

**《数据库技术实验》**

**期末大作业报告**

题 目： 数据库技术期末大作业

学生姓名： 黄珀芝

学 号： 20201050331

专 业： 计算机科学与技术

任课教师： 周小兵

评分（百分制）：

2023年1月

1. **需求分析**

**1.1 实验目的**

构建一个拥有**可视化**界面的网上购物-顾客管理系统平台

**1.2 实验仪器设备及软件**

Idea、数据库、sql

**1.3 实验方案**

（1）应用系统分析，建立该系统的功能模块框图以及界面的组织和设计；

（2）分析系统中的各个实体及它们之间的关系；

（3）根据问题描述，设计系统的类层次；

（4）完成类层次中各个类的描述；

（5）完成类中各个成员函数的定义；

（6）完成系统的应用模块；

（7）功能调试；

**1.4 用户需求**

1、系统中有两种类型的用户:**管理员，客户**

2、**管理员的功能如下**:

(1)登录

管理员账户预先设定账户名和密码，方便后台管理。

(2)密码管理

(2.1)修改自身密码

修改管理员的密码。

(2.2)重置指定客户/用户的密码

管理员可以重置普通用户的密码，但不能查看普通用户的密码。

(3)客户管理

(3.1)列出所有客户信息：

客户信息包括：客户ID、用户名、用户手机号、用户邮箱；

(3.2)删除客户信息：

删除客户信息之前要给出警告提示，请用户确认是否继续删除操作。

(3.3)查询客户信息

可以根据客户ID或者客户的用户名进行查询，也可以一次查询所有客户的信息。

(4)商品管理

(4.1)列出所有商品的信息

商品的信息包括：商品编号、商品名称、生产厂家、品牌、型号、库存量、商品价格、类别。

(4.2)添加商品的信息

商品的信息包括：商品编号、商品名称、生产厂家、品牌、型号、库存量、商品价格、类别。

(4.3)修改商品的信息

可以对上述的商品信息进行修改。

(4.4)删除商品的信息

可以删除商品，删除后无法恢复，请用户确认是否继续删除操作。

(4.5)查询商品的信息

可以根据商品名称、生产厂家、零售价格进行单独查询或者组合查询（例如：查询联想公司、1000以上的商品清单）。

(5)退出登录

退出管理员登录。

3、**客户的功能及需求**说明如下:

