



**QG工作室项目报告**

学 院 自动化学院

专 业 物联网工程

班 级 22级物联网工程2班

组 别 软件组后台

姓 名 尹俊标

学 号 3122001417

2023年4月30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)

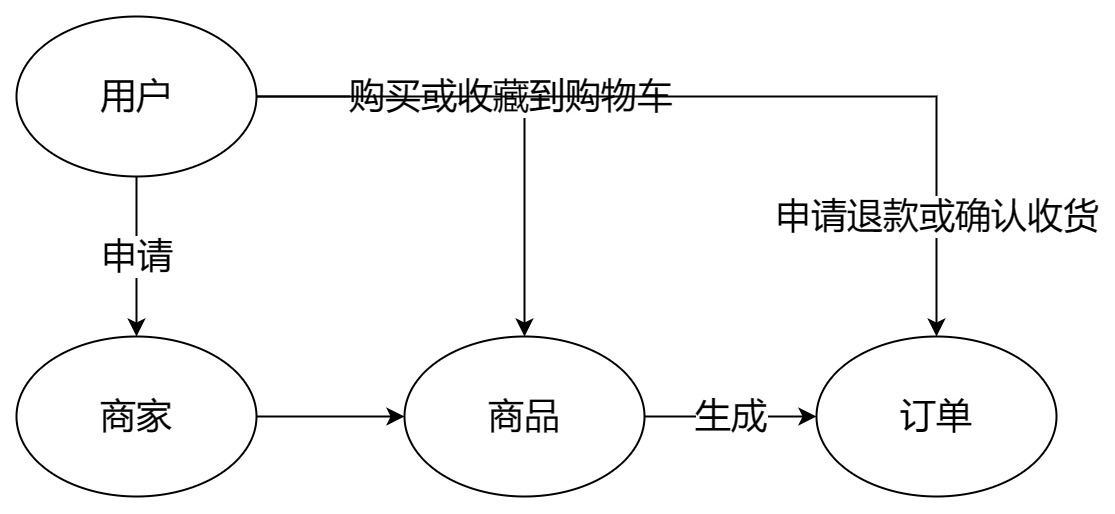
[6. 心得体会 1](#_Toc25747)

