《软件开发综合实践》实验报告

信息学院 学院 计算机科学与技术 专业 2021 级

实验时间 2023 年 8 月 15 日

姓名 李东骏 学号 20211120141

实验名称 购物管理系统

实验成绩

一、实验目的

培养软件项目管理的意识，使学生掌握规范的软件开发和维护的方法。

二、实验软件

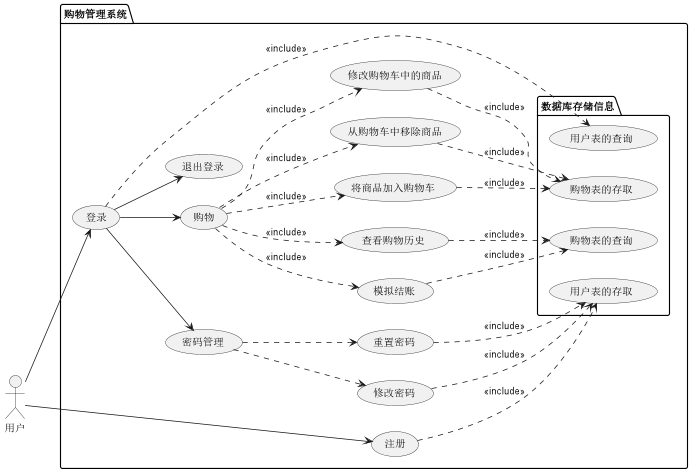
Idea，Springboot，Mybatis，Maven。

1. 实验方案

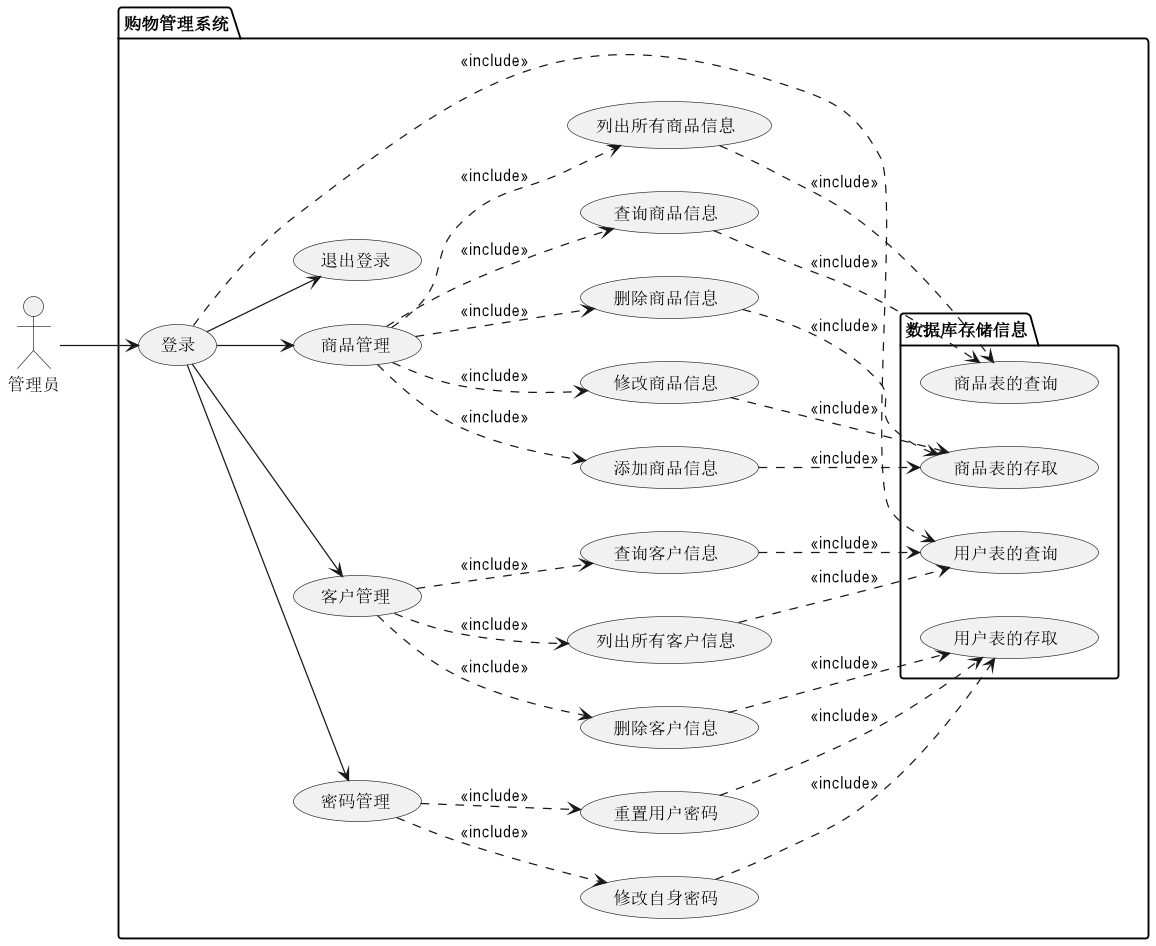
因为本次实验要求的是写一个购物管理系统，分别有管理员和客户两种角色，并且商品以及个人信息需要进行存储以及一系列操作，所以应用到了mysql数据库来进行数据的存储，其次就是对数据进行操作，我学习了mybatis，并且使用mybatis来进行对数据库的操作，我使用了maven来管理依赖项和构建过程，并且使用了springboot来简化程序的开发。