(1)注册：

用户名长度不少于5个字符；密码长度大于8个字符。

(2)登录：密码连续输入错误5次就锁定账户。

(3)密码管理

(3.1)修改自身密码

密码长度大于8个字符，必须是大小写字母、数字和标点符号的组合。

(4)购物

(4.1)查询商品信息

用户可以输入商品编号和商品种类名称将指定的商品直接购买。

(4.2)直接购买

直接选择商品购买。

(4.3)重置购物车中的商品

修改商品数量，当数量小于或等于0，则将该商品从购物车中清除。

(4.4)结账

模拟支付操作，购买前会再弹出询问是否确定购买。

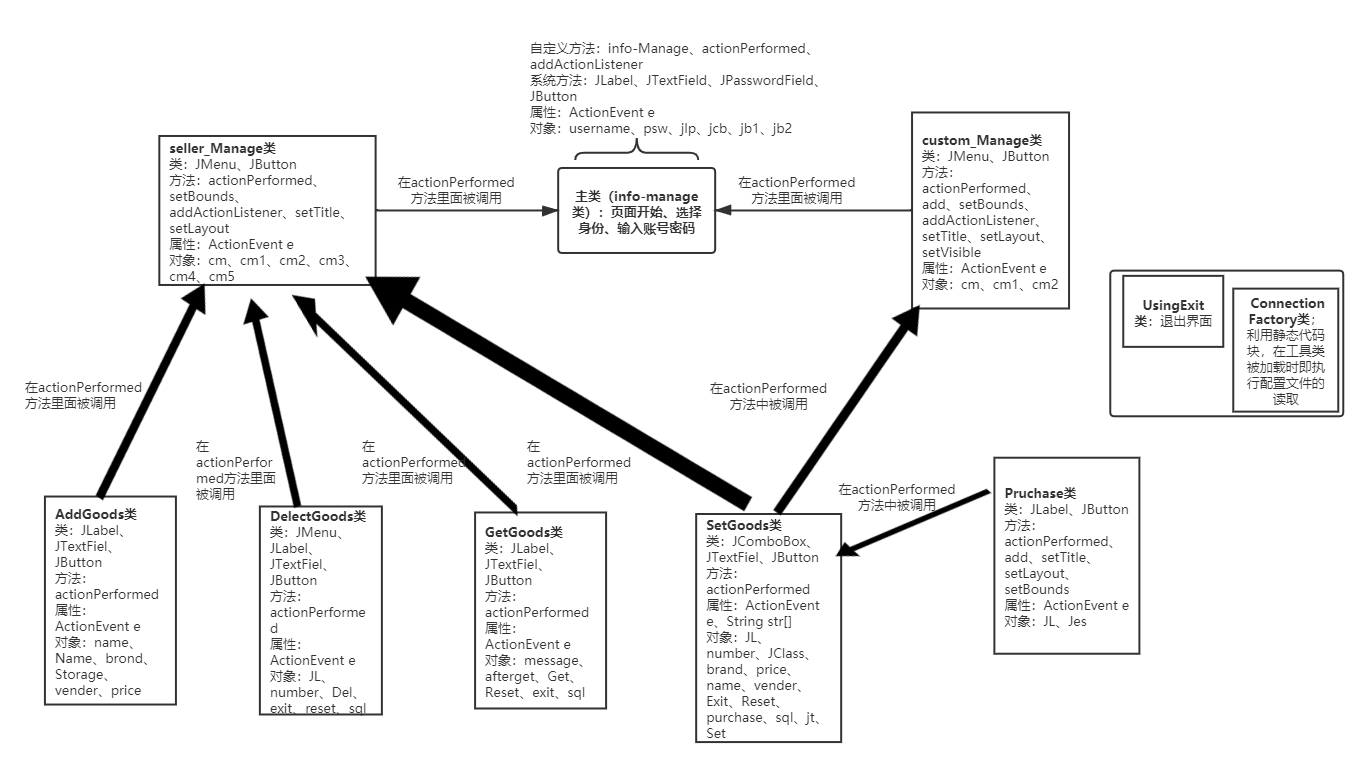
(4.5)查看购物历史

查看用户的购物历史：时间、购买的商品清单。

(5)退出登录

退出登录。

**1.5 实验步骤**

类图如下：

因为空间的限制，看不清图，所以把图的链接放在这里，老师可以点击查看（在总文件夹里面也以及放置了图片）：

附件在这里：（双击点开）



1.思考文件所给的购物系统要求内一共少而精的可以分成多少个来，分析需要的类与方法，以及各种类的调用

2.定下类的名称，尽量做到见名知其意

3.画出类图（UML图），使脉络清晰（我使用的画图软件为PROCESSON）

4.在类里面思考需要用到的方法与属性，填充类

5.在idea里面先创造好package与各种类，在其下不断填充

6.编写完成后build与run，是否可以运行且运行正确

**二、系统功能设计与划分**

最重要的过程是分类：

**先从总体思考：项目有多少类**:（全部都放在一个包下）

启动（Starts）的大类：有注册或者登录输入用户名和密码的界面，以及选择身份是顾客还是管理员（商家）；这是第一层大类。

输入账号密码，选择身份之后，来到第二层：即顾客界面（custom-manage类） 还是管理员界面（seller-manage类） 这里是第二层类，在starts类下面，被starts类调用。

**1.顾客在商品界面可以实现以下功能：**

进入顾客界面后，顾客只能选择查看商品和查看商品历史，其余操作是在选择这两个按钮之后才出现的，因此有两个平行的大类。

选择查看、点击、购买、删除商品，若是不购买就返回；购买商品以后实现结账，并且可以查看购买历史。这些功能需要一个大类来实现：SetGoods类。

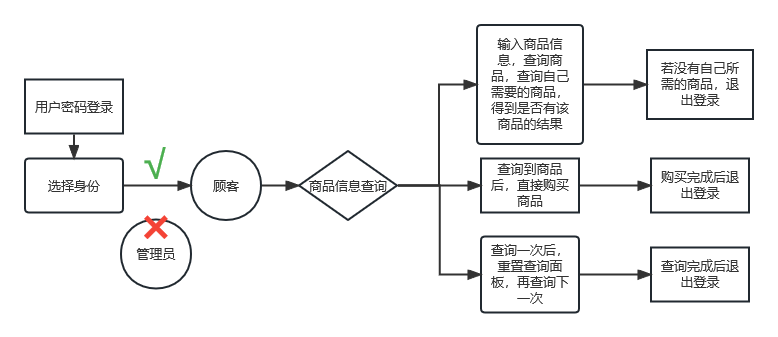
而购买功能是在顾客选择商品后才出现的，因此在SetGoods类下面还需要有一个Purchase类。购买后跳到下一个界面为结账，因此在Purchase类下面还需要一个CheckIn类。因此为：starts类-custom manage类-SetGoods类-Purchase类-CheckIn类。