# 项目简介

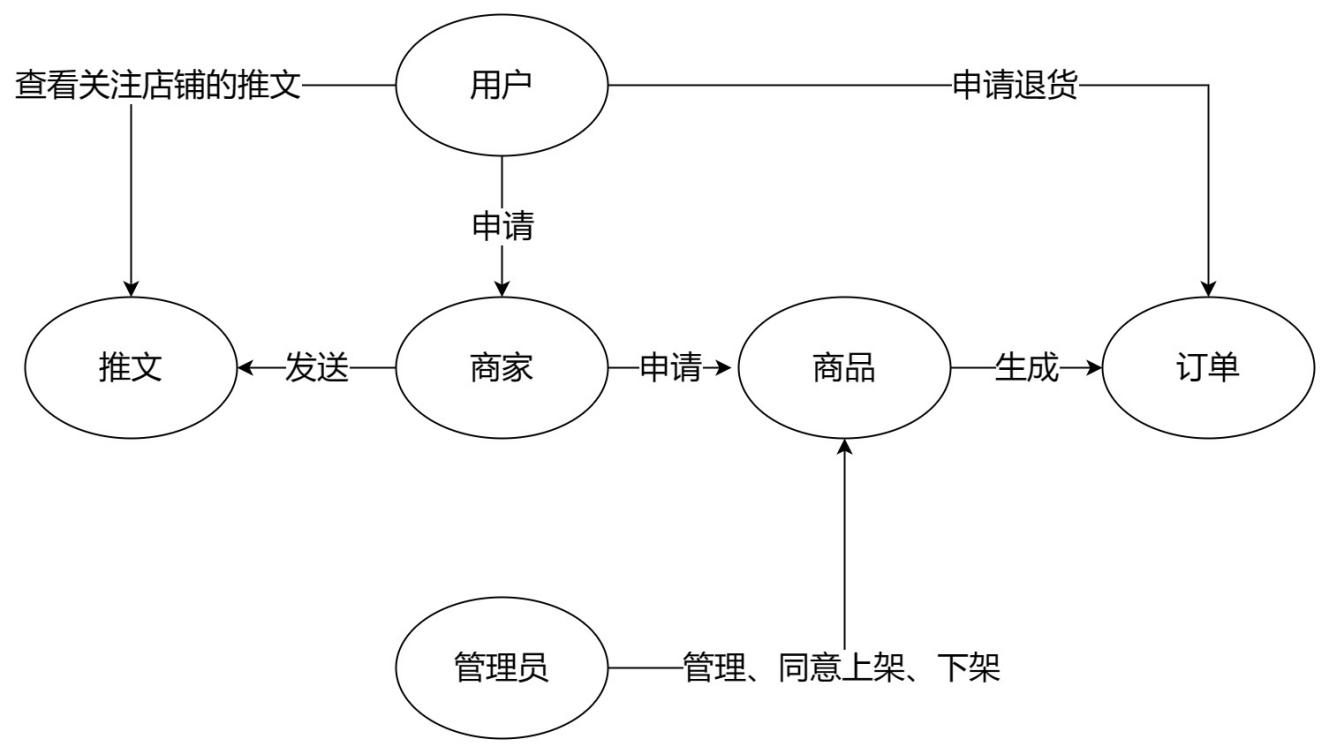
题目：QG购物网

1. 项目概述：项目为一个基于Web技术的在线购物平台，包括前台网站和后台管理系统，前台网站为用户提供商品浏览、购买、收藏、支付、搜索等功能，后台管理系统为管理员提供订单管理，商品管理、订单管理、店铺管理、用户管理等功能，以满足多用户的购物需求和管理员的管理需求，提供安全，高效，方便的购物体验
2. 技术架构：项目采用了以下技术架构：
   1. 前端技术：HTML、CSS、JavaScript、Vue、Element等，
   2. 后台技术：Java，MySql，Tomcat，Servlet，Filter等
   3. 开发工具和环境：IDEA、Maven、Git、阿里云OSS等
3. 功能介绍
   1. 用户注册和登录：用户可以在前端网站进行注册和登录，可以通过手机号修改密码
   2. 商品的浏览和搜索：用户可以浏览商品列表，还可以根据商品名称和介绍、所属店铺名称进行模糊搜索，用户可以直接购买搜索出来的商品
   3. 商品详情和评论：用户可以对商品进行评论或提问，也可以回复其他人的评论或回复
   4. 购物车：用户可以将商品加入购物车，加入购物车后可以查看购物车内的商品进行购买或删除
   5. 订单管理：用户可以查看自己购买商品的订单列表，可以查询订单状态，对于正在发货的商品可以确认收货或对未申请过退款的订单申请退款等操作
   6. 商品管理：管理员可以审批商家申请上架的商品，决定是否可以上架、也可以对违规商品进行下架处理，商家可以上传商品的介绍与图片，用户可以查看商品介绍、图片、销量、库存等信息，还可对违法商品进行举报，等待管理员审批，管理员可以根据用户描述决定是否要下架商品
   7. 用户管理：用户可以自行修改基本信息，也可以通过名字查询其他用户，用户可以设置私密防止他人通过名字查找到自己
   8. 商家管理：普通用户可以申请成为商家，管理员审核通过后可以成为商家并申请上传商品，管理员审核后可以上架，店铺还可以发送推文吸引顾客，用户关注店铺后可以看到店铺发送的推文
   9. 管理员功能：管理员可以查看用户的店铺申请并根据申请内容决定是否通过用户的请求，也可以查看商家的货物上架申请并根据货物信息决定是否通过请求，还可以查看用户的举报记录并决定是否下架货物，管理员还可以自行查询商品，如果发现有违规的货物可以直接进行下架，对于违规的评论及回复可以直接删除

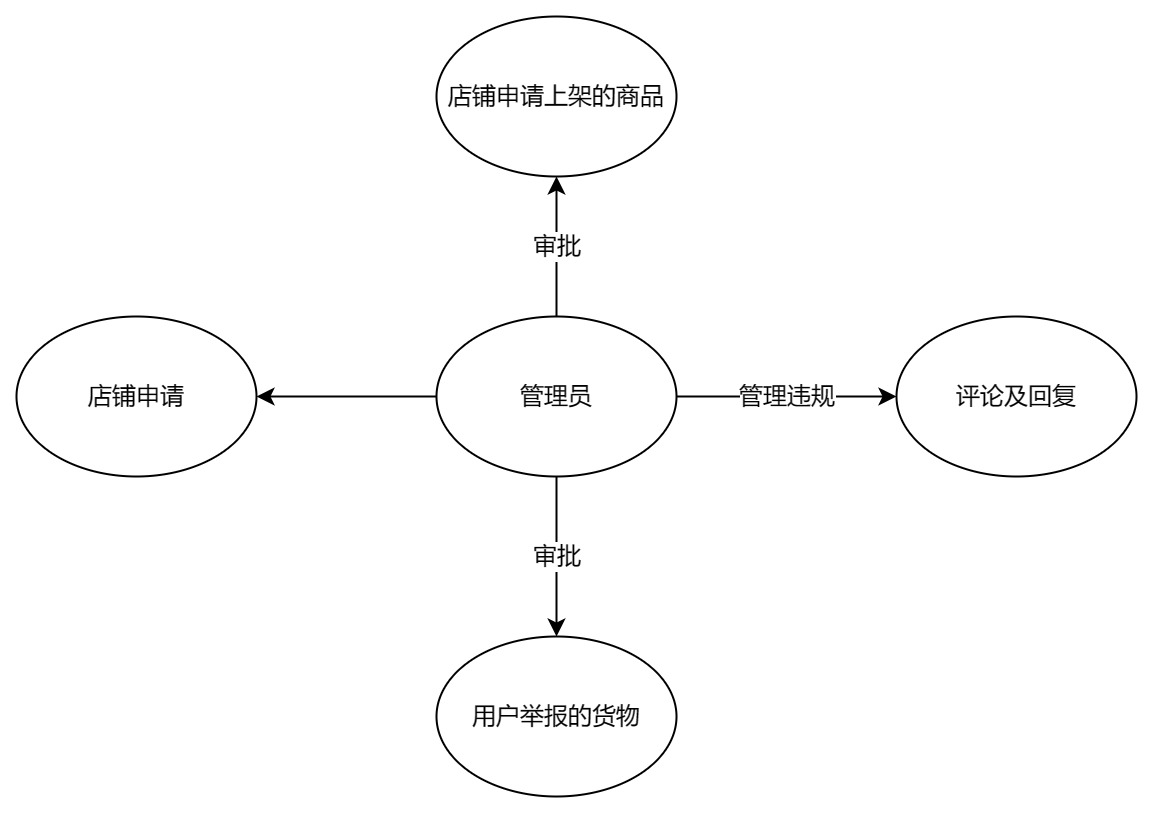
用户消费关系图



商店管理订单



管理员功能



# 设计思路

一．需求分析：

1.商品展示：展示商品的名称、价格、介绍和图片等信息以供用户查询

2.购物：用户能在购物网站上浏览并查找商品，并且能够方便快捷的完成购买流程，网站需要提供安全保障，确保交易的安全性

3.购物车：用户可以将商品添加到购物车中，后续可以删除或进行批量购买

4.订单管理：用户购买货物后生成订单，也可以查看自己的订单，包括未发货、已发货、已收货、退款中、退款成功、退款失败等状态，用户还可以对购买的商品申请退款，但只能申请一次