在进行程序的开发之前，我首先根据系统的要求画出了UML用例图和顺序图：

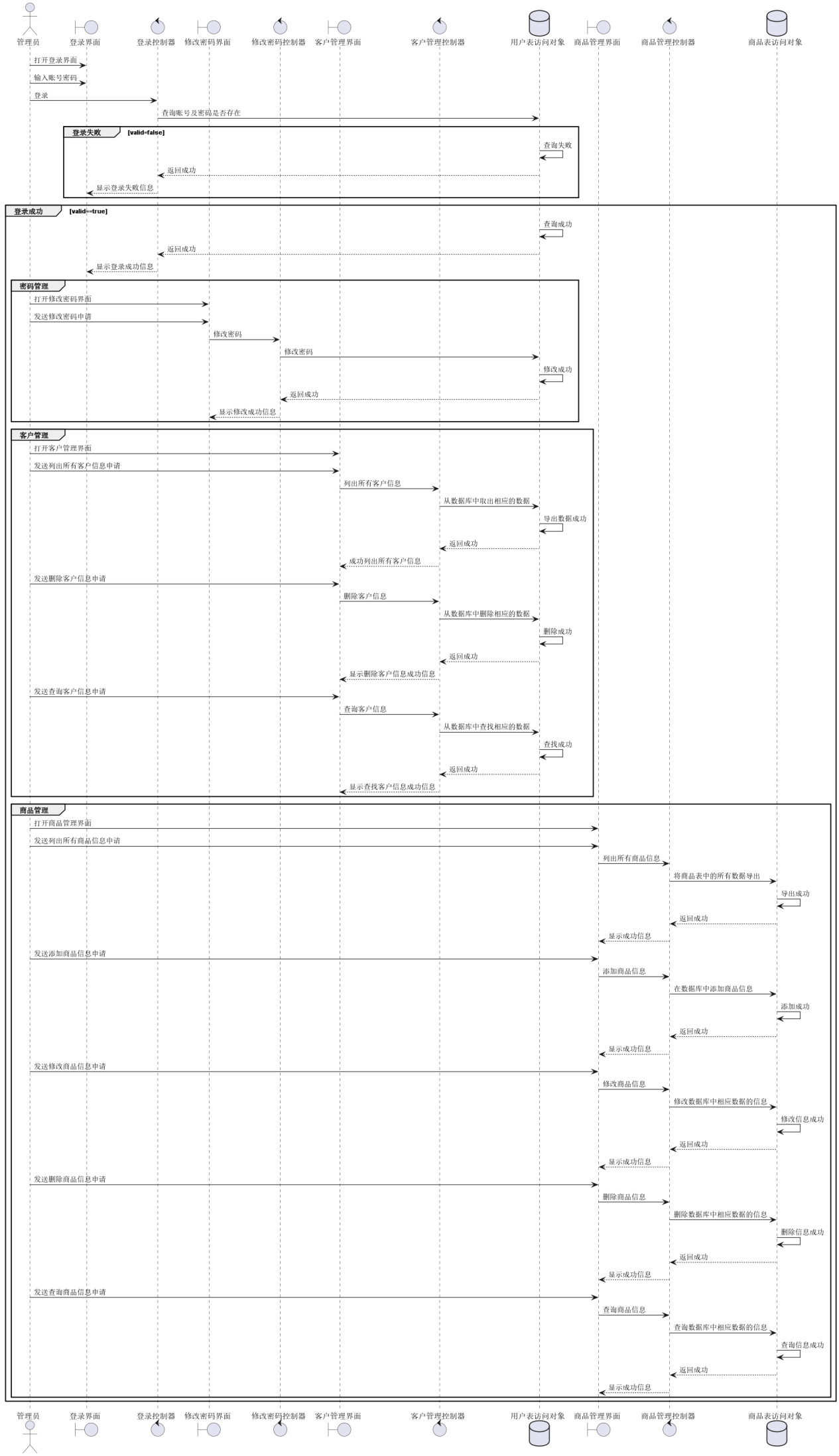
客户的用例图：



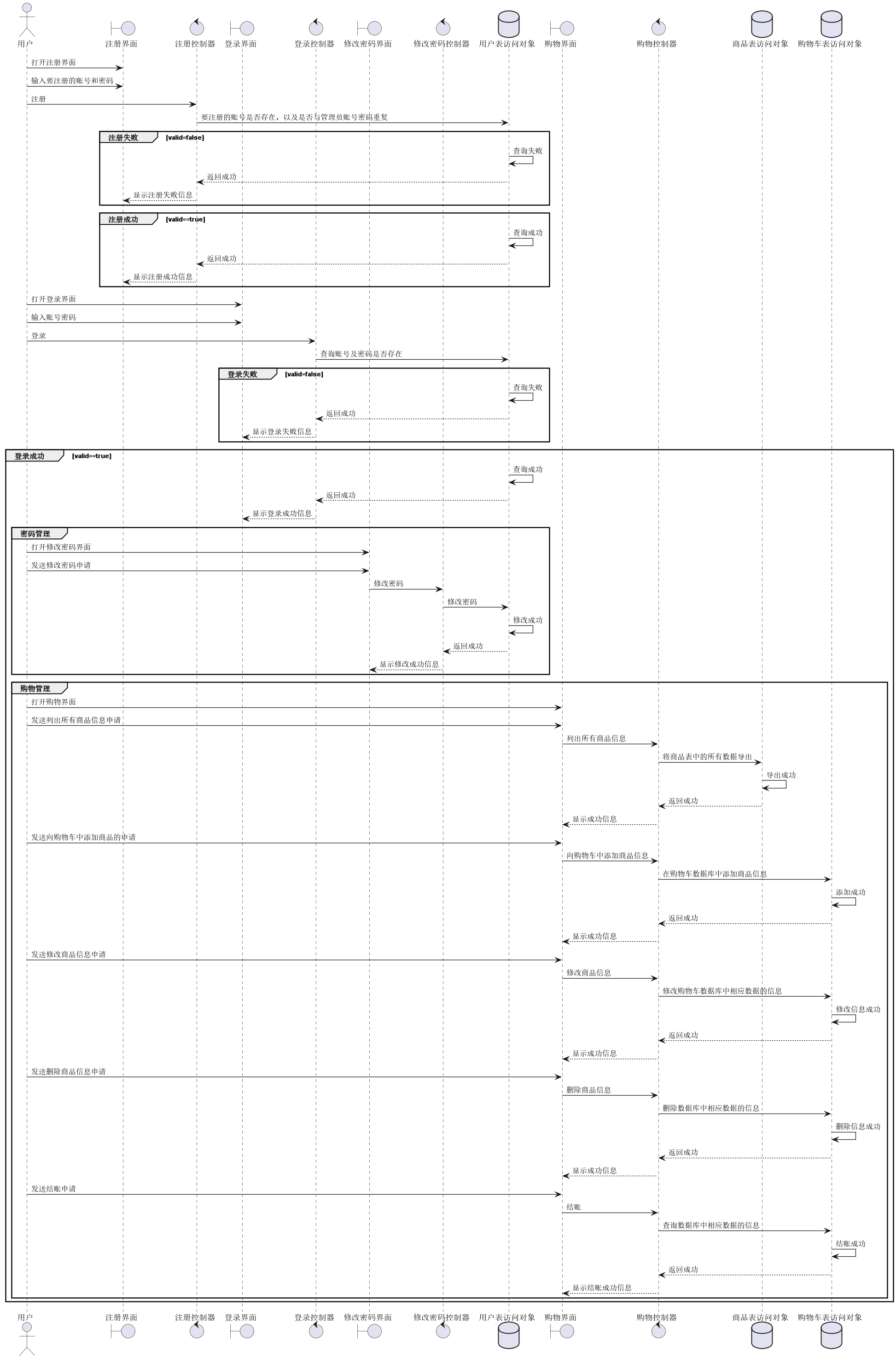
管理员的用例图：



管理员的顺序图：



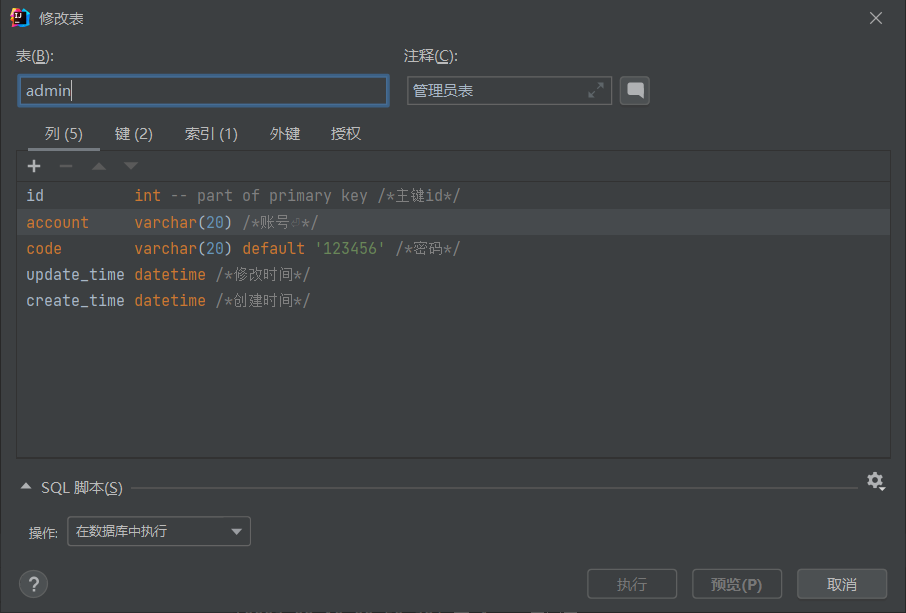
客户的顺序图：



1. 实验步骤

在画出上面的用例图以及顺序图之后，我首先对数据里面需要创建的表进行了涉及，因为有客户登录，管理员登录，商品，购物车，四个表进行了设计，下面是通过可视化工具idea对表进行创建：

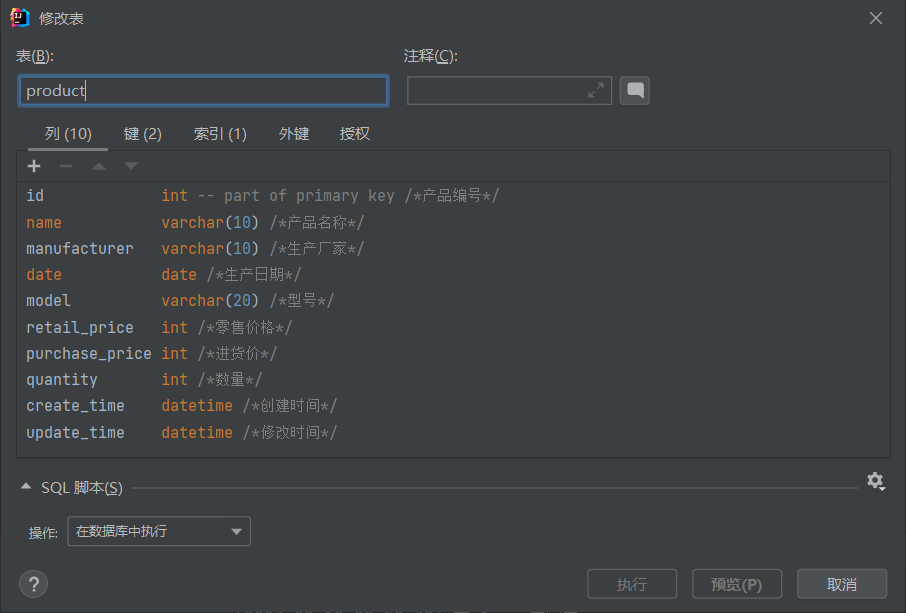
管理员表：



用户表：



商品表：



购物车表：



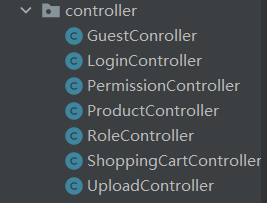
在编写程序的过程中，我采用了前后端分离的程序开发模式，在后端中我主要通过三个层来完成用户的每一项操作，分别为控制层Controller和业务逻辑层Service以及数据访问层Dao。

因为需要在java程序中要对数据库进行操作，所以要根据数据库表的创建创建出一个实体类，然后一个系统首先就是需要登录，所以登录之前就不能对其他的功能进行操作，在这里我采用了Intercepter拦截器进行身份验证，用到拦截器的时候需要一个配置类来配置和管理拦截器的行为和生命周期，这里同样会用到jwt令牌，所以需要创建出一个工具类用来进行JWT令牌的生成。在系统操作中难免会有异常出现，我创建了一个异常类来对全局的异常做一个统一的处理。