**2.管理员在管理界面可以实现以下功能：**

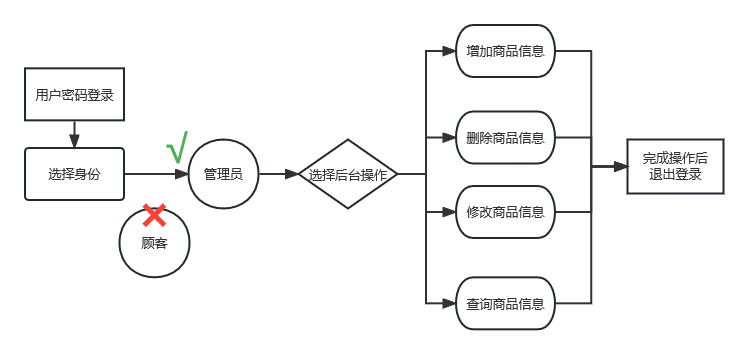
由于进入管理员的界面后，管理员可以选择的操作按钮有很多，而选择每个按钮之后的下一个界面并没有接下去的步骤，因此管理员只有平行的四个管理商品的大类：设置商品类（SetGoods类）、添加商品类（AddGoods类）、删除商品类（DelectGoods类）、修改商品类（GetGoods类）。

**三、数据流图**

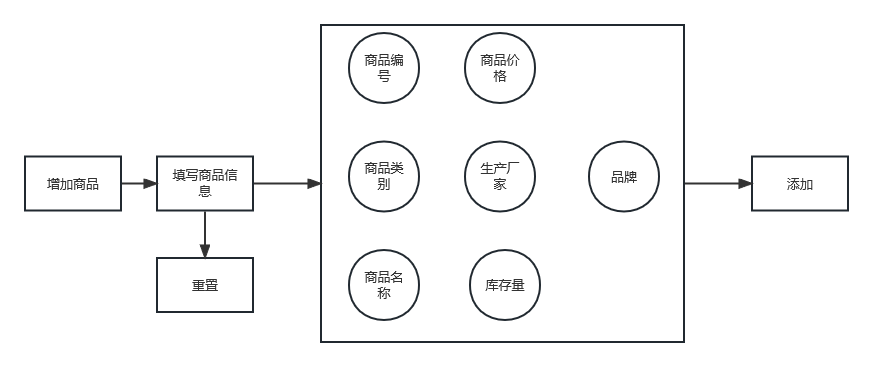
顾客操作分数据流：



管理员分数据流：

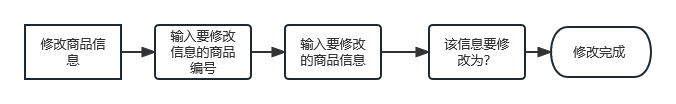


增加商品分数据流：

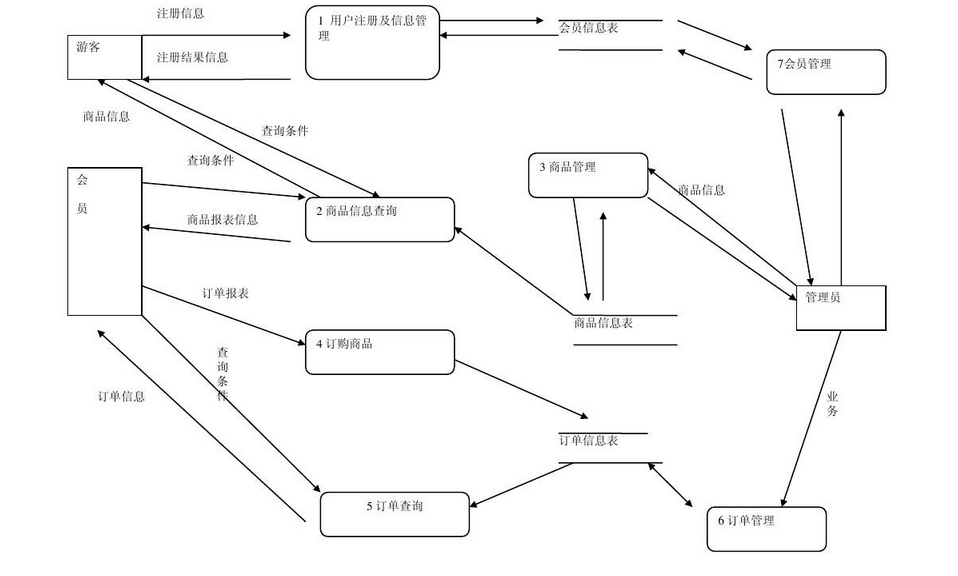


删除商品信息与添加商品信息类似。

修改商品信息分数据流：



总数据流：



**四、数据字典**

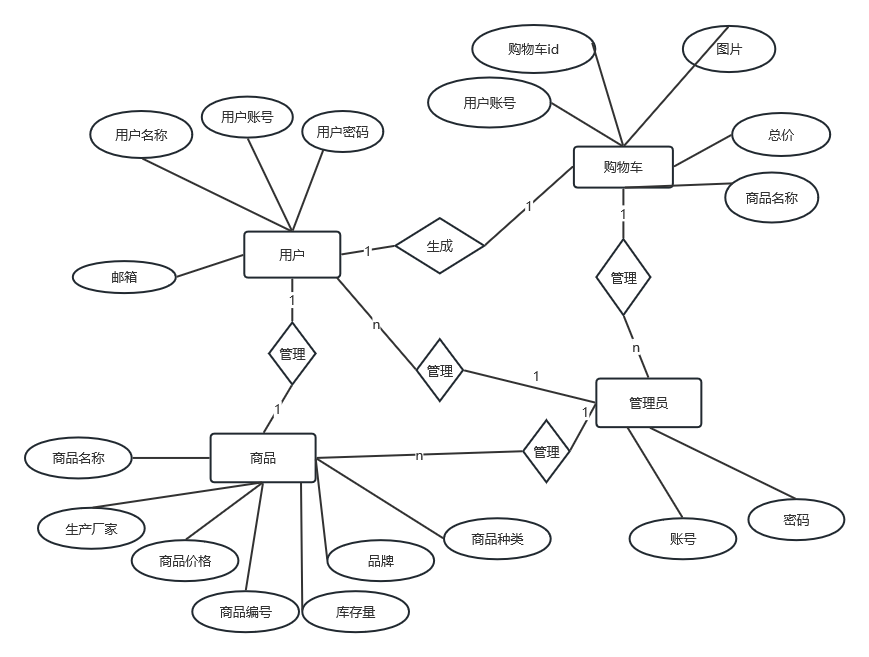
1. info-Manager类：用于存储用户和管理员的账户和密码。既然我们需要用户、管理员登录，那么肯定需要一个单独的类来管理用户、管理员的账户和密码，从而使系统的独立性更强一些。本次设计的用户和管理员只有一个账户和密码。（所涉及的修改密码与注册在基础版里还不会orz，已经尽力了，在后续升级版里会添加上去）
2. Seller-manager类：用于管理员信息以及操作的管理。该类实现了一些基本功能：比如说查询、修改、删除等等。当进入到管理员操作管理这个菜单时候，需要对各个商品信息进行许多操作，为了方便管理这些操作，并且考虑到后序扩展性，这里把商品管理信息的所有功能都抽象出来放在此类中，上一级菜单通过调用该类中方法实现商品信息管理。