5.支付功能：由于没有支付的接口，所以网站采用简单确认支付功能

6.举报功能：用户可以对违规商品进行举报，待管理员审核后对商品进行下架，商品下架后会提醒店铺

7.评论和回复：用户可以在商品下进行评论或询问，其他用户和商家可以进行回复，管理员可以删除违规的评论和回复，评论和回复被删除后用户会受到提醒消息

8.推文：店铺可以发布推文以吸引粉丝，用户关注店铺后可以查看店铺发送的推文

9.关注：用户可以关注店铺，店铺发送推文后可以查看到

10.个人信息：用户可以自己修改基本信息，还可以设置是否公开被其他用户查询

11.上架商品：商家可以向管理员申请上架货物，管理员同意后可以用户可以查看到商品进行购买

12.退款审核：商家可以根据用户申请退款订单的描述决定是否退款

13.管理员：管理员可以查看用户的商家申请并根据信息决定是否让通过用户的申请，商家可以上传商品上架申请，管理员根据商品内容决定是否通过，用户举报商品后，管理员可以查看举报记录并决定是否将商品下架，还可以自行搜索下架商品或者删除违规的评论

二.系统设计

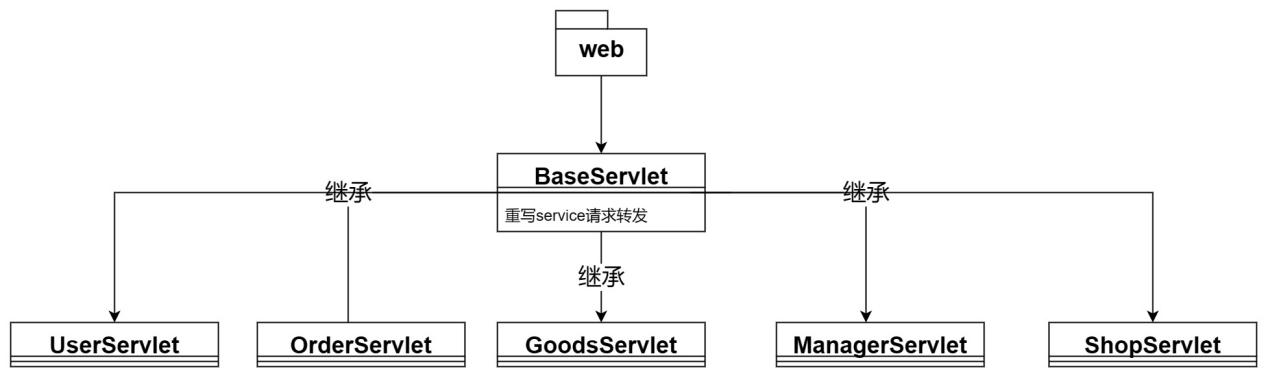
1.用户模块：包括用户的注册、登录、退出登录、忘记密码、修改和查看头像等基本信息，将商品添加到购物车、查询用户自己的购物车、购买或删除用户自己的购物车中的商品，查看自己的订单，申请一家店铺，搜索其他公开的用户，关注店铺，查看或取关用户关注的店铺，查看用户关注的店铺发送的推文，对违规商品进行举报，查看和删除系统的提示信息

2.商店模块：查询店铺未发货、已发货、正在申请退款的订单，将订单从未发货设置为已发货，查询申请退款的订单，同意或拒绝退款申请，申请添加商品，发送推文，查看店铺正在申请上架的货物并上传图片，发送、查看和删除推文，查看和删除店铺的消息，用户根据店铺名称查询店铺，查询店铺的商品

3.订单模块：用户购买商品后自动生成订单，用户将自己的订单从发货中设置成已收货，用户申请自己的某个订单退款

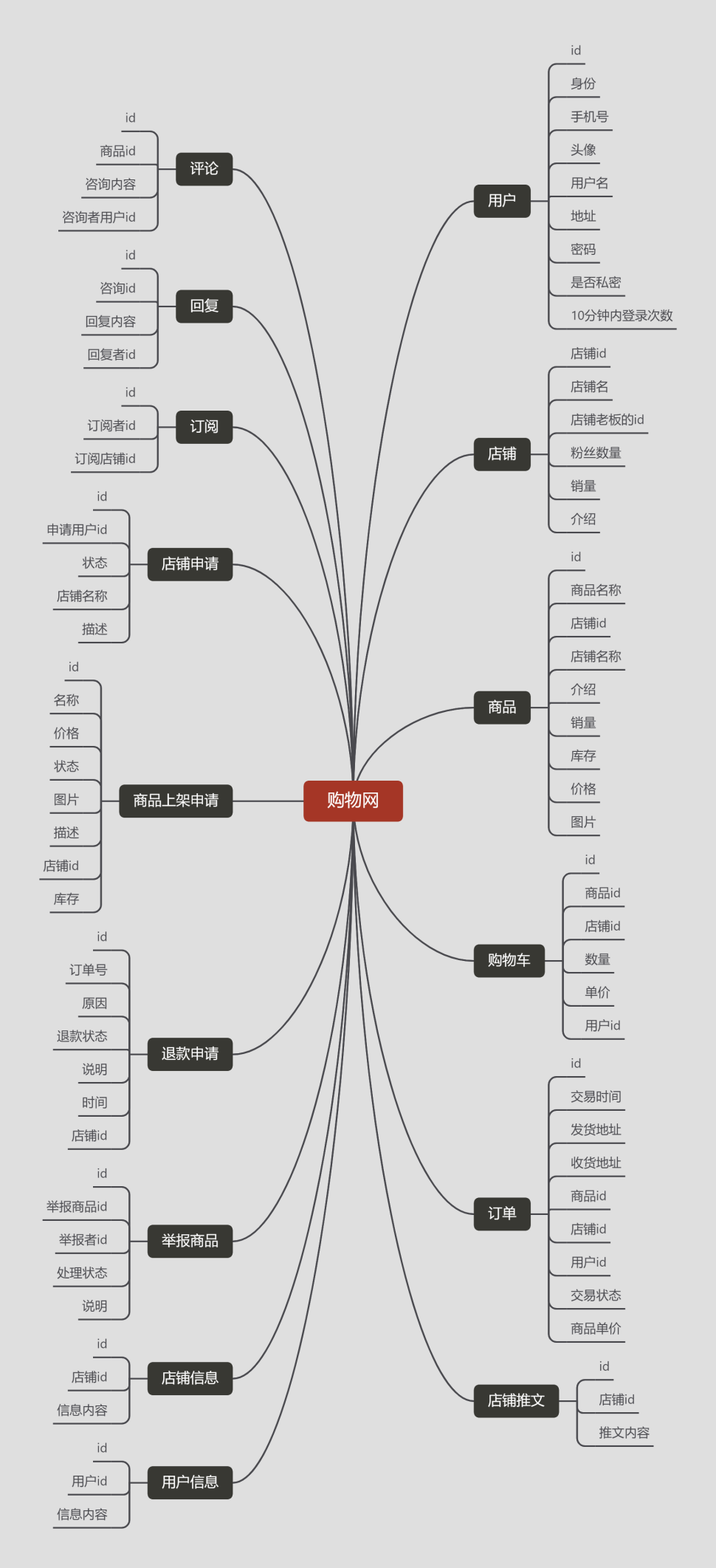
4.管理员模块：进入管理员界面权限判定，查看用户申请店铺并同意或拒绝店铺申请，查看店铺推送的商品并同意或拒绝商品上架，删除违规商品，删除违规的评论及回复，查看用户举报的商品并同意或拒绝商品下架