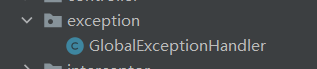
拦截器的配置类：



控制器类：



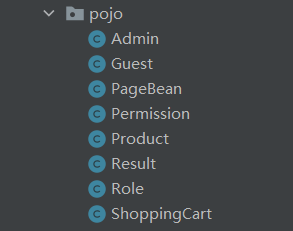
异常处理类：



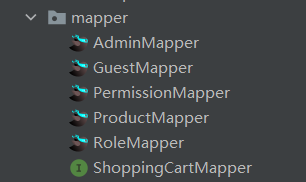
拦截器类：

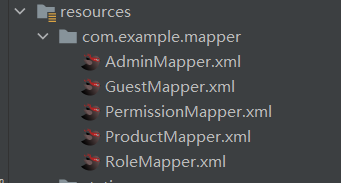


实体类：

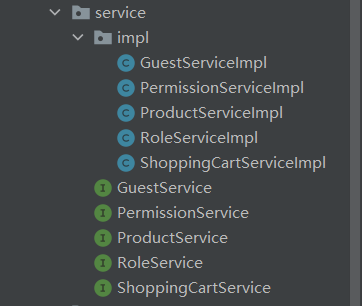


Mapper类（Dao层，因为用到了mybatis，所以这里用mapper类）：





业务层：



整体的类图如下所示：

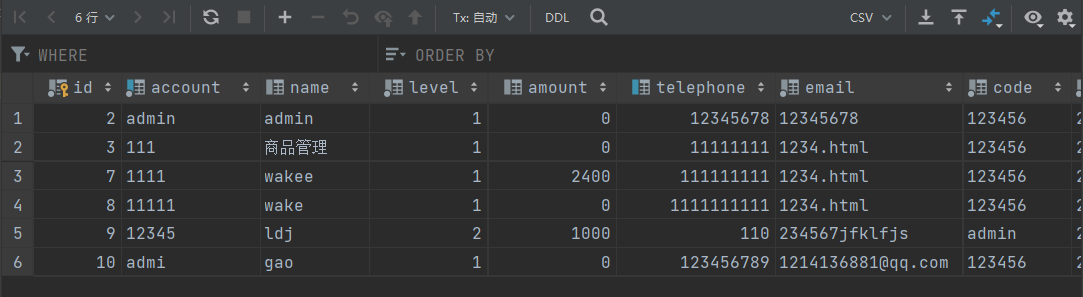


五、实验结果及分析

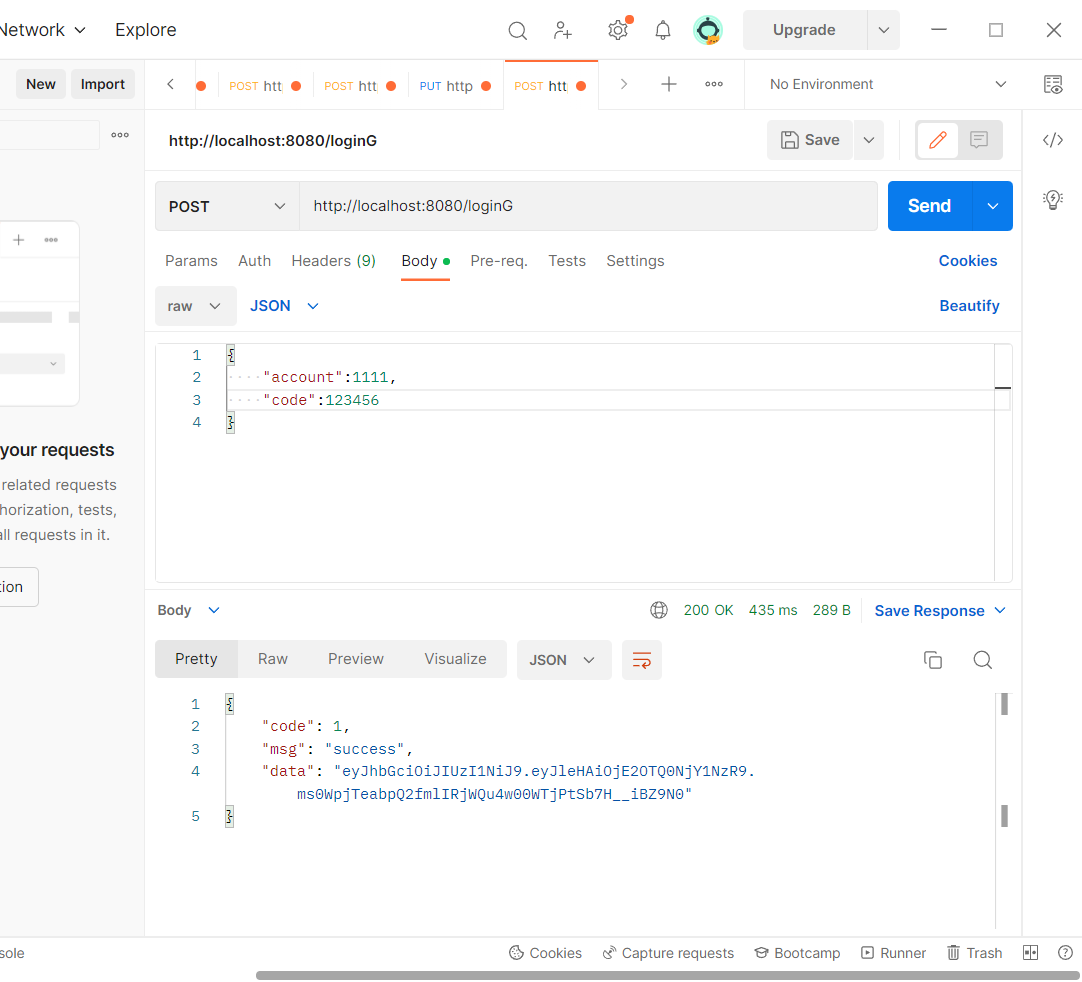
这里我使用了postman来对所有的功能进行测试：

1. 管理员与用户的登录功能：

在进行测试之前首先往数据库中添加了几条数据：



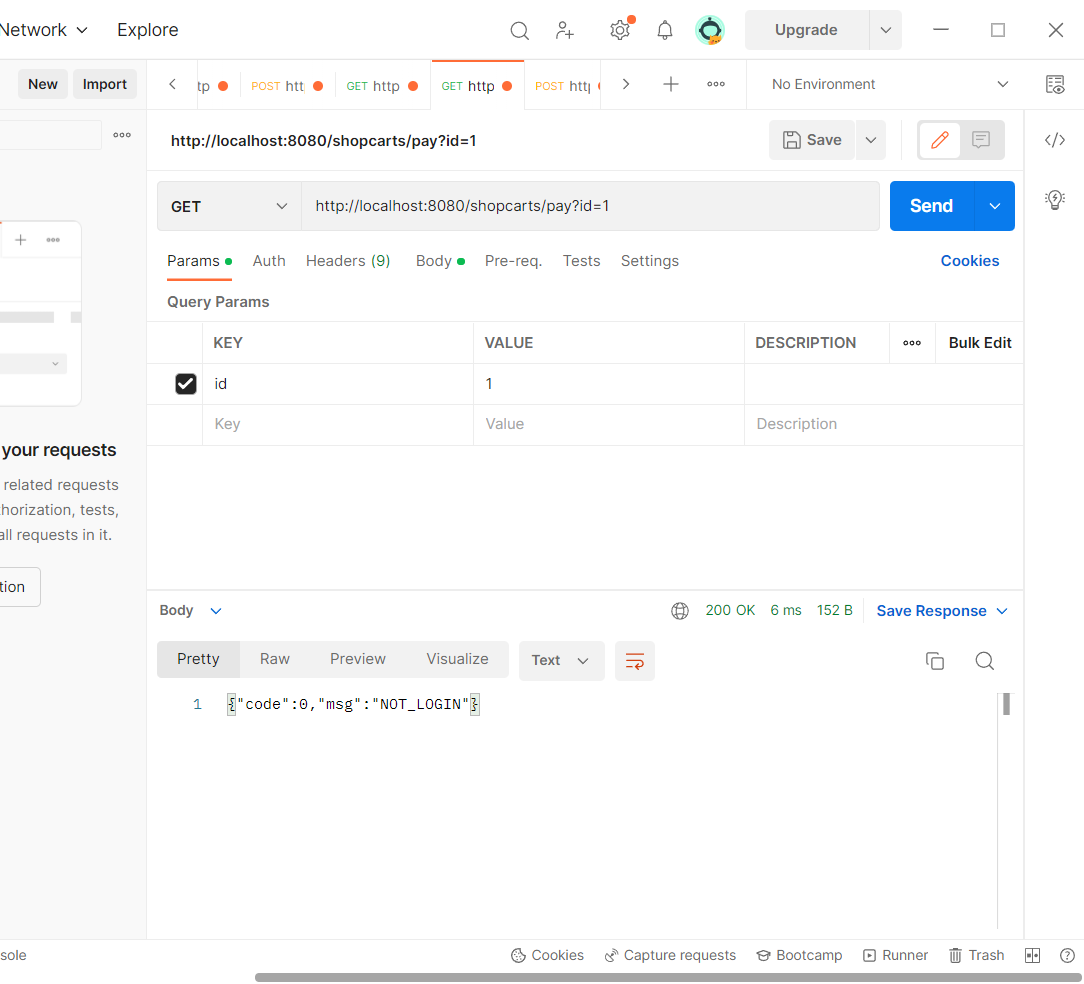
然后通过postman进行测试：



可以看到成功登录并且返回了令牌，并且在日志中显示出了有客户正在登录：



在这里我为了后续的测试关闭了拦截器，否则将无法进行其他功能的测试：



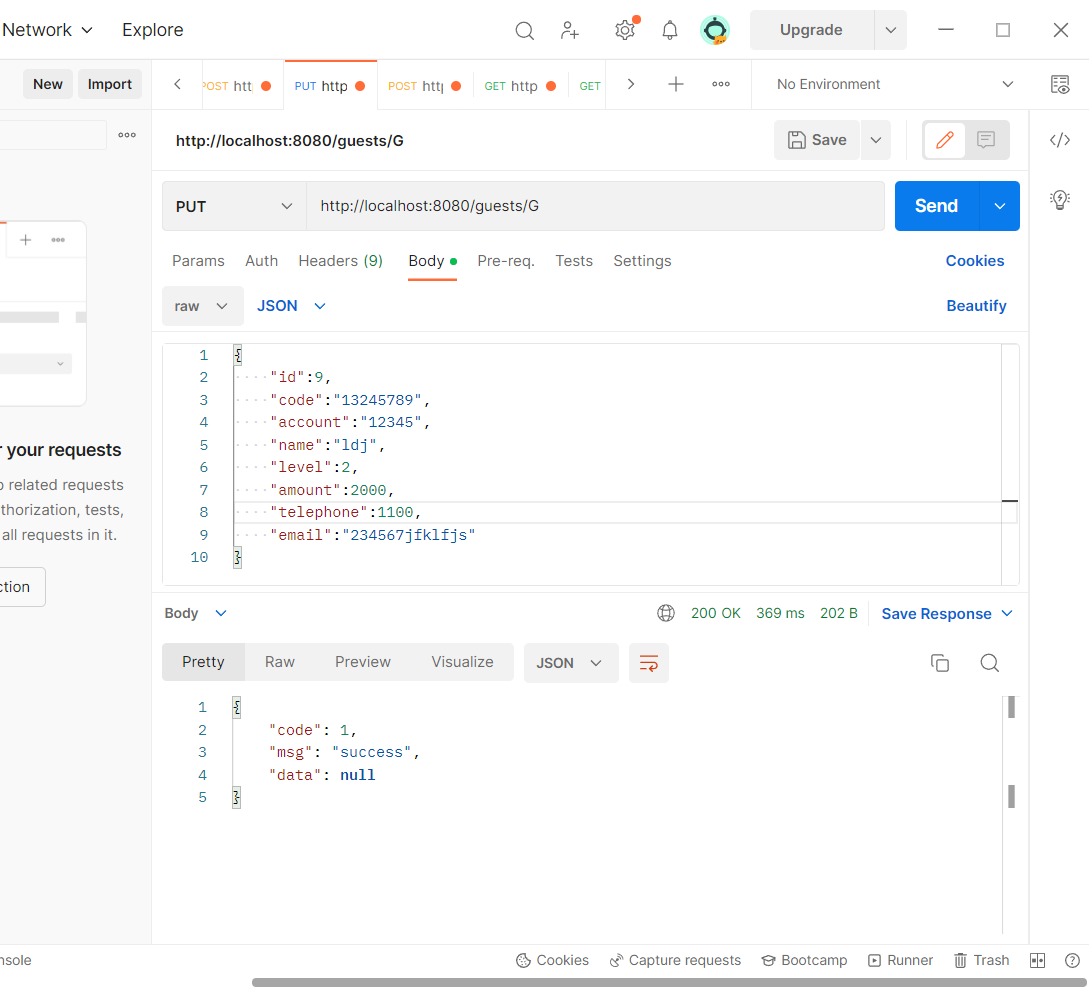
在这里我调用了付款的功能但是显示没有登录，所以不会造成没有登录便可以对其他功能进行操作的错误。