3. custom-manager类：类似于seller-manager类，不再详细说明了。
4. add-good类：管理员添加商品，只有管理员可以操作，因此只被seller-manager类调用。
5. Delect-good类于get-good类：都是管理员进行的操作，只被seller-manager类调用。管理员删除商品或者修改商品。
6. set-good类：管理员与顾客都可查询商品，被seller与custom类共同调用。
7. Pruchase类：只有顾客可以购买商品，且在购买之前，是先要顾客可以查询（或者说看到）此商品，所以应该是在set-good类中被调用。

连接数据库的代码：database：

driver=com.mysql.jdbc.Driver  
url=jdbc:mysql://localhost:3306  
user=root  
password=root

总之，上面的类都是经过一些功能模块划分后抽象出来的，有些地方也并不一定合理，主要还是需要看需求，根据不同的需求制定不同的方案。比如set-goods类，这个类的设计十分符合面向对象的思想，举个例子来看，如果购物系统中需要很多商品，比如食品、化妆品、日用品等，那么如果我们一个个写这些商品，会使得系统代码十分冗余，因为商品的属性基本一样，所以我们就可以抽象成一个类，从而在需要什么商品的时候只定义一个对象，然后赋予一定的属性即可，比如需要手机、电脑，那么我们只需要new一个set-goods类的对象，然后在需要食品的时候设置其属性为食品，在需要化妆品的时候设置其属性为化妆品，需要什么设置什么即可，这样就使得代码得到了简化，也使得结构比较清晰。