5.商品模块：展示所有的商品、按照商品名称和介绍模糊搜索商品，查看商品下的评论及评论的回复，发送商品的评论和回复



# 详细功能设计

一．数据库表的设计



二．用户模块

* 1. 注册：以手机号为标识，用户需要输入手机号、用户名、密码、地址等信息进行注册，运用了正则表达式匹配手机号，必须要是全部数字且开头为1的手机号才能进行注册，为了防止用户误输入密码，使用双重校验，两次密码一样才会进行注册，同时还有验证码验证是人进行操作，手机号是唯一标识，传到后台时会先验证手机号有没有被注册，只有没有被注册的手机号才能注册成功，密码传输到后台后会先进行加密传输再传输到数据库中
  2. 登录：以手机号作为登录标识，传输到后台中根据手机号查找对应用户，在匹配用户输入的密码加密后是否跟数据库的密码一样，如果一样则登录成功，登录失败后会记录到数据库中，10分钟内登录超过5次则不能再登录，登录成功后会返回带有用户id的jwt令牌，前端将jwt令牌存储在本地缓存中，同时后台也会将jwt令牌设置为cookie保存
  3. 忘记密码：输入手机号作为标识，（因为没有短信验证码资源，所以用图形验证码模拟），然后输入要修改的密码，只有当密码与原密码不一样时才会更改
  4. 查看和修改基本信息：前端将jwt令牌设置为 “Authorization”请求头，后台解析请求头的jwt令牌后可以获取用户的id，并进行搜索，返回用户的基本信息或将用户传输的信息更改到数据库中
  5. 添加商品到购物车：前端将商品信息传输到后台，后台读取请求头中jwt令牌的用户id和商品信息后，先搜索用户有没有添加过该商品，如果添加过则上锁后将原来的购物车商品的数量加上要添加的数量，如果没有提交则加锁双重判断后进行插入
  6. 查看购物车：后台代码读取请求头jwt令牌中的用户id后从数据库中查询该用户id的内容并返回给前端展示给用户
  7. 购买或删除购物车中的内容，前端将选中的多件或单件商品以数组的形式发送给后台，后台接收得到的数组，并进行购买或删除操作，购买时会先判断商品库存是否充足或商品是否属于自己的店铺，如果是自己店铺的商品或内存不够则不能购买，返回失败，否则购买成功，删除购物车中选中的内容并生成响应的订单
  8. 查看自己的订单：前端将要查询的订单类型传输到后台，后台读取请求头jwt令牌中的用户id和类型，并通过id查询响应的订单返回前端
  9. 申请开店：用户填写申请的店铺名称和店铺的介绍，后台接收用户填写的信息并获取用户id，查询数据库中是否存在该用户的正在审核中的申请，如果有则返回失败，如果没有则加锁双重查询后插入
  10. 查询其他用户和店铺：前端将用户输入的内容传输到后台，后台通过模糊搜索找到对应的用户和店铺，将敏感信息去除后返回给前端
  11. 关注店铺：前端将店铺id传输到后台，后台接收到店铺id和用户id后在数据库中经加锁双重查询后插入对应数据
  12. 查询和取关店铺：查询用户id所对应的映射关系，并将对应的店铺信息返回前端，前端将店铺的id传给后台，后台再根据店铺id和用户id删除关注映射
  13. 查看关注店铺的推文：后台根据用户id查询关注的店铺，再从关注的店铺中查询推文，并按时间降序排序
  14. 举报商品：前端将商品的信息返回到后台，后台根据用户id和商品id查询用户是否有正在审核的举报，如果没有则直接插入，如果有则插入失败返回信息
  15. 查看和删除系统的提示信息：根据用户id查询用户提示信息，根据信息的id将信息删除