1. 管理员进行客户管理：

首先就是客户信息的修改，首先是修改之前的用户数据：



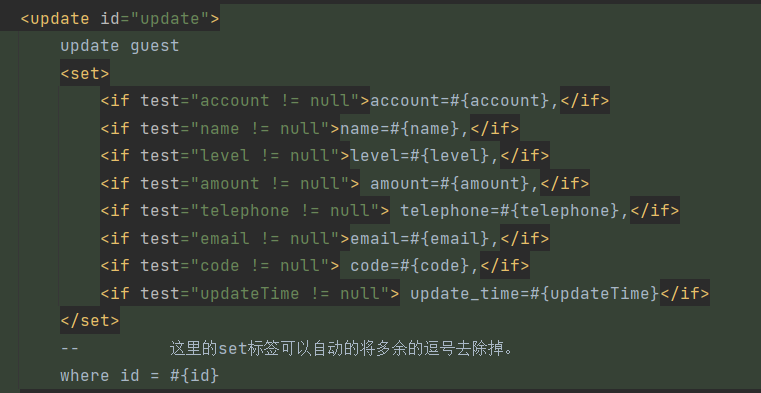
运用postman进行操作：



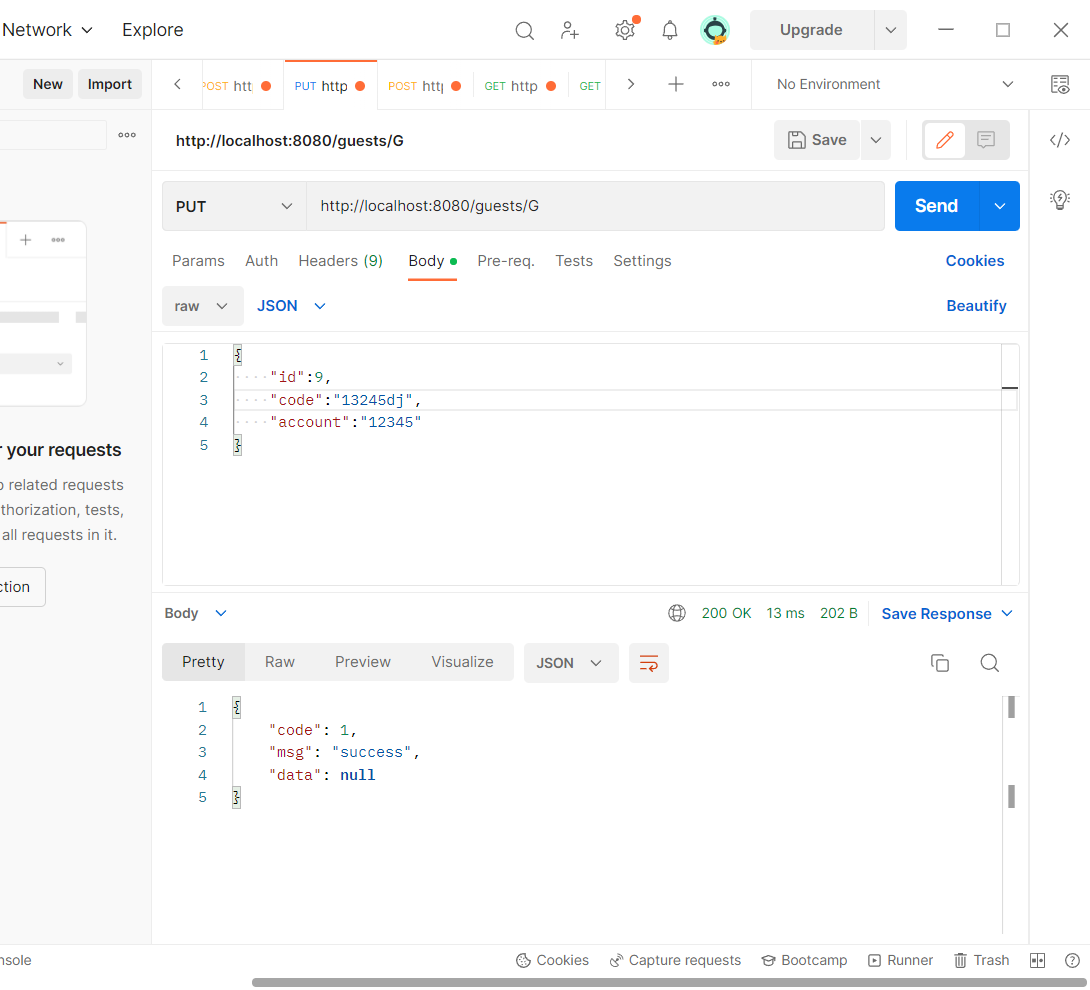
数据库中信息的变化：



可以看到用户的数据成功的修改，并且在updatetime可以看到时间发生了变化，而且因为我在mybatis中使用了if来判断数据是否为空：