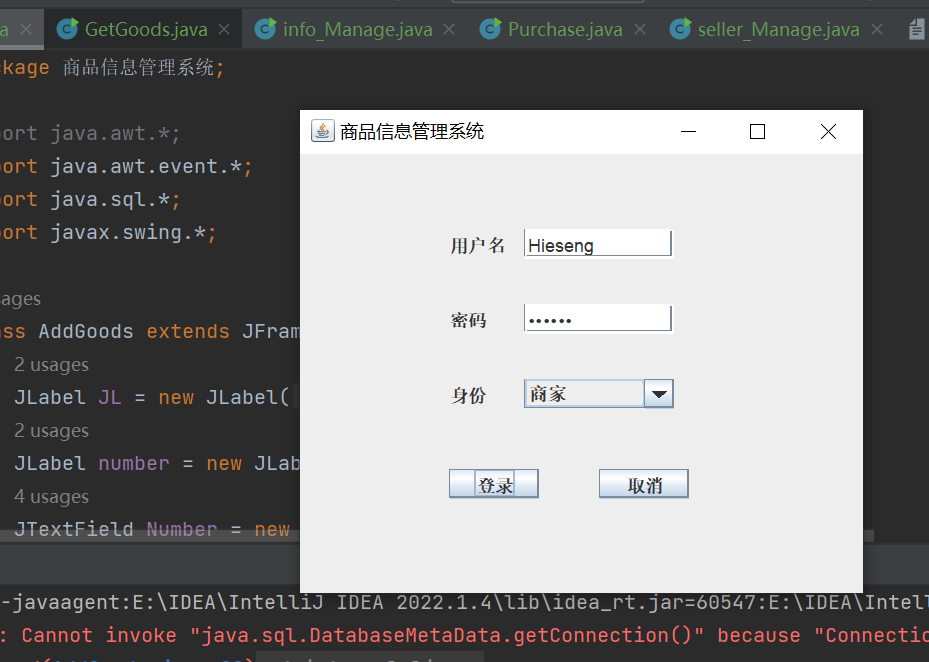
1. **基本ER图**

****

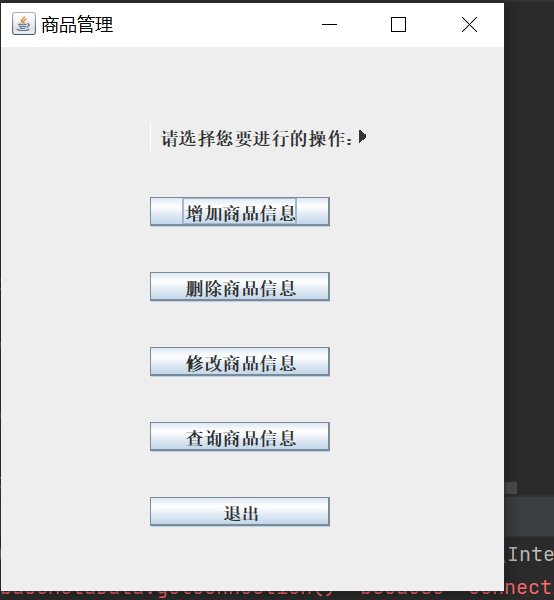
1. **展示**

**商家管理者页面：**

**登录界面：**

****

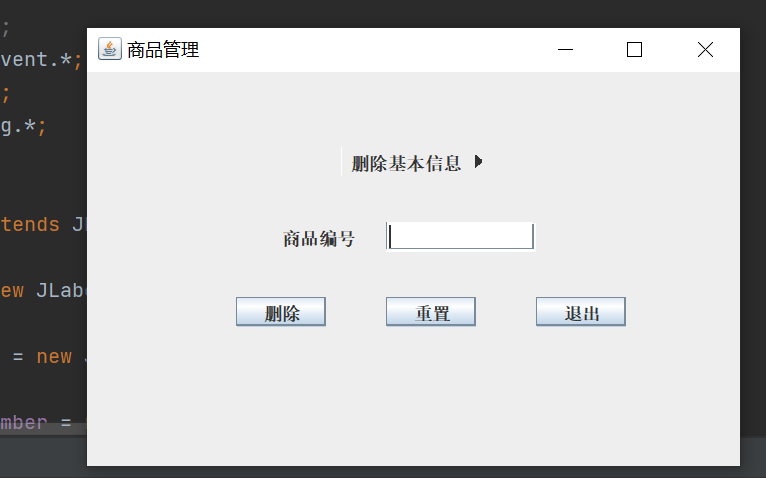
**商家选择操作界面；**

****

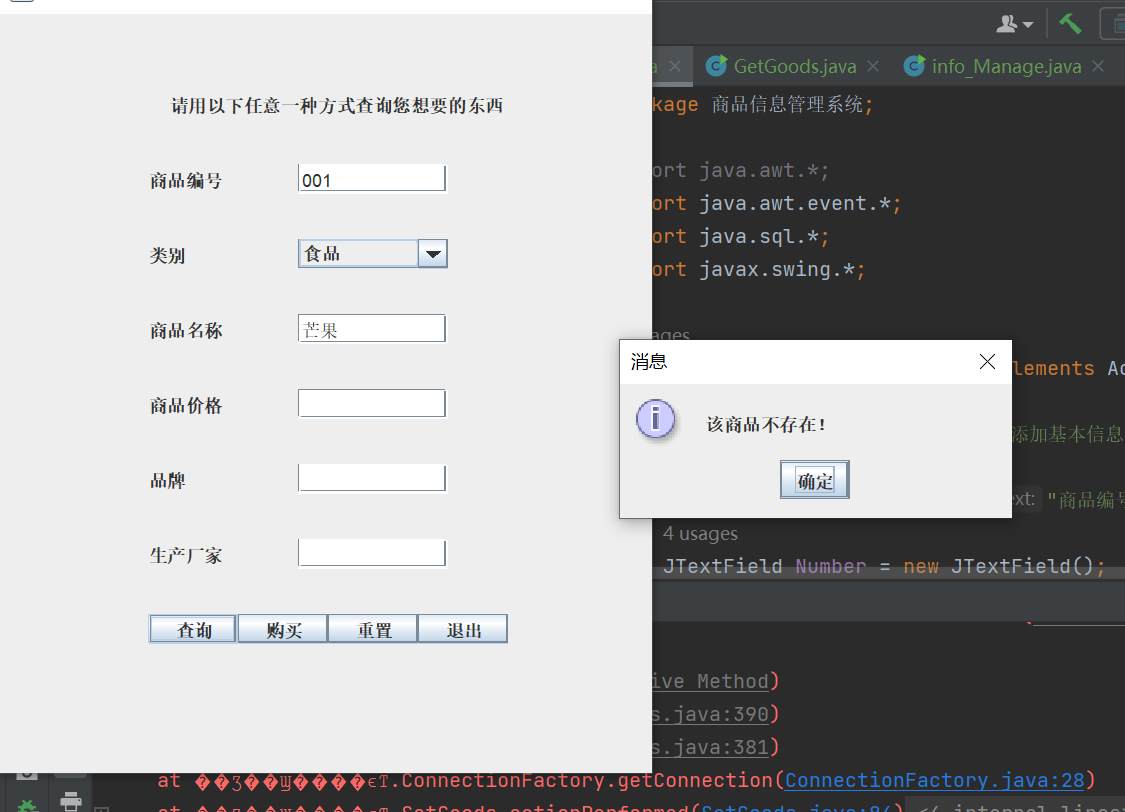
**添加商品信息界面：**

****

**删除商品信息界面；**

****

**查询商品信息界面：**

****

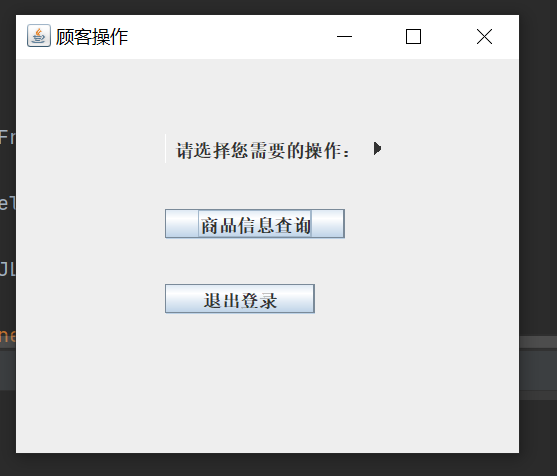
**商家修改商品信息界面：**

**顾客界面：**

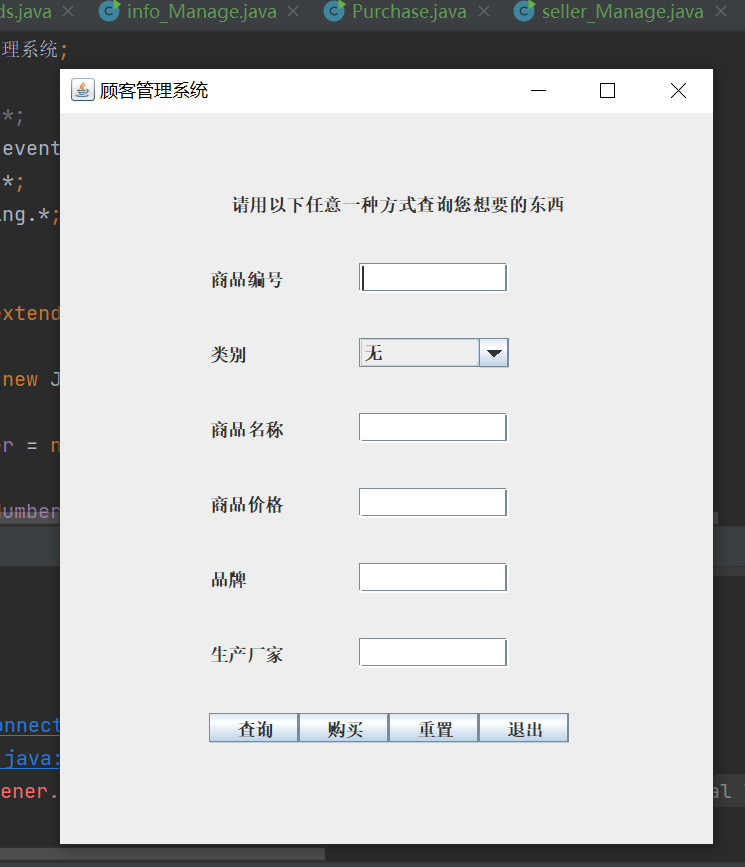
**顾客登录界面：**

****

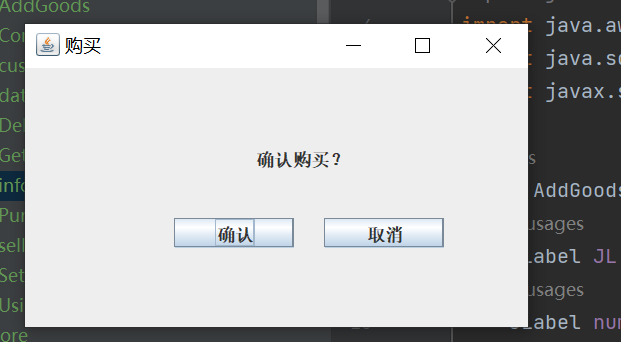
