**水果源项目文档**

**项目名称：水果源**

**项目经理：彭佳豪**

**项目助理：王天戈、邵宇果**

**文档设计：张平、郑德洋**

**汇报文档：崔鑫鹏**

**讲解答辩：何木帆**

**目录**

一、概述……………………………………………………………………

……1.1开发背景…………………………………………………………

……1.2开发目标…………………………………………………………

……1.3设计原则…………………………………………………………

二、需求分析………………………………………………………………

……2.1需求陈述…………………………………………………………

……2.2操作用例…………………………………………………………

……2.3功能分析划分……………………………………………………

三、总体设计………………………………………………………………

……3.1系统建模…………………………………………………………

四、接口设计………………………………………………………………

……4.1类图设计…………………………………………………………

……4.2接口设计…………………………………………………………

……4.3数据库结构设计…………………………………………………

五、详细设计………………………………………………………………

……5.1程序流程图………………………………………………………

六、实现……………………………………………………………………

……6.1测试要点…………………………………………………………

……6.2测试结果和总结…………………………………………………

一、概述

1.1开发背景

在经过整个小组的深思熟虑后，我们小组决定开发一个居民可以线上购买果蔬的APP，同时在深思熟虑后，发现此方面十分可行，特别是将其与我们的种植户直接对接，我们只需要把持品控，并与送货上门相结合，大有作为

1.2开发目标

我们的开发目标是创建一个线上的大型的一个跨区县市的购物平台，方便对于不便买果蔬，或者优质的果蔬种植户难以打出名气，提升他们的竞争力，同时这也是在提升我们的竞争力，我们的目标是瞄准全国能够提供优质果蔬的种植户，提高他们在高端市场的竞争力，同时我们也希望将这些优质的食材提供至全国有名的食府，我们不但瞄准下沉的普通家庭的市场，也瞄准了我们的高端市场，同时在以后争取将我们的带入全世界

1.3设计原则

(1)可靠性 用软件系统规模越做越大越复杂，其可靠性越来越难保证。应用本身对系统运行的可靠性要求越来越高，软件系统的可靠性也直接关系到设计自身的声誉和生存发展竞争能力。软件可靠性意味着该软件在测试运行过程中避免可能发生故障的能力，且一旦发生故障后，具有解脱和排除故障的能力。软件可靠性和硬件可靠性本质区别在于：后者为物理机理的衰变和老化所致，而前者是由于设计和实现的错误所致。故软件的可靠性必须在设计阶段就确定，在生产和测试阶段再考虑就困难了。

(2)健壮性 健壮性又称鲁棒性，是指软件对于规范要求以外的输入能够判断出这个输入不符合规范要求，并能有合理的处理方式。软件健壮性是一个比较模糊的概念，但是却是非常重要的软件外部量度标准。软件设计的健壮与否直接反应了分析设计和编码人员的水平。

(3)可修改性 要求以科学的方法设计软件，使之有良好的结构和完备的文档，系统性能易于调整。

(4)容易理解 软件的可理解性是其可靠性和可修改性的前提。它并不仅仅是文档清晰可读的问题，更要求软件本身具有简单明了的结构。这在很大程度上取决于设计者的洞察力和创造性，以及对设计对象掌握得透彻程度，当然它还依赖于设计工具和方法的适当运用。