所以这个功能还可以专门用来修改用户的密码：



数据库的变化：

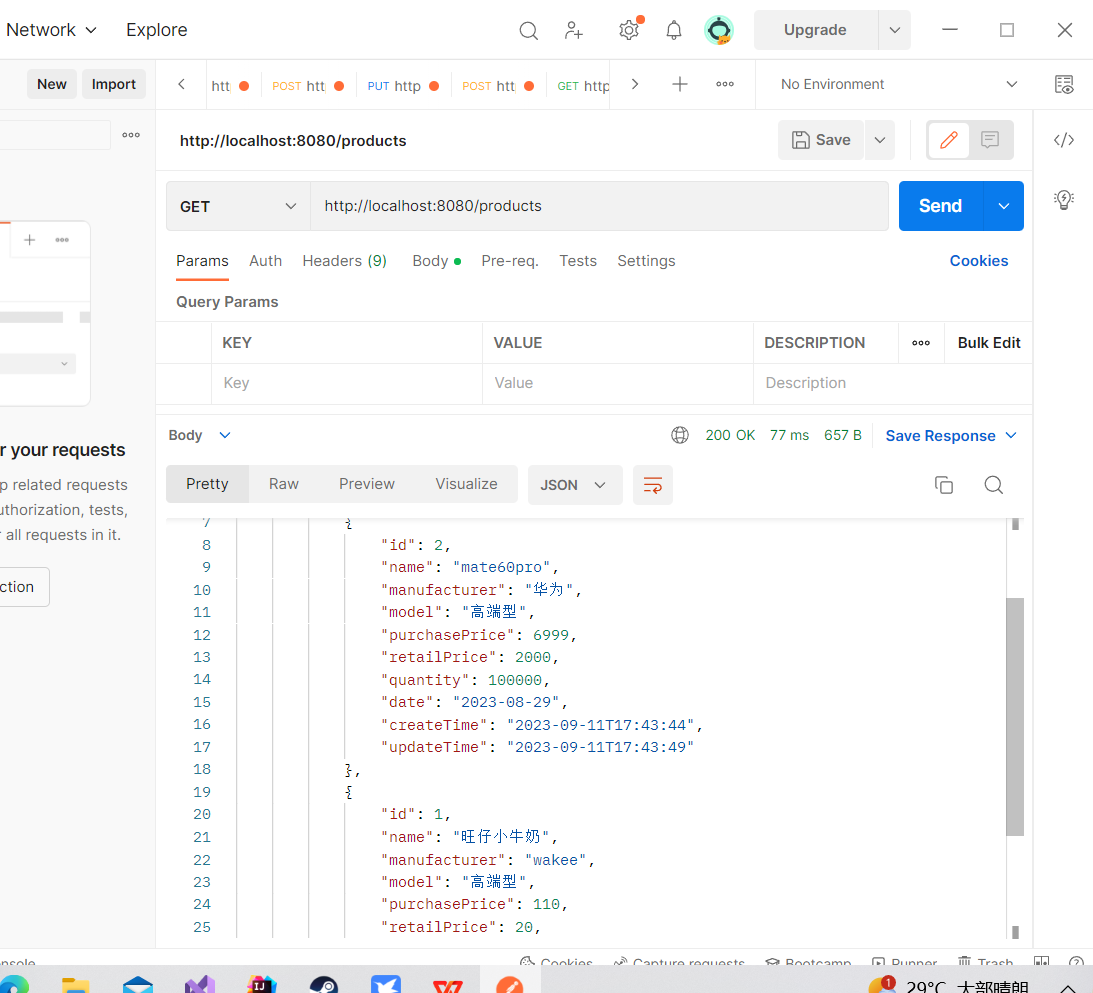


我们可以看到除了密码被改变，其他的数据都没有发生变化。

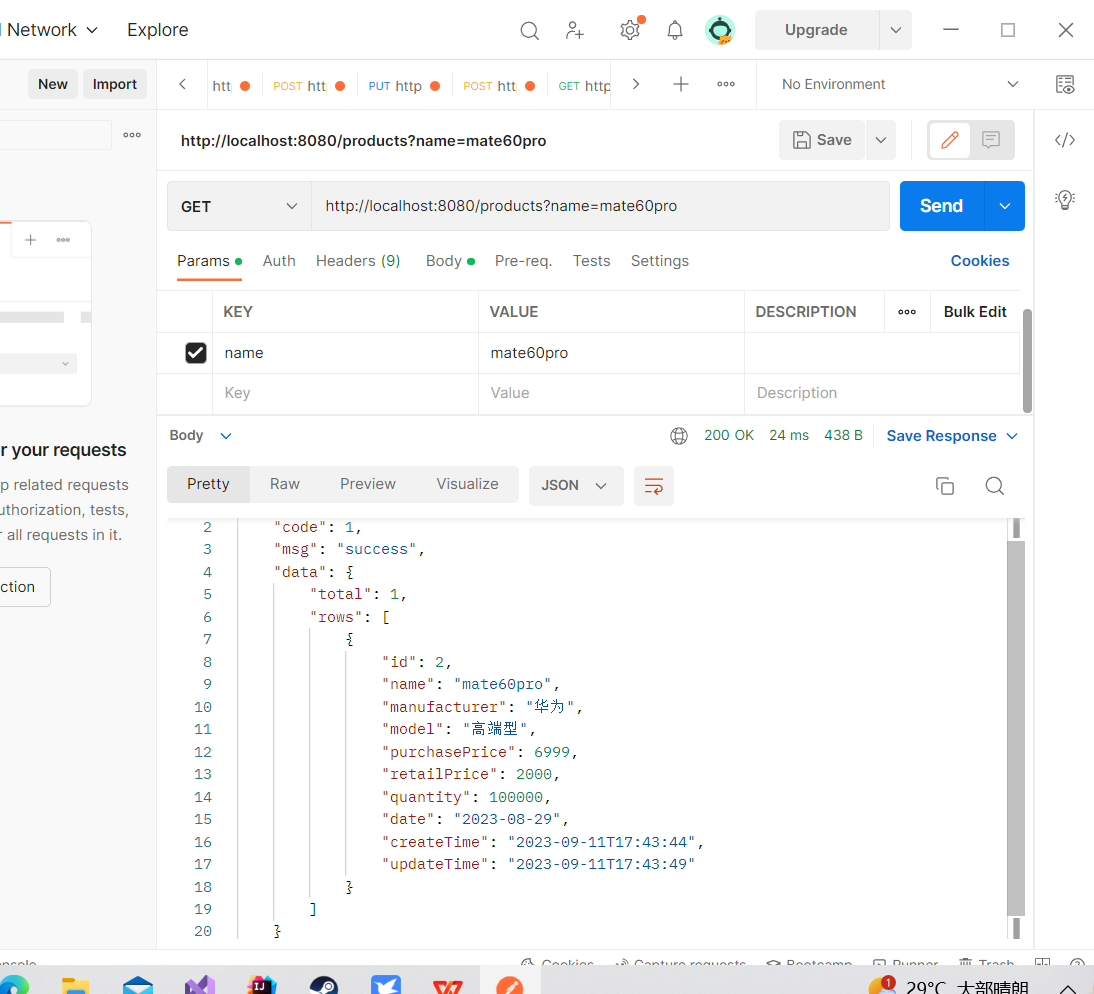
1. 管理员进行商品管理：

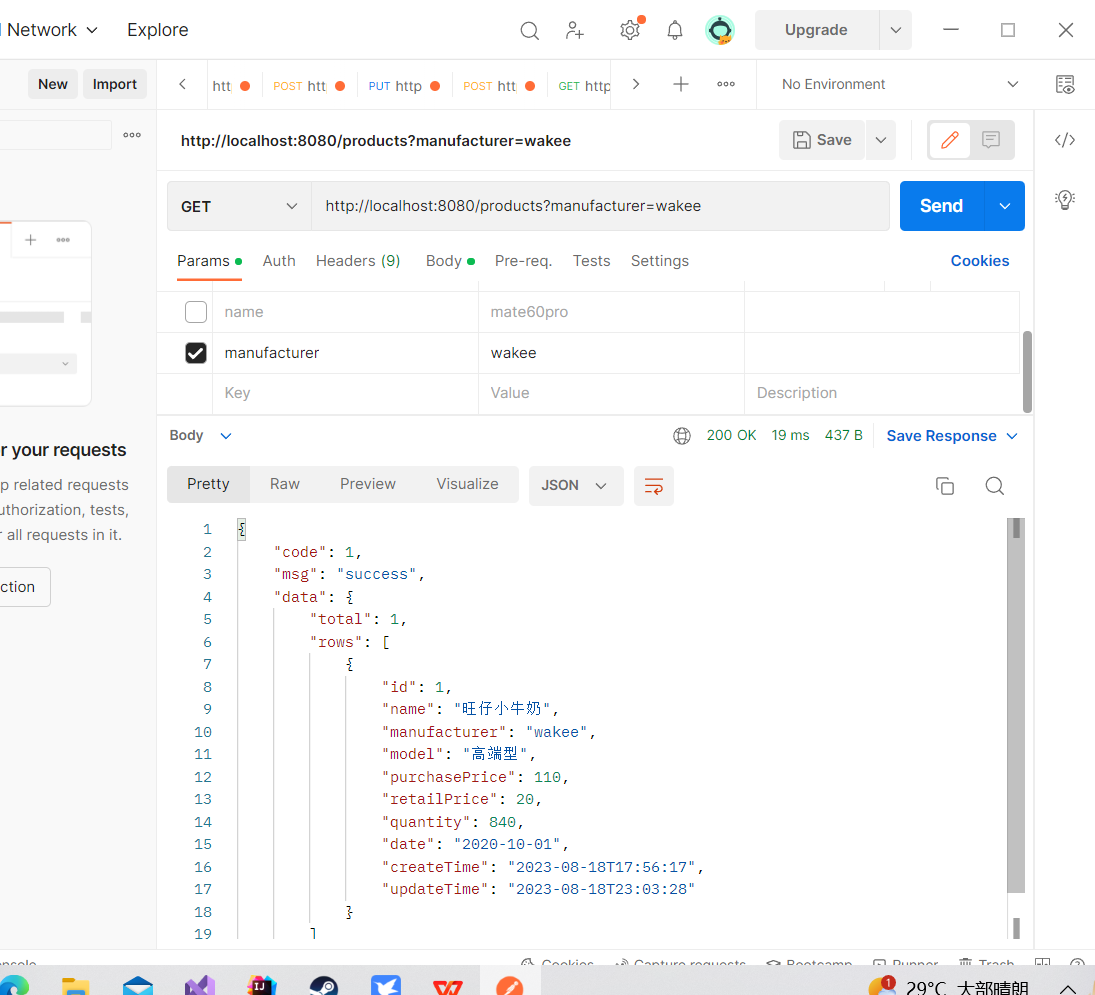
因为在上面我主要介绍了修改信息的功能，在这里我主要讲解对商品的分页展示，查找以及批量删除功能：