**顾客选择操作界面（顾客只有查询购买权限）；**

****

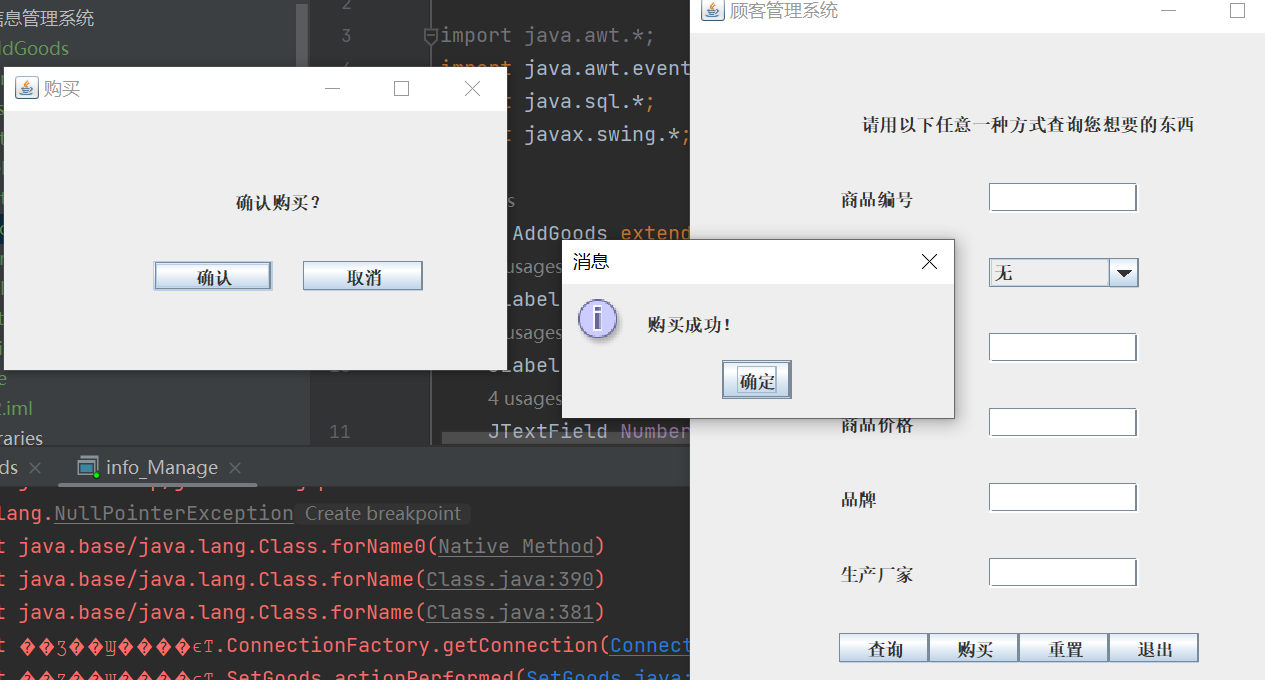
**顾客查询自己所需商品界面：**

****

**顾客购买商品前的再次确认界面：**

****

**购买成功界面：**

****

1. **总结**

今天碰到了一个错误：在显示“顾客购买界面”的时候需要显示商品的名称，而由于我拿到的是这个订单所对应的商品的集合，所以需要把集合转变成字符串，所以使用了以下方法：

　　private String listToString(List<Prod> prodList) {  
        StringBuffer sb = new StringBuffer();  
        for(Prod p : prodList) {  
            sb.append(p.getProdName()).append(",");  
        }  
        String str = sb.toString();  
        str = str.substring(0,str.length()-1);  
        System.out.println(str);  
        return str;  
    }

出现了java.lang.StringIndexOutOfBoundsException: String index out of range: -1，原因就出现在

str = str.substring(0,str.length()-1)这一行，我在订单成功提交之后的页面又刷新了一次，又创建了一个新的订单号，但是却没有商品传过来（因为代码中当订单完成之后就会把购物车清空了，所以此时购物车中是为null），所以此时的str的长度其实是为null的，如果此时再对str进行截取字符串的操作的话，就会出现StringIndexOutOfBoundsException。