(5)程序简便

(6)可测试性 可测试性就是设计一个适当的数据集合，用来测试所建立的系统，并保证系统得到全面的检验。

(7)效率性 软件的效率性一般用程序的执行时间和所占用的内存容量来度量。在达到原理要求功能指标的前提下，程序运行所需时间愈短和占用存储容量愈小，则效率愈高。

(8)标准化原则 在结构上实现开放，基于业界开放式标准，符合国家和信息产业部的规范。

(9)先进性 满足客户需求，系统性能可靠，易于维护。

(10)可扩展性 软件设计完要留有升级接口和升级空间。对扩展开放，对修改关闭。

（11）安全性 安全性要求系统能够保持用户信息、操作等多方面的安全要求，同时系统本身也要能够及时修复、处理各种安全漏洞，以提升安全性能。

二、需求分析

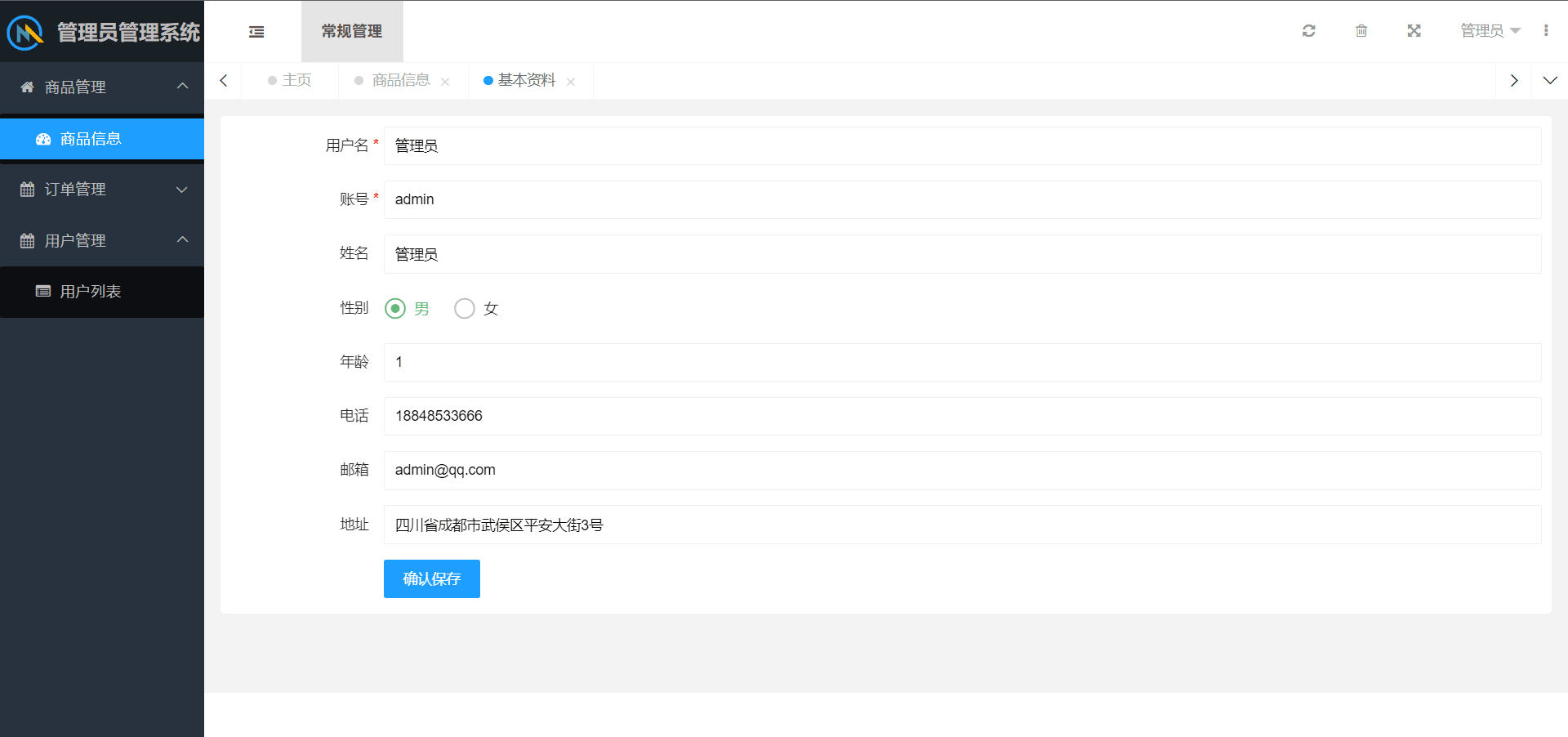
2.1需求陈述

1. 我们需要将用户进行分割，将用户分为我们的普通用户与管理员用户，其中普通用户又分为购买的用户以及出售的用户，其中出售用户与管理员用户都是通过我们的网页端进行登录，而购买的用户则是通过移动端APP进行登录，方便我们的购买用户进行购买、查看，同时管理员通过网页端也能够更加直观、方便的对用户进行检测管理
2. 购买用户需要购买时进行登录，绑定支付平台后进行购买
3. 购买用户与管理员用户可以对订单进行查看管理，而出售用户不能够对订单进行管理，发生错误后由管理员进行查错控制
4. 购买用户可对自己的信息进行修改，出售用户也能够对自己的出售物品进行修改，管理员用户能够对用户进行操作，也能够对订单以及商品信息进行管理
5. 商品信息包括介绍，分类，价格，库存，类型，看到的图片
6. 当商品被卖出后，该商品的库存会自动减少相应的数量
7. 对商品进行搜索时，可以搜索出相应的正在出售的商品
8. 可以对商品信息等进行打印操作，用于对我们的商品的出售等有更加详细的查