首先是查找功能，当我不传任何参数的时候就是可以展示所有的商品信息：

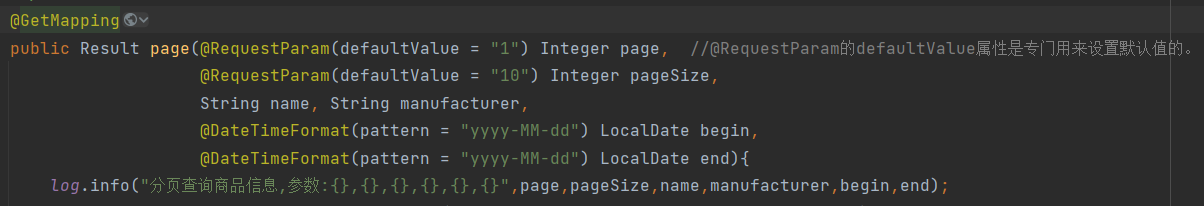


然后可以根据单个或多个条件进行查找：



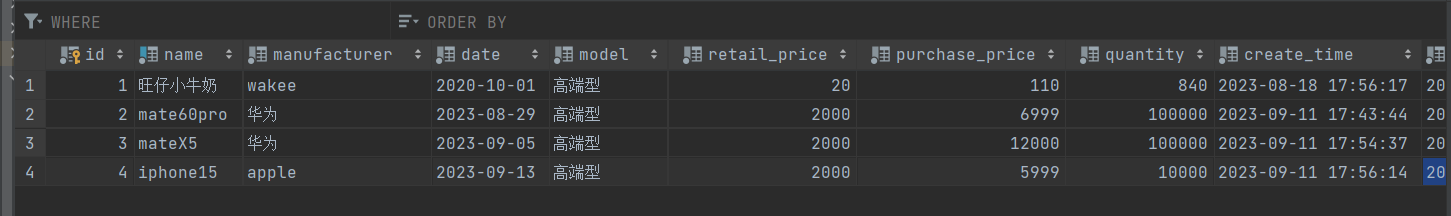


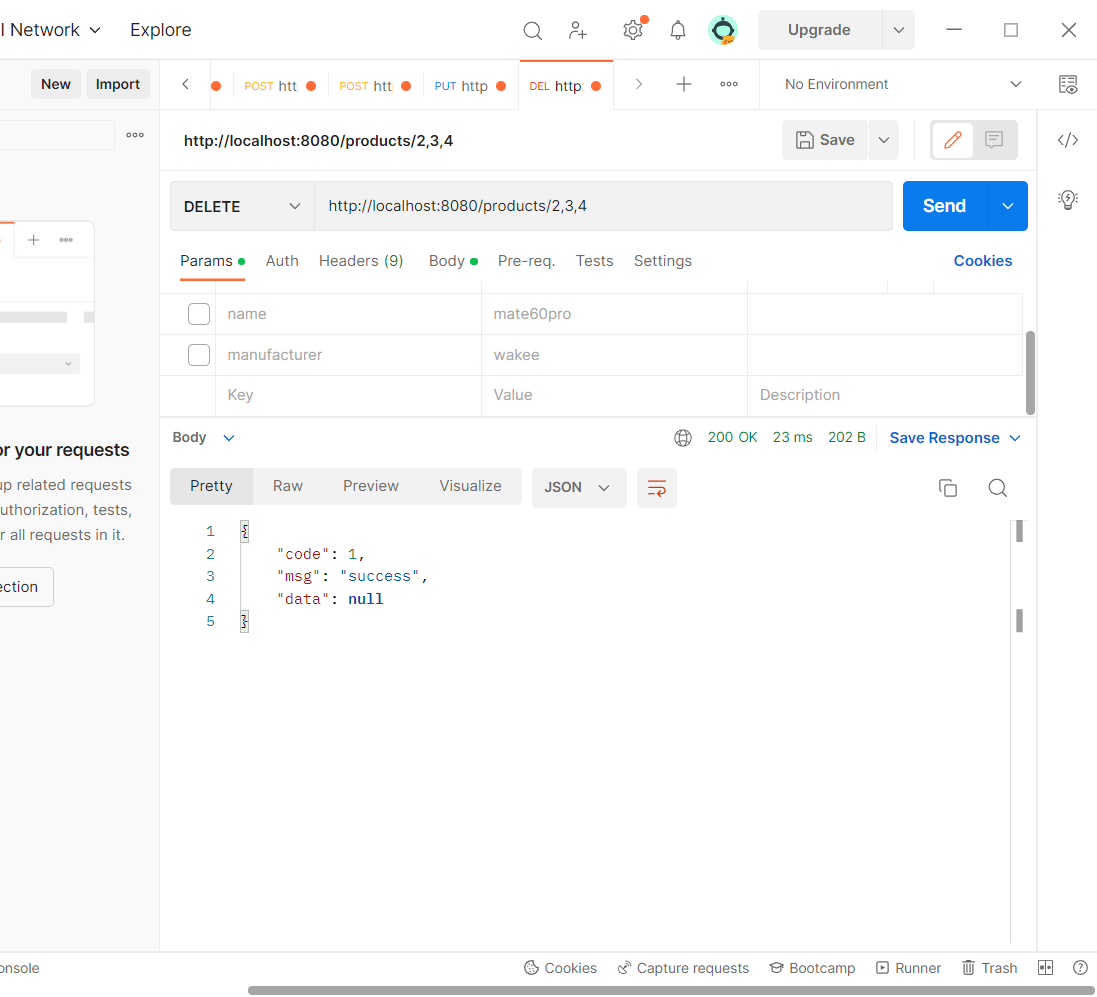
因为数据量过少无法展示分页功能，但是我设置了分页展示功能：



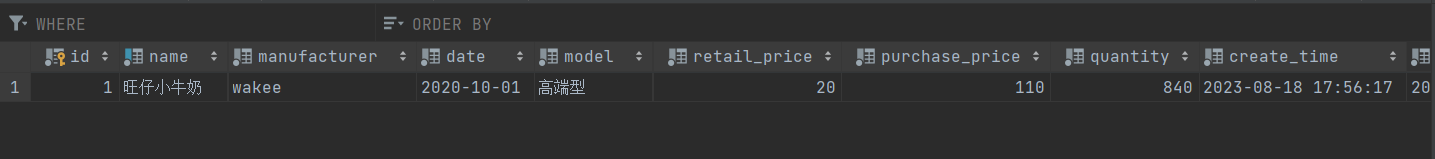
然后就是批量删除的功能：

修改之前的数据库数据：



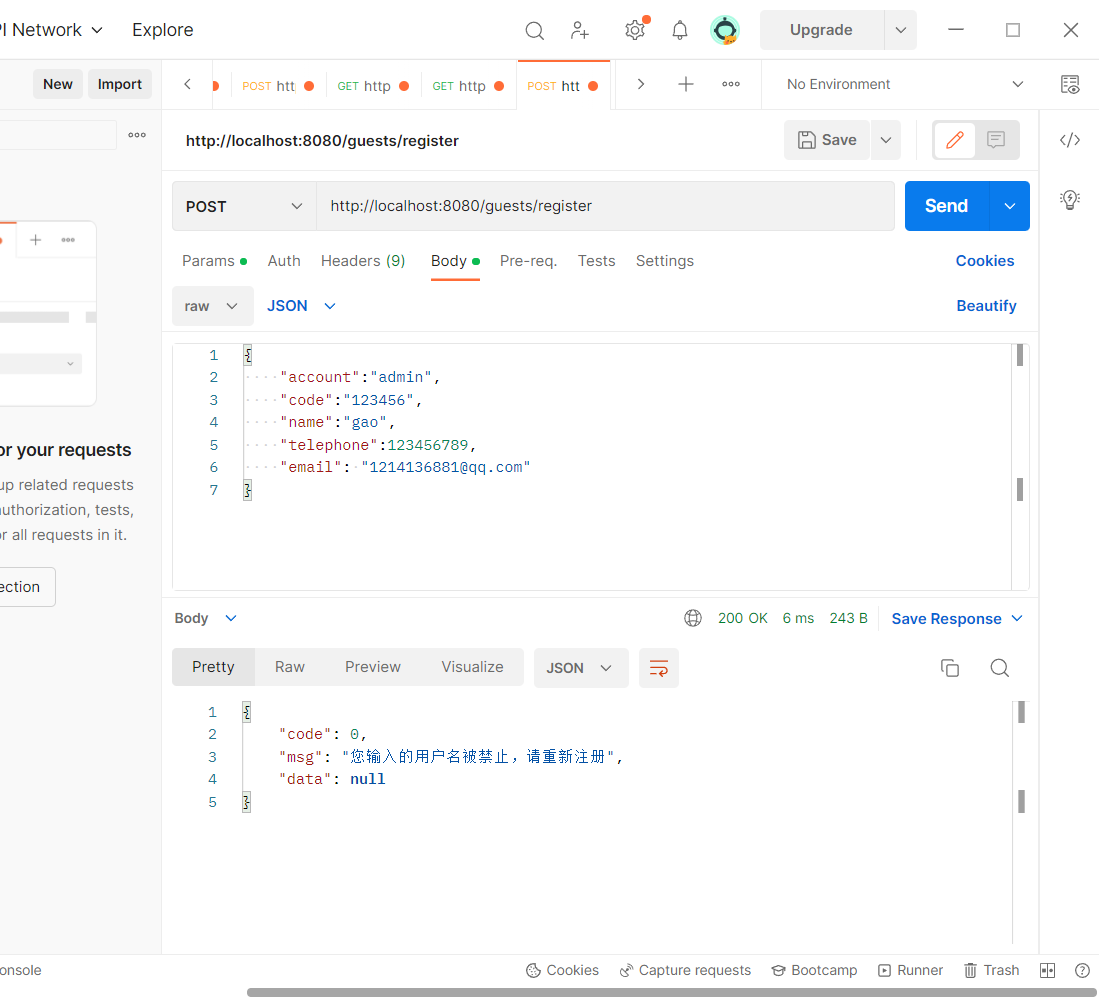


删除后的数据库数据：



1. 用户进行注册功能：

注册功能没有什么独特的地方，就是注册的时候不能与管理员的账号发生重复：



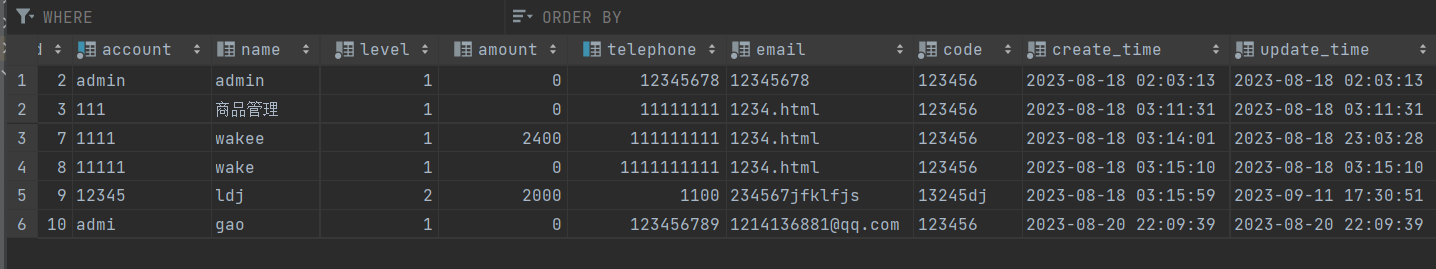
如上若是用户自定义的账号等于admin的话会使得注册过程失败。

1. 用户购物功能：

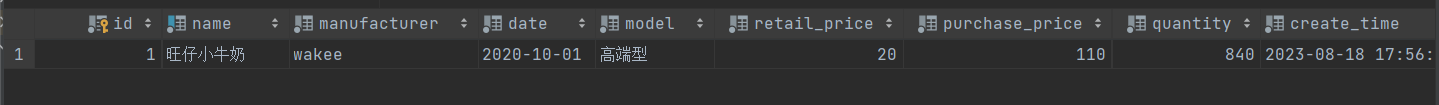
这部分我主要介绍一下结账功能：结账功能会涉及到三个表的操作。

在操作之前的数据库信息：

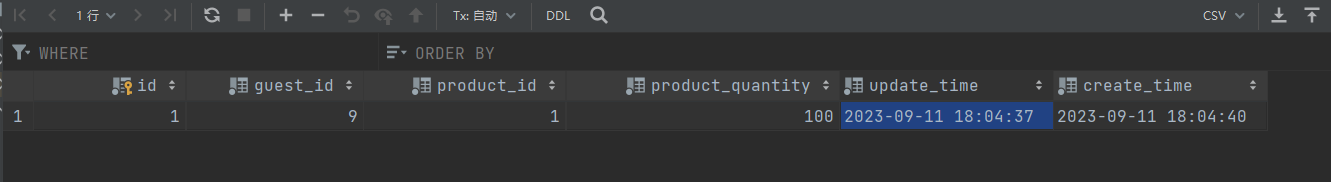
客户表：