在这次实践的编写开发系统中，我明显感觉到了java面向对象这个根本建立基础，以及类的属性和方法的构造的重要性。编写程序，是为了解决实际问题，而解决实际问题的方法，就是从面向对象里面来的。

如果按照面向流程的思想来设计，就会划分很多功能模块，然后按照流程一步步走就行，但是采用面向对象的思想来设计，那么应该如何考虑设计框架呢？面向对象的主要思想就是将一些需求抽象为许多类，然后建立这些类之间的联系，通过不同类之间的协同合作，就可以实现所有的功能。所以，现在的主要任务就是如何合理地抽象出这些类，以及这些类要实现什么功能，类之间的联系又是什么？

这次的购物管理系统开发，如果直接思考，会发现根本无从入手，我要怎么做才能分开管理员与顾客？增加、修改、删除商品等等的功能，我要怎么实现？我要用什么类？如果每个小功能都创建一个类，这好吗？越细分的类越好吗？类的数量应该有限制吗？这些类之间的关系是什么？

在我最开始一收到文件要求时，我直接打开了idea就想下手编写代码。过了几分钟之后，这些问题一个个的突然浮现出来，我甚至不知道要怎么开始。

周围一看，大家也都束手无策。

于是我才开始不着急的思考。其实通过一个简单比喻可以看透面向对象的本质：我要把大象塞进冰箱里。那么这个过程如何实现？

对于java来说，就是很简单的直接想法：我是执行的人 而大象就是我要执行操作的对象 最后的结果就是赛进冰箱。此时的对象已经找到了。那么接下来就是怎么完成这个”塞进冰箱“的动作。那就是：打开冰箱门、把大象塞进去、关上冰箱门。而这三步动作，其实就是类。

所以明白了这点，知道了最关键的一步是不断优化自己罗列出来的类，我从一开始狂打了十三个类最后缩减到九个类（这还是建立在我所编写的购物系统里，没有注册和通过wx等方式支付等等功能的情况下（因为有点超过了我个人能力，在后面的五个升级版里面我会努力尝试添加进去））。之后的事情可以说是水到渠成，特别是在我先做出了类图之后，一切按照上面不断填充润色即可。

剩下的事情就是考验java掌握的基本功了。

在这次的购物系统编写中，除了自身思考带来的受益外，还有很多其它途径的受益，更还有不断出现的error使我几乎找不出来痛不欲生，以及方法和属性的设置之多，名字为了直观看出来意义而取的英文之长带来的时刻凌乱，这些都带给了我无法言语的”满足“与无声无息中的受益。甚至只是制作类图的时候，都花去了我很多时间，而且制作出来的也没有我预期里的直观与美观......就算文字详细写下了每个类后，制作类图并不会变得简单...

这些漫长的过程只是编写完程序之前。如果说是爬越山谷，那么思考与编写的过程是那前九十五步，而build就是第九十六、七、八步（无数莫名其妙的error的困扰，最要命的是idea提示的存在error的哪一行，但是真正出错的地方其实在此行的上面或者下面），那么run就是最后一步。

最大的感悟就是：无论怎么样的创造，都是漫长而已在不断试错与失败中磨砺出来的。在不久之后的将来，在我已经拥有了更加多的经验、学习到了更加多的知识、了解到了更多方面之后，我一定会觉得此时编写的这个购物系统非常的幼稚与不成熟，甚至不可思议于为什么我还要思考那么久做那么多准备修改那么多次才完成一个基础版，但是我一定会尊重我此刻真实的努力与创造。正是因为刚刚开始不断努力，才会一步步成熟。而这个过程，恰好是一种课外自动参与学习的习惯。因为这里面有太多我不懂的知识，所以我不断的在b站看尚硅谷的java进阶课程，不停的在csdn上搜索问题的解决办法。甚至所给的文件里面提及要交【项目工程文件】，我第一反应是这是什么？是可以直接运行的jar包？还是源代码？还是直接复制文件夹？在这个不确定的过程中，我却知道了很多idea的功能，知道了cookie是什么，怎么导出jar包，甚至还隐隐约约感觉到了架构知识。

这些知识是我无法从课程上，或者甚至可以说在普通练习中获得的。需要一定的压力，一定的难度，一定的时间充裕度，因此这是一个很好的暑假学习机会。

在后面的升级版中，反而越是看起来简单的要求对我而言更加无法理解。就好像读取txt，如果是直接按照前人已经反复修改好的代码版本直接复制进去，那么呈现出来的只是直接读取出来....并没有和我原本的基础版购物系统形成联系。而要求里面看起来很简单的文字，我却也读不太明白..因为可以说是CSDN上的教学也是直接开始，我有时候有很多东西的最本质的理解都没有，这就造成了我的迷茫。反而在假期内认真研究了一下其它的方面，通过观摩各种大佬的教学与编程，对数据库有了自己的理解。