1. 商店模块
   1. 查询店铺订单：前端返回要查询订单的状态码和店铺的id，后台根据状态码和店铺的id查询对应的订单并返回
   2. 订单发货：前端先判断订单是否处于未发货状态，如果是未发货的则将订单信息发送到后台，后台根据前端发送的信息将订单状态从未发货更新为已发货
   3. 查询申请退款的订单：前端发送申请退款中的状态码和店铺id到后台，后台根据状态码和店铺id查询订单并返回前端
   4. 申请上架商品：用户填写申请上架商品的信息，再把商品信息和店铺id传输到后台，后台查询是否有相同的记录，如果没有则直接插入，如果已存在则插入失败
   5. 发送推文：用户填写对应的推文内容并和店铺id一起发到后台，后台把信息直接插入到数据库中
   6. 查看店铺的正在申请上架的货物：前端将店铺id发送到后台，后台根据店铺id从数据库中查询对应店铺id正在申请的数据返回前端
   7. 上传或修改申请上架货物的图片：前端发送申请的信息和图片的二进制数据到后台，后台将图片发送到阿里云OSS并将url存储到数据库中
   8. 发送推文：前端将店铺id和推文内容发送到后台，后台直接插入到数据库中
   9. 查看和删除店铺发送的推文：前端发送店铺id，后台根据店铺id查找对应的推文，再把推文信息返回到前端，前端可以发送对应推文的信息到后台进行删除
   10. 查看和删除店铺的信息：前端发送店铺id到后台，后台根据店铺id查找对应的信息，前端再把要删除信息的id发送到后台，后台直接删除
   11. 查询店铺：用户在前端界面根据用户输入的名字进行模糊搜索查找对应的店铺
   12. 查询店铺的商品：前端发送店铺的id，后台根据id搜索对应的商品并返回展示给前端
2. 订单模块
   1. 购买商品：前端返回用户要购买的商品id、数量，后台根据商品id、数量、用户id等生成订单，并插入到数据库中
   2. 确认收货：前端返回要确认收货的订单信息，后台判断订单是否处于已发货状态，如果是则更新为已收货状态
   3. 申请退款：前端返回要退款的订单信息，后台查询订单是否正在退款或申请超过两次，如果不是则插入到数据库退款，否则申请失败
3. 管理员模块
   1. 进入管理管界面：根据jwt令牌获取的id从数据库中查询用户身份，如果是管理员则放行，否则返回提示权限不足
   2. 查看用户申请店铺记录：查看店铺申请：前端向后台发送请求后，后台从数据库中查找状态是未处理的店铺申请记录并传给前端展示给管理员
   3. 处理店铺申请：管理员根据后台传回的店铺申请，根据用户申请内容决定是否通过请求，将请求的id发到后台后修改数据库中的状态码
   4. 查看商品上架申请，前端向后台发送请求后，后台从数据库中查找状态是未处理的商品上架申请记录并传给前端展示给管理员
   5. 处理商品上架申请：管理员根据后台传回的商品上架申请，根据店铺申请商品内容决定是否通过请求，将请求的id发到后台后修改数据库中的状态码
   6. 删除违规商品：前端传回违规商品的信息，后台再根据商品id删除商品及对应的订单、购物车等的信息，成功后将商品被删除的信息插入到店铺的消息通知中
   7. 删除违规的评论和回复：前端将违规评论或回复的id传回后台，后台再根据id将评论或回复删除，再将评论或回复被删除的信息插入到用户的信息通知中
   8. 查看用户举报商品记录：前端向后台发送请求后，后台从数据库中查找状态是未处理的商品举报记录并传给前端展示给管理员
   9. 处理商品举报申请：管理员根据后台传回的商品上架申请，根据用户描述说明和商品内容决定是否通过请求，将请求的id发到后台后修改数据库中的状态码，根据商品id删除商品，及对应的订单、购物车等的信息，成功后将商品被删除的信息插入到店铺的消息通知中
4. 商品模块
   1. 展示所有的商品：前端将商品分页情况传给后台，后台根据前端传来的分页信息查询商品信息并传回给前端
   2. 根据商品名称和介绍模糊搜索商品：前端将用户输入的信息传输给后台，后台根据输入内容进行模糊搜索并将信息返回给前端
   3. 查看商品的评论：将商品的id传给后台，后台根据商品id查询对应的评论并返回给前端
   4. 查看回复：前端将评论的id返回给后台，后台根据评论的id查询对应的回复，并返回给前端
   5. 发送商品的评论和回复：前端发送商品id或评论的id和内容，后台根据商品id或评论id将评论和回复插入

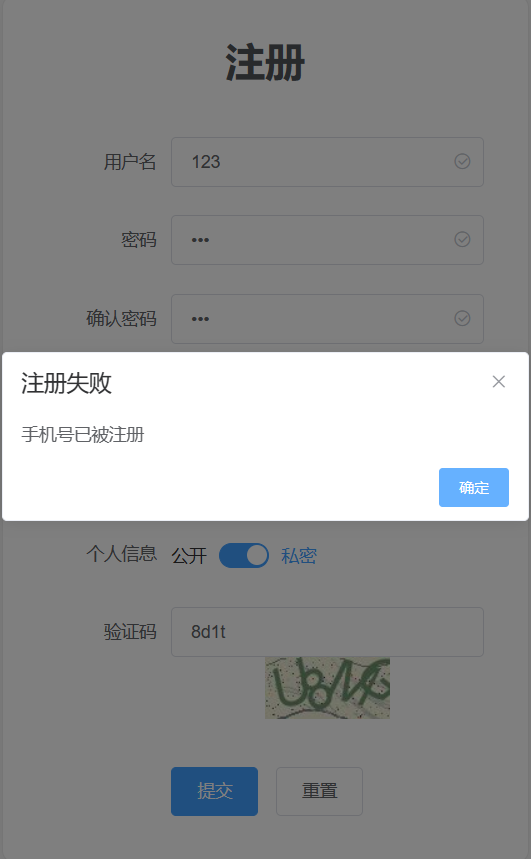
# 程序测试

注册测试：用户注册时需要填写用户名、密码、手机号、地址等信息这些信息都不能为空且手机号为登录标识，必须要唯一且满足正则表达式匹配，必须全为数字且开头为1，项目中用图形验证码模拟手机验证码，需要正确填写验证码才能进行注册，注册时会先判断手机号是否被注册，如果没有被注册则先加上锁进行双重判断，确认没有注册后将用户信息插入数据库中

* 1. 缺少信息



* 1. 输入已注册手机号



3.正确输入信息

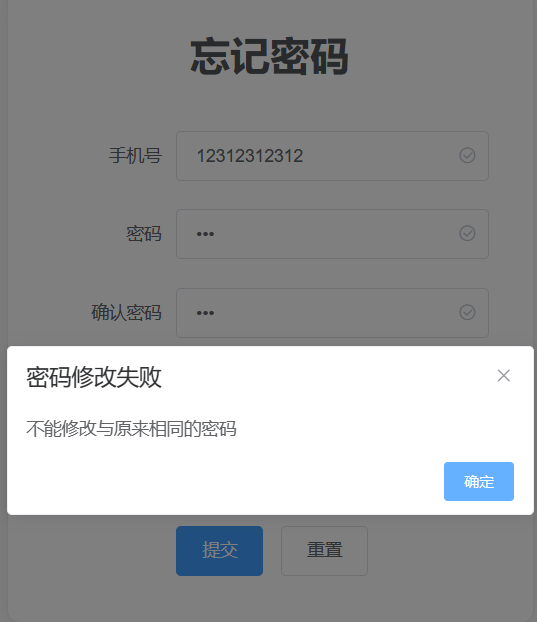


忘记密码测试：用户忘记密码时可以申请修改密码，需要输入手机号匹配账号，再用图形验证码模拟短信验证码，输入正确信息后可以进行修改密码，改密码时会先判断是否与原来密码相同，如果是相同的密码会提示给用户不能修改