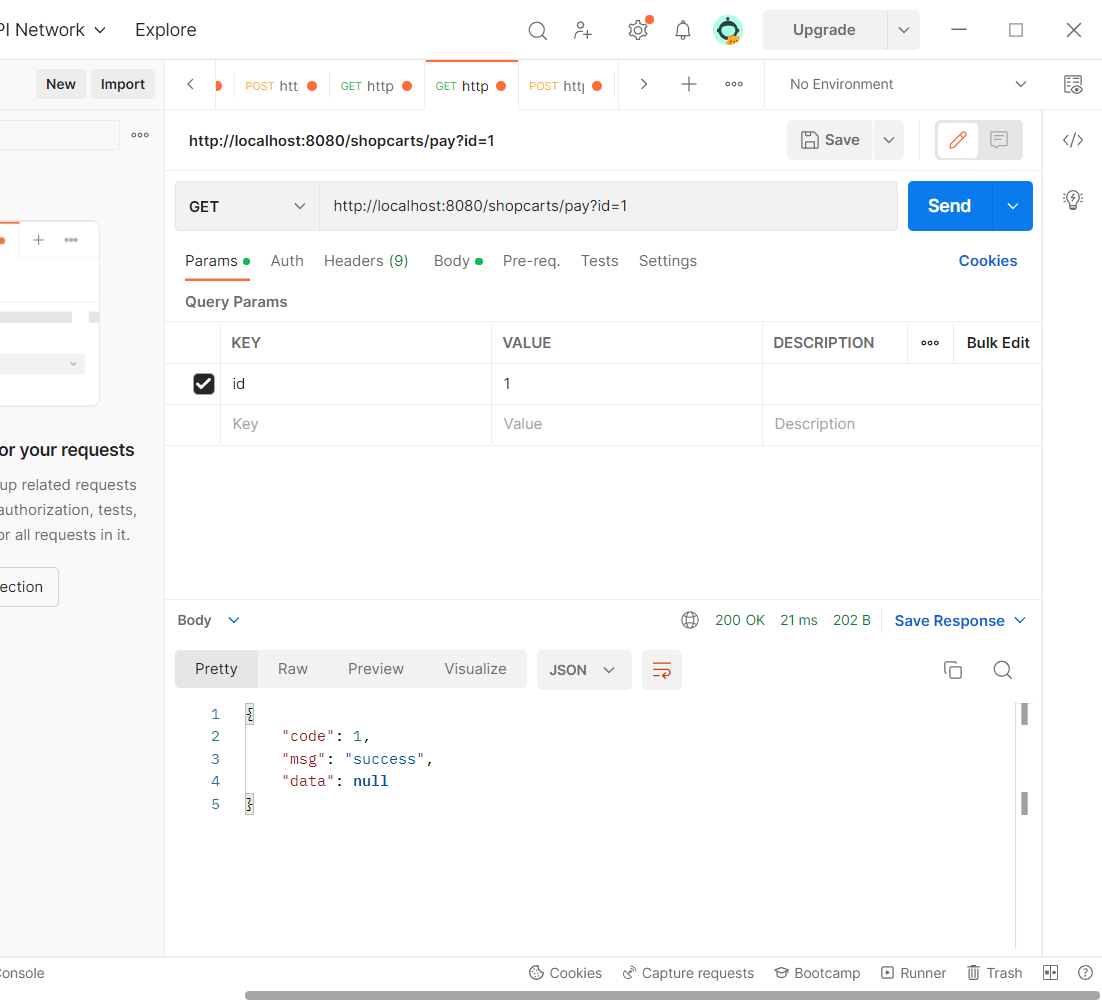
商品表：



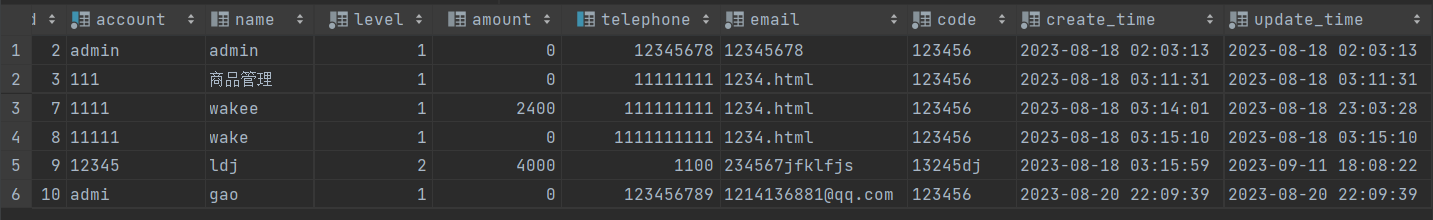
购物车表：

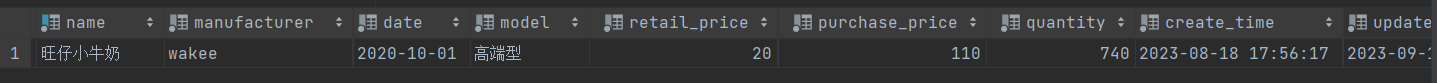


通过postman进行付款操作：



操作之后数据库中数据的变化：

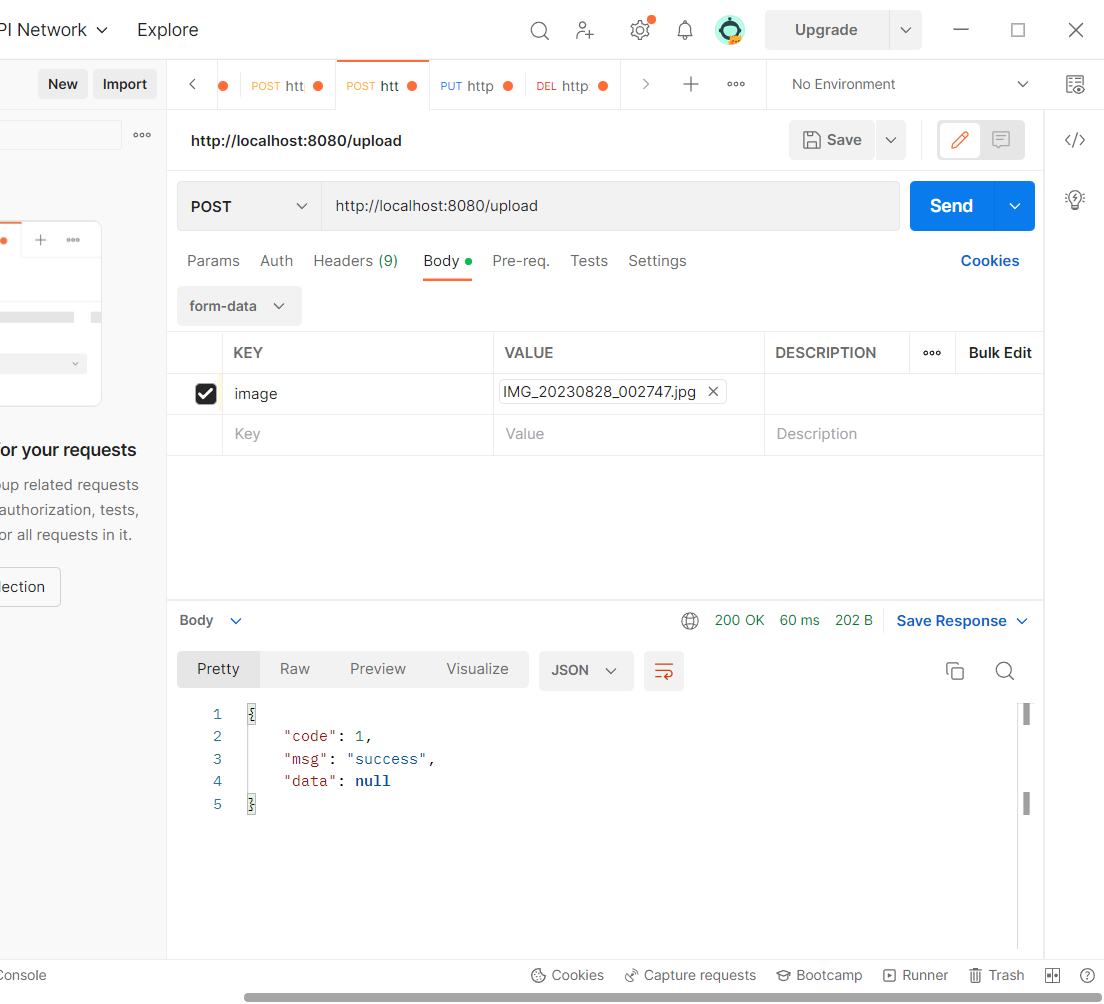


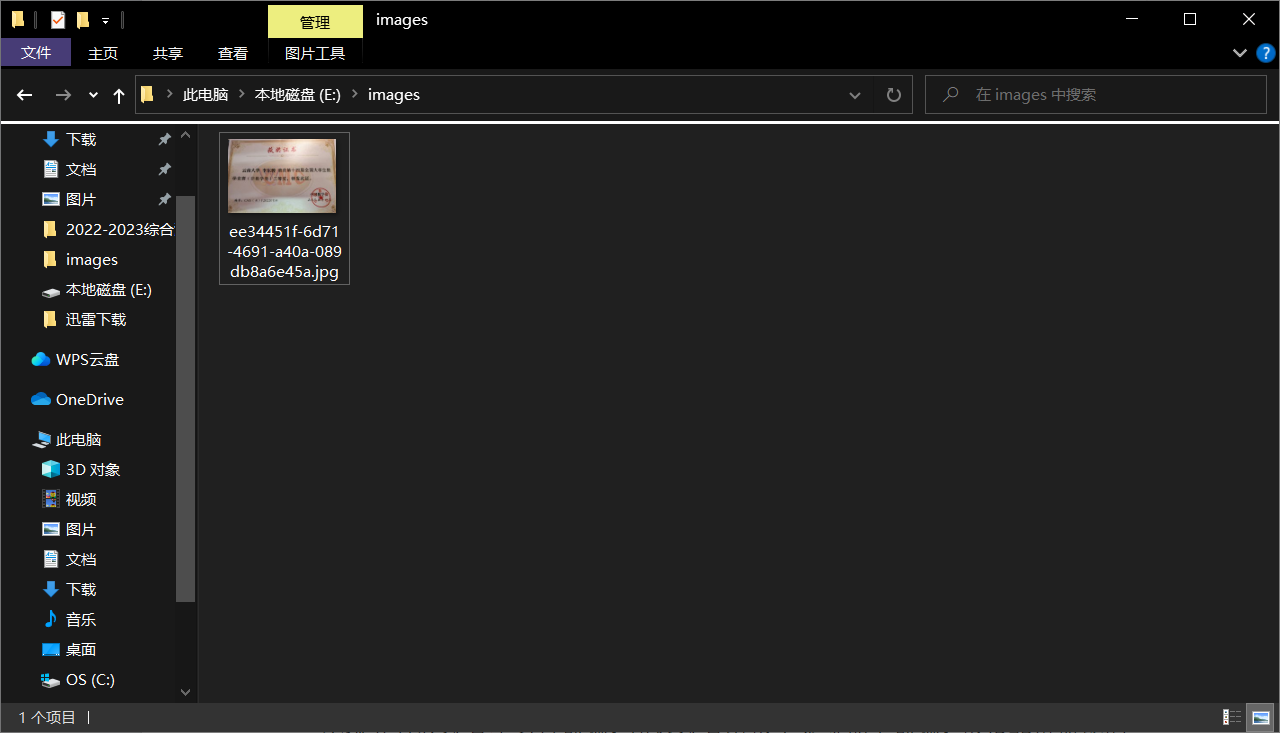


可以看到用户表中的顾客的消费总金额发生了改变，并且更新时间发生了变化，

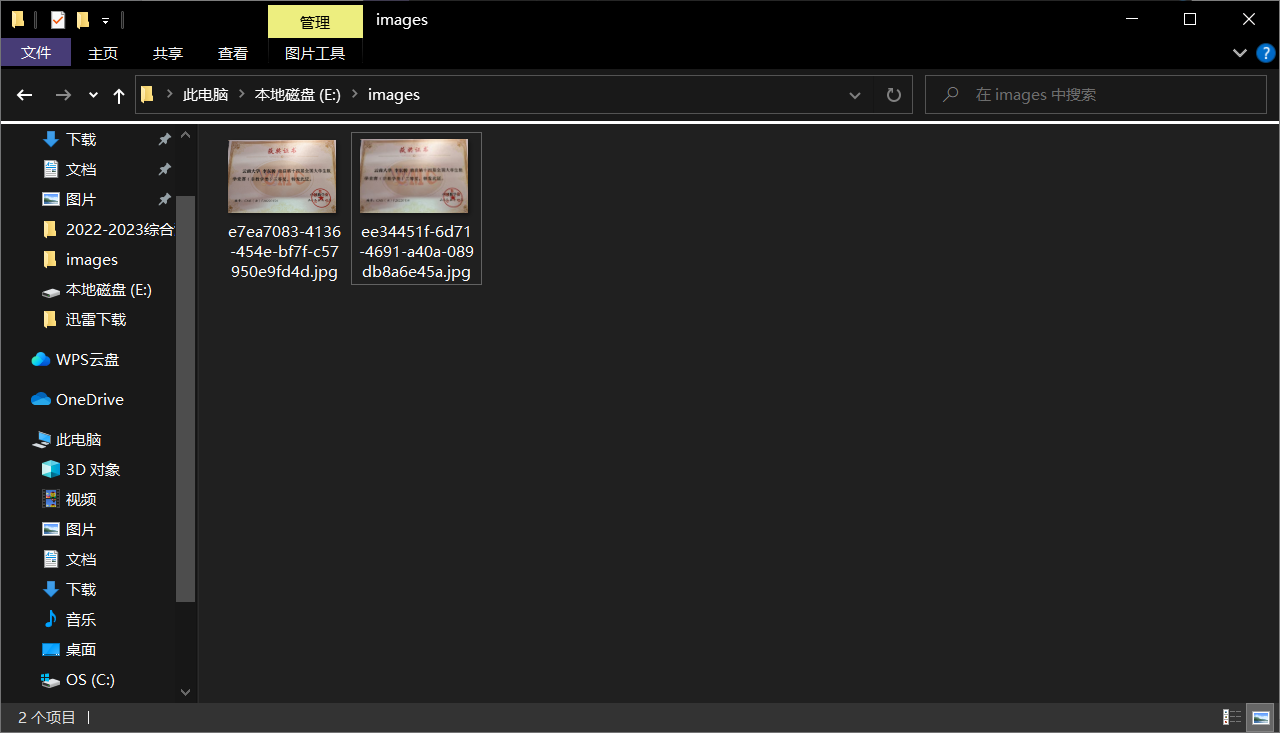
而且商品表中的商品的数量发生了变化。

最后因为商品的信息中可能会涉及到图片的信息，所以在这里我加了一个上传照片的功能：



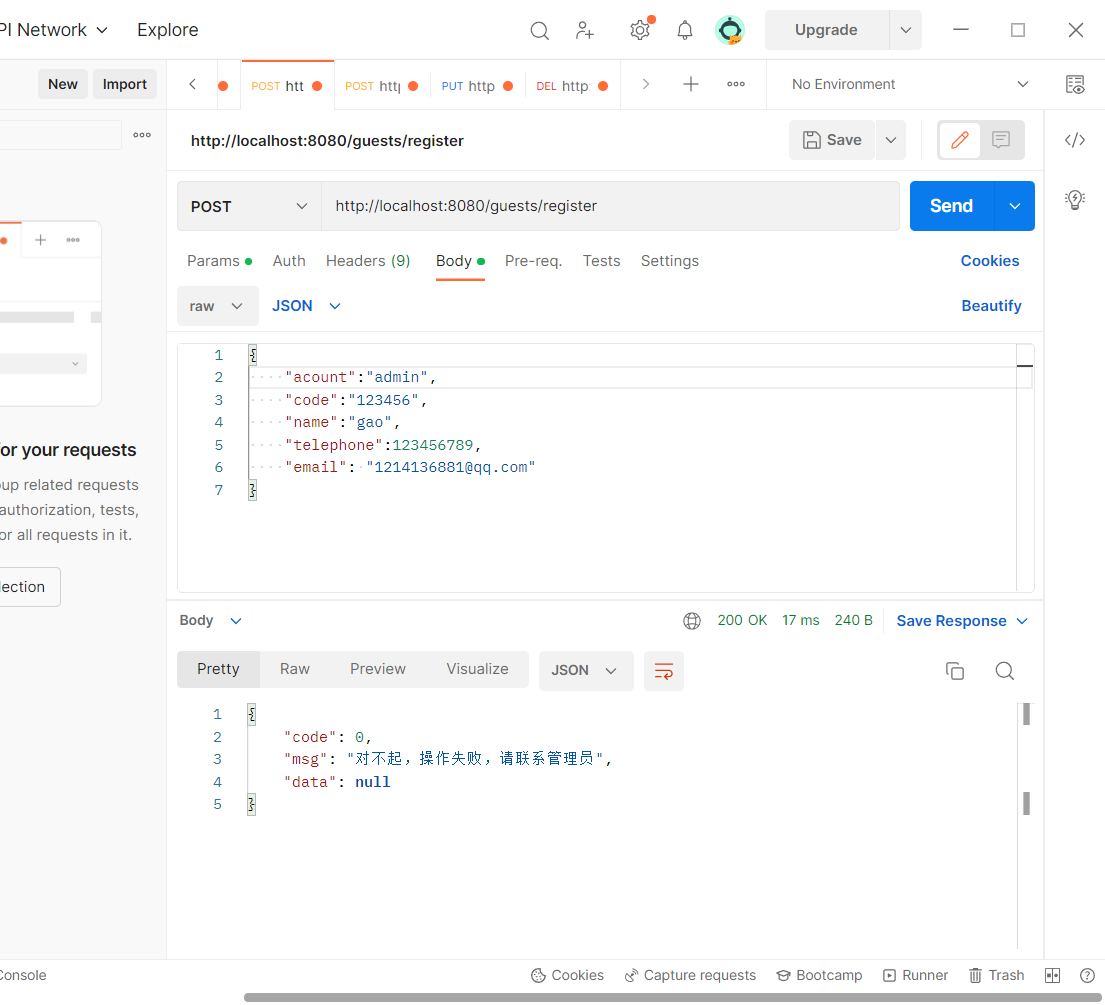


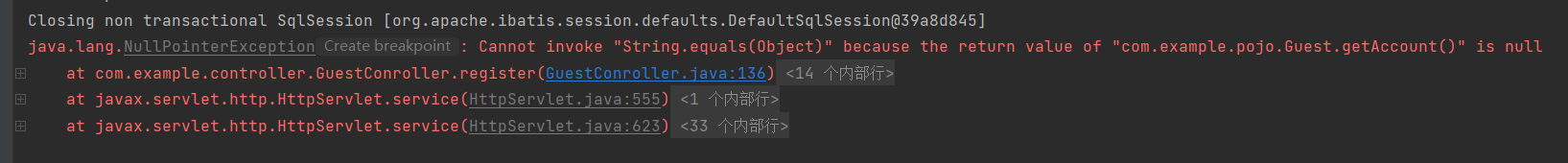
可以看到照片被成功的存到了对应的本地位置，当然这里也可以存到云服务中，并且当我们再次启用相同的功能存储相同的照片时：



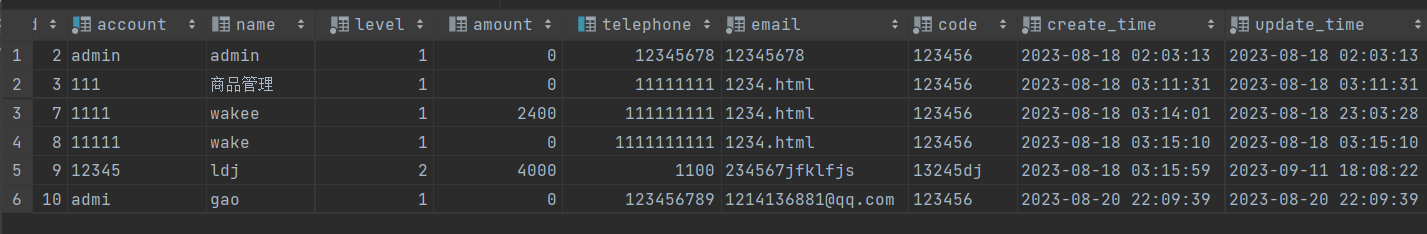
可以发现两次的名字会不一样。

最后就是异常处理，在这个简单的购物系统中我没有将异常进行细分，只是简单的对异常进行处理：





