吗？对于一个没写过项目的大一学生来说两个星期内要写完项目的必做项还有选做项，同时要兼顾通用加分项难度太大了，无异是与时间赛跑，但终于还是将项目完成了，而且功能还挺丰富的，这给我的成就感是很大的。欧耶，真是美妙呢！

然后写代码的时候每写一个小功能，就要进行一次测试，从开始写到写完就要测试几百上千次，就是为了不出现不可控bug。因为我知道一旦出现了连环bug，就是不同的板块都有bug，是很难改的，然后我就会破防，每次改bug超过一个小时我就会破防。不过还好由于我经常测试的习惯，就很难出现大bug，半小时内能修好。这是我两星期内能写完项目的只要习惯基础，所以我写代码可能很慢，但产出还是很稳定的。

然后写代码的时候我还是很注重前端的颜值的，我要让前端长得更好看一些，我已经尽力了啊啊啊啊啊！

真是麻烦呢，现在是公元2025元4月26号，我完成了项目，现在闲的要死，看到好多人都放弃了或者没写完，我就在想要不要用一晚上的时间精进一些功能呢？但是我已经累了，我从开学到现在天天在学习新技术，没有娱乐，好累啊，前所未有的累啊，今晚我要好好放松了欧耶！

现在在想一个问题，其他人会不会也和我一样写完了选择项呢？哎呀，感到焦虑了！

------2025.4.26.17.41