1.缺少信息



1. 修改与原来相同的密码



3.正确输入信息

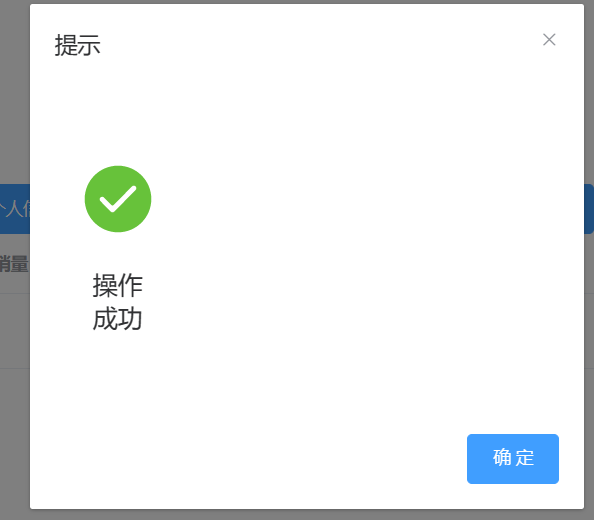


申请店铺测试：用户可以通过提交名字和介绍申请开店铺，如果正在申请时无法继续提交申请，申请经管理员审批同意后可成为商家，成为商家后可以在店铺上传商品，也可以发送推文，用户关注后即可查看到店铺推文，商家可以管理自己发送的推文，还可以管理店铺的订单

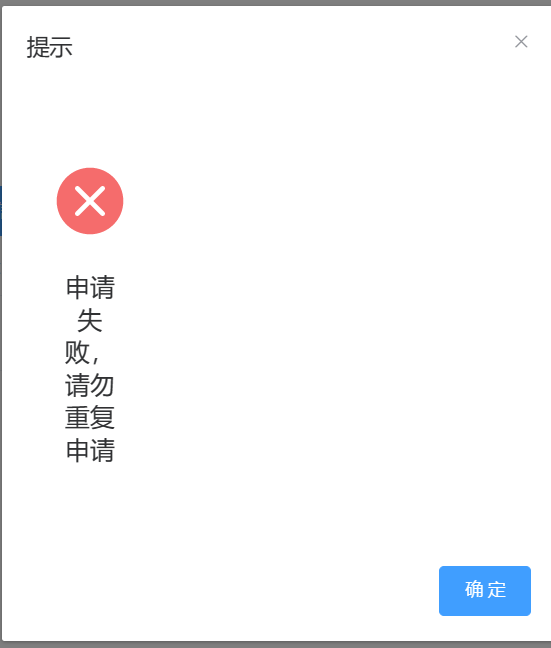
1.第一次申请店铺



结果



重复申请则会报错



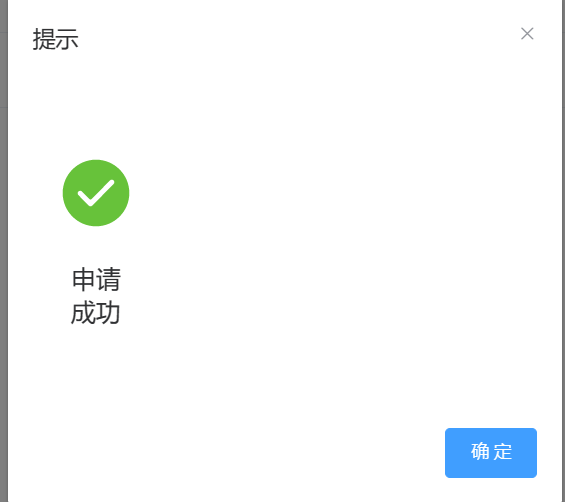
推送商品测试：商家填写商品名称，价格、上架数量后可以进行提交上架申请，商品名称、价格和上架数量必须填写，提交申请后可以查看正在申请的记录，可以上传或修改商品的图片