这里假如账号account被错误拼写成了acount时，会显示操作失败，请联系管理员。



并且这些错误的操作并不会执行，所以不会对数据库造成改变。

六、实验总结及体会

本次实验中，我在学校进行的几天中，我打算的是像大一写宿舍管理系统一样，用纯java代码写出一个在控制台中进行操作的代码，很多类也已经创建好了，但是我觉得这样写起来很麻烦，越写越乱，所以我放弃了一开始写的一切，学习了SpringBoot，Maven以及mybatis。然后重新开始写购物系统，写的十分顺利，而且因为有三个层，所以修改以及思路都十分的清晰，但是由于学的还是不深并且我没有学习过前端，所以没有给这个系统加上前端，这是非常遗憾的一点，其次就是我开始的时候打算的是给系统再加上两个表，一个为角色表，一个为权限表，角色表中我打算的是加入管理员，金牌用户等不同的角色，而权限表中加入不同的权限，给不同的用户提供不同的福利什么的，但是因为这里的角色表中会包含多种权限，但是我不知如何在表中存储这么多权限，所以在我尝试了两天之后，我无奈放弃了这个功能。其实这个系统还远远需要进行完善，就比如事务，有些操作必须同时进行，若某些地方出现错误，便需要将之前的所有操作全部撤回。

七、教师评语

(中文字体宋体、英文字体Times New Roman，小四，1.5倍行距，首行缩进两个字符，请删除此行)

(A4 双面打印，请删除此行)