1.缺少信息



正确填写信息

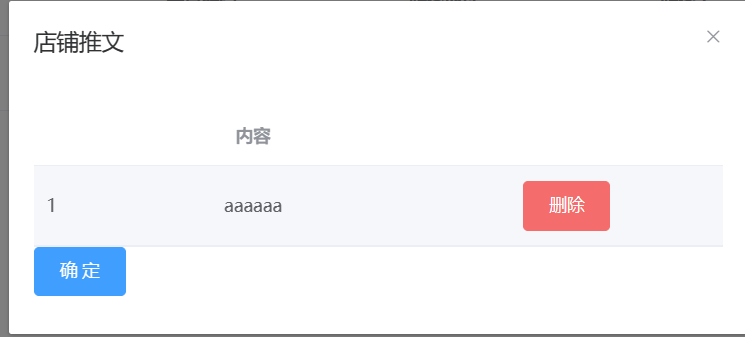




查看上架申请和修改图片：在店铺主页可以查看正在申请的商品的记录，也可以上传或修改图片

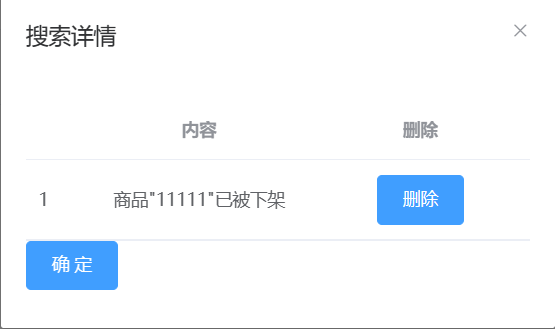


查看和删除推文：商家在店铺主页可以查看店铺发送的推文并且或者将推文删除

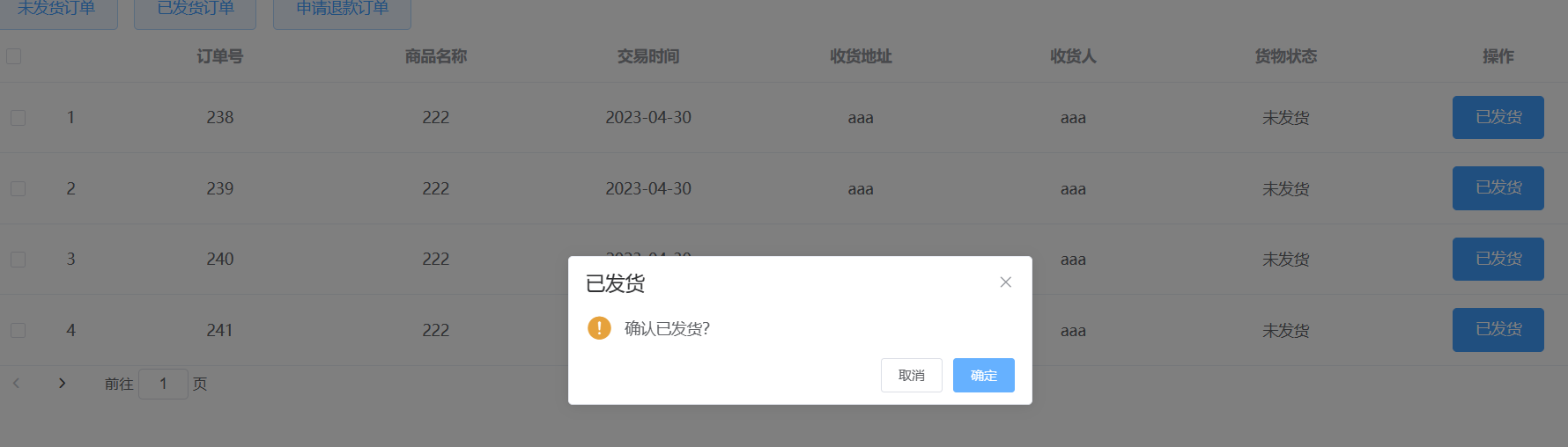




查看和删除店铺信息：在店铺主页可以查看店铺收到的信息，商品被下架后会在消息提醒中查看到被下架的通知，店铺可以将消息提醒删除



确认发货：店家可以在店铺页面查看用户下单信息，下单后可以在系统确认货物已经发送，将订单状态从未发货改为已发货，如果是已经发货的商品再次提交发货申请，就会提示已经发货了





对已发货的商品再次确认已发货



查看退款申请：商家可以在店铺主页进入查看商品退款申请记录，可以根据用户退款申请的原因和描述决定是否同意退款，同意和拒绝退款申请后不可修改

店铺同意退款



拒绝退款



用户申请退款：对于未发货、已发货和已收货的商品，用户可以在订单列表中填写退款原因和说明后申请退款，商家收到退款申请后可以根据用户所写的原因决定是否通过申请，但是用户不可以重复申请退款





对已申请退款的订单再次申请



搜索商品：用户在主界面可以输入关键词搜索商品，后台会根据用户搜索的关键词查找商品名称或介绍或所属店铺名称含有关键词的商品



查找用户：用户在主界面输入关键词，后台会根据用户输入的关键词查找对应的信息公开的用户，用户可以修改个人信息设置为私密防止其他用户查找到自己



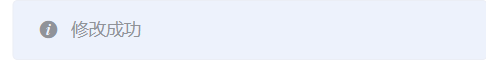
查找店铺：用户在主界面输入关键词后，后台根据用户输入的关键词查找店铺，并将店铺信息返回给前端



关注店铺：用户查找到店铺后可以关注店铺，后台会先判断用户是否关注过店铺，如果关注过店铺则返回提示，如果没有关注过则返回关注成功的提示



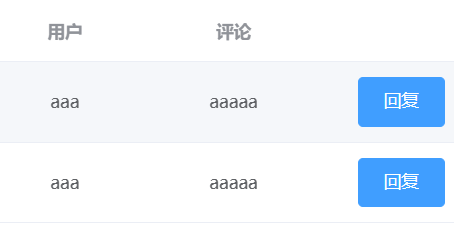
修改个人信息：用户在主界面可以查看自己的个人信息，可以直接进行修改并提交，还可以修改头像



直接购买商品：用户在主界面中可以直接进行购买商品，购买后会生成订单



发送评论：用户可以在商品下查看其他用户的评论，也可以自行发送评论



发送回复：用户可以通过评论下的回复查看其他用户的回复，也可以自己发送回复



举报商品：用户如果发现有违规的商品可以进行举报，举报后管理员可以查看用户的举报记录并决定是否下架商品



重复举报报错：对于正在处理的举报请求，用户只能发送一次



购买购物车商品：用户可以查看自己购物车中的商品，在购物车中，用户可以批量的购买商品



删除购物车商品：除了批量的购买商品外，用户也可以批量的删除购物车中的商品



查看关注店铺：用户可以在主界面查询关注的店铺



取消关注：用户查询到关注的店铺后可以进行取关



查看推文：用户可以查看关注店铺发送的推文



查询和删除用户消息：用户可以在主界面查看收到的消息，用户的评论或回复被删掉后会收到通知



# 项目亮点

1. 手写数据库连接池：在项目中使用xml配置文件，阻塞队列等方式实现了数组库连接池，实现了连接的复用，及多线程访问的安全性，并且实现数据库连接池的增容和扩容，通过读取xml文件的数据配置数据库连接池的参数，提高了连接池的可维护性，通过阻塞队列模拟事务队列实现数据库连接池
2. 减少了线程竞争，使线程在高并发时获取连接进入阻塞状态，防止线程竞争导致的性能下降
3. 优化资源利用：使用阻塞队列可以保证线程池的连接得到充分利用，通过线程池的缩容和扩容避免了因连接池中连接不足或过多导致的资源浪费
4. 提高容错：当数据库连接池连接数达到最大连接数时，如果没有阻塞队列，连接请求会被直接拒绝，这会导致部分请求失败。使用阻塞队列可以将请求放入队列中等待空闲连接，从而提高容错性
5. 降低系统负载：使用阻塞队列可以控制并发访问连接池的数量，从而降低系统负载，提高系统的稳定性和可靠性
6. 手写ORM框架：
   1. 通过手写DataSource、TranSaction、MappedStatament、SqlSession、xml配置文件及注解手写了ORM框架，简化了底层CRUD的开发时间，可复用性强
   2. 复用上述的数据库连接池，提高系统的性能和容错
   3. 使用反射手写ORM框架：通过反射可以在运行时动态地获取类的属性和方法信息，可以大大减少重复代码的编写，同时也可以提高代码的灵活性和可维护性
   4. 通过注解手写ORM框架：使用注解可以在代码中添加额外的数据信息，可以让代码更加简洁和易于理解。在 ORM 框架中，将sql语句和对象的变量名和数据库表中变量名的映射关系的id存放在注解中，可以更加方便的进行数据库的操作，也提高了代码的可读性
   5. 动态代理：使用动态代理在运行时动态地生成代理类，方便地实现一些通用的操作。在 ORM 框架中，通过动态代理生成dao层接口的代理对象，减少书写重复代码，通过该getMapper接口配合注解和反射，实现mapper代理开发，实现对数据库操作的统一管理，从而可以大大减少重复代码的编写。
   6. 使用LocalThread、手写连接池和事务管理器管理Connection对象，便于管理事务及复用Connection
7. 手写IOC框架：
   1. 通过注解方式实现了IOC框架，实现控制反转和依赖注入，进一步降低程序的耦合度，降低了程序的维护成本，可以根据业务需求拓展内容，更加灵活的满足业务需求，提高系统性能
   2. 通过二级缓存实现IOC框架，解决循环依赖问题
   3. 通过注释将一些类设置成单例，在需要使用单例对象时直接从单例池中拿到，减少对象资源存储的消耗
   4. 通过BeanFactory工厂类生成对应的Mapper接口的代理对象，自动实例化并装配Bean对象，避免重复创建对象的繁琐工作，提高了开发效率，beanFactory可以自动注入对象间的依赖关系，避免手动控制对象之间的依赖关系，降低耦合度，易于维护
8. 在并发问题中使用双重检查锁，解决程序并发性问题，并且减少了锁的粒度，降低了锁的开销，提高效率
9. 数据库敏感数据的加密存储：使用了MD5和加盐方式对数据库敏感数据进行加密，避免敏感数据在数据库的明文存储，提高业务的安全性
10. 使用BaseServlet转发器，将一些通用的代码封装到一个基类中，让子类继承基类从而实现基类代码的复用，减少重复代码的编写，还可以简化代码的结构，通过重写service方法根据url获取方法名称并通过反射找到对应的方法，可以将一些通用的逻辑代码从servlet中剥离出来，使代码更易于维护，并且提高了代码的可读性，后续还可以在基类中实现一些安全性相关的逻辑，如身份验证，可以提高代码的安全性
11. 使用jwt令牌存储对象的信息，提供跨域支持，只有具有正确的密钥才可以创建和解析令牌，可以确保数据传输的安全性，还可以通过查看有无jwt令牌判断是否登录，还可以在jwt令牌中添加自定义信息，如用户id，方便满足用户身份校验、授权等安全和验证需求，jwt令牌无状态，服务器不需要在每个请求维护状态信息
12. 使用Filter配合jwt令牌判断用户身份实现权限限制，通过将jwt令牌添加到cookie和请求头中，可以在一些需要判断用户权限的地方拦截非法请求，且可以通过改变jwt令牌中所含自定义信息扩展业务

# 心得体会

通过完成这次购物网的项目，让我了解到了新的技术，如jwt令牌鉴权模型还有数据库敏感数据加密存储等方法，丰富了我的技术栈，此外也丰富了我的开发经验，让我对业务的流程更加熟悉还有就是增强了我的解决问题的能力。在这次项目中遇到了很多的问题，例如：

1. 在手写IOC框架时，dao层的注入问题困扰了我很久，dao层只有一个接口，不能直接new出来，如何能解决在IOC容器中获得代理对象以及应该由orm还是ioc框架生成mapper的代理对象成了难题
2. Mapper自动注入后不需要SqlSession生成，类中也不需要了SqlSession对象，又该如何进行事务的提交和回滚
3. 图片的存储是否应该放在数据库中

这些问题最后通过学习在网上查询资料和与同学的讨论，都找到了解决方法，

1. 在生成dao层对象时，可以使用beanFactory工厂对象生成，在工厂对象中使用ORM框架提供的getMapper接口，将所需要的interface接口交给beanFactory对象，ORM框架将interface接口对象动态生成对应的代理类，然后为需要依赖注入的dao层对象加上@mapper注解，并在扫描时将ORM框架生成的代理类赋值给需要注入的dao层对象，从而解决mapper对象依赖注入的问题
2. 封装一个SqlSessionUtil对象，就可以直接在工具类中将SqlSession对象交给IOC框架创建mapper代理对象，同时还可以实现事务的回滚与提交
3. 图片可以存在云服务器中，在数据库中存下图片的url地址，然后需要用的时候从网上查找

在解决这些问题的过程中，培养了我分析问题和解决问题的能力，要从造成问题的原因出发，再找到方法解决问题。还有在项目的开发的反思过程中，让我更加好的考虑到各个方面的细节问题，例如安全性和可扩展性和可维护性，这样可以更方便的在后续维护项目。

在完成项目的过程中，由于一开始的数据库并没有设计好，导致后续频繁的更改，让我意识到在完成项目的过程中，首先要完成优化数据库的设计，数据库表的设计非常重要，根据业务需求设计出合适的数据库表结构，可以减少数据冗余和复杂性，提高数据库的查询效率。还有在开发的过程中，要进行单元测试和系统测试，确保系统的安全性和可靠性，这样才能开发出高质量的系统网站。