2.2操作用例

**管理员：**

（1）管理员对于个人信息记性修改

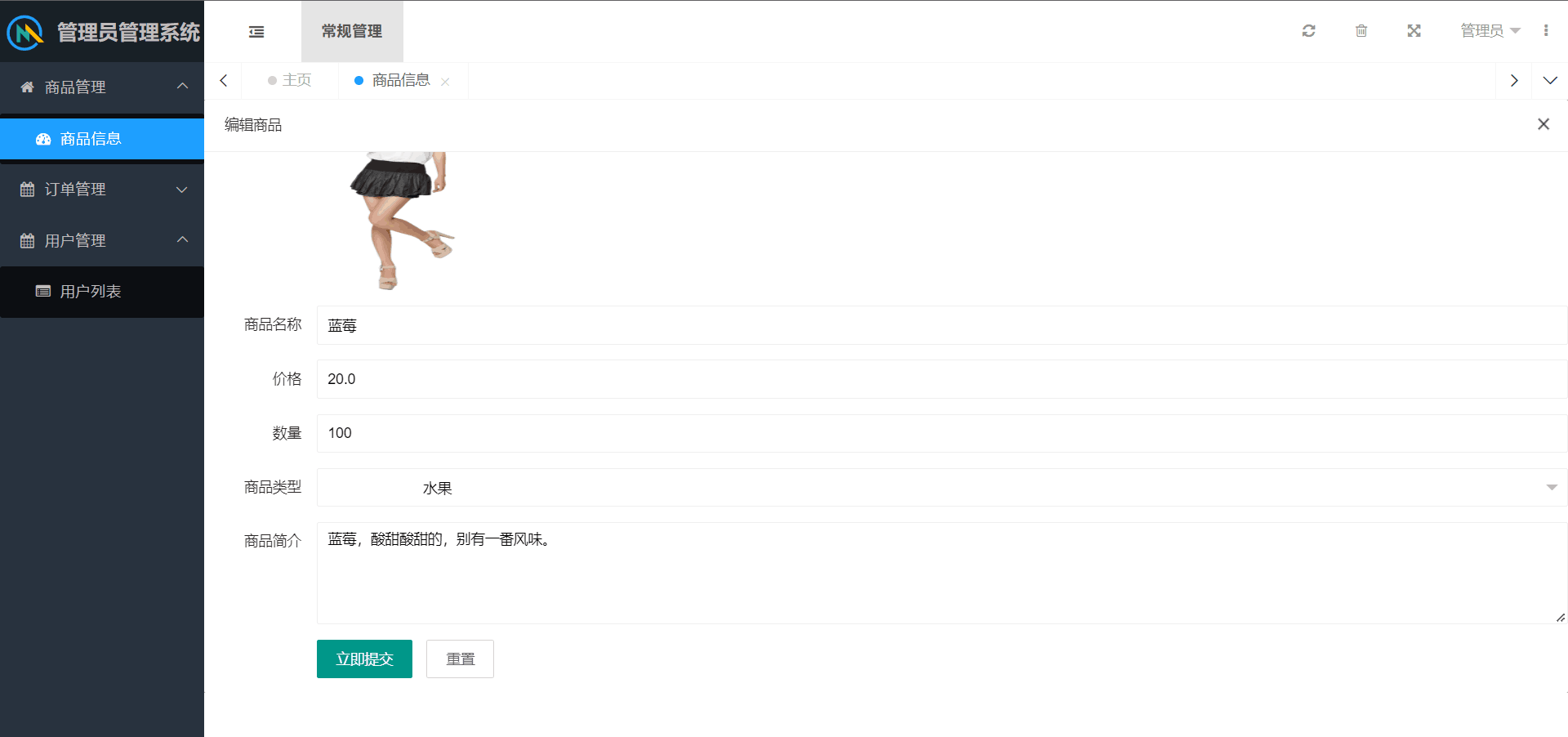


在将个人信息进行修改后点击下方确认保存后进行保存

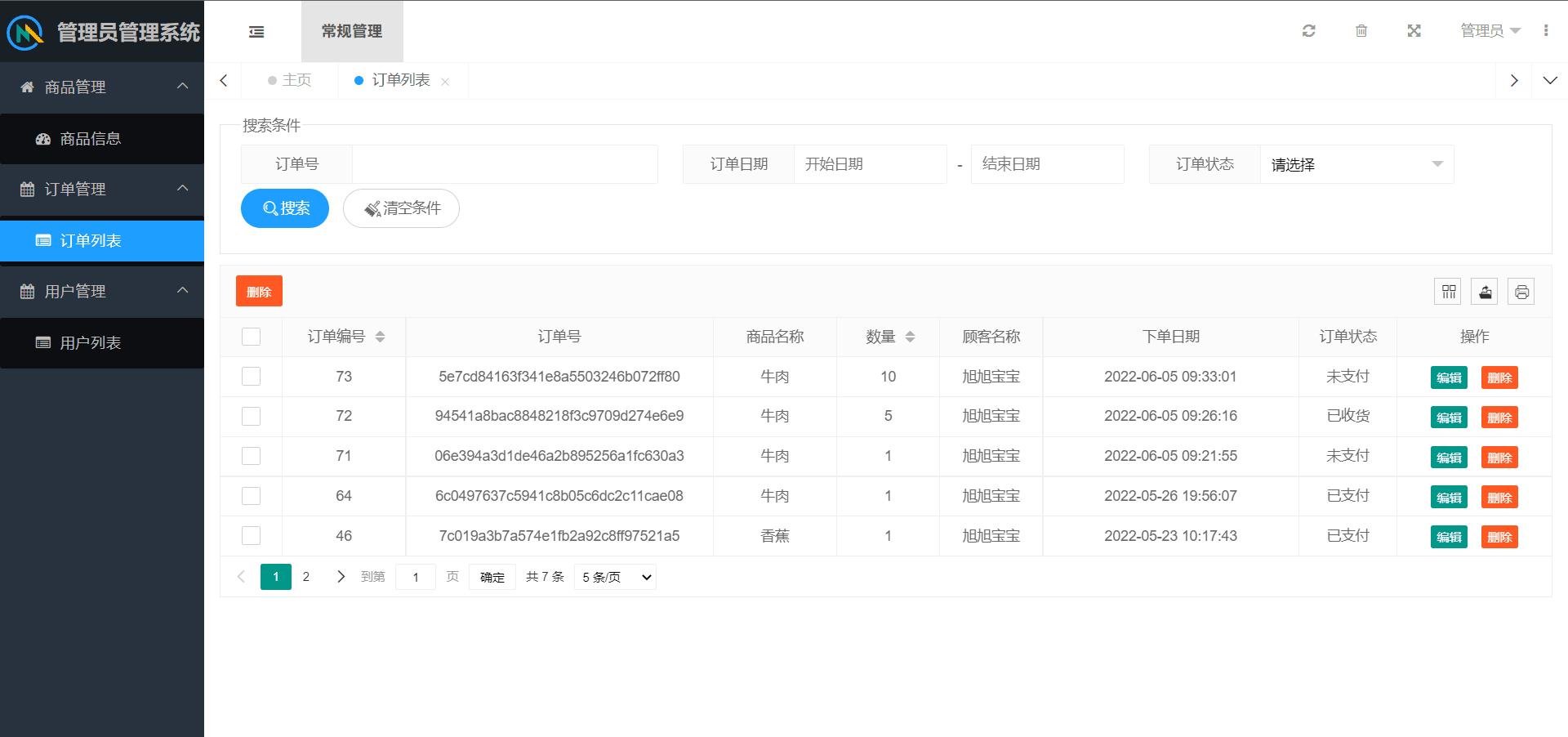
1. 管理员对商品信息的修改



管理员可通过搜索框对商品进行搜索，同时也能够对商品进行删除，同时也可通过勾选前方的方框进行多个操作

管理员可以通过点击编辑对商品进行编辑，点击下方的立即提交可以对商品信息的修改进行保存

1. 管理员对于订单的管理



管理员可以通过上方的搜索框对订单进行追查，同时也可通过右上角的功能栏对于订单进行打印，更加直观书面，同时也能够如同商品的管理一样，对订单进行操作，也能够直观的查看订单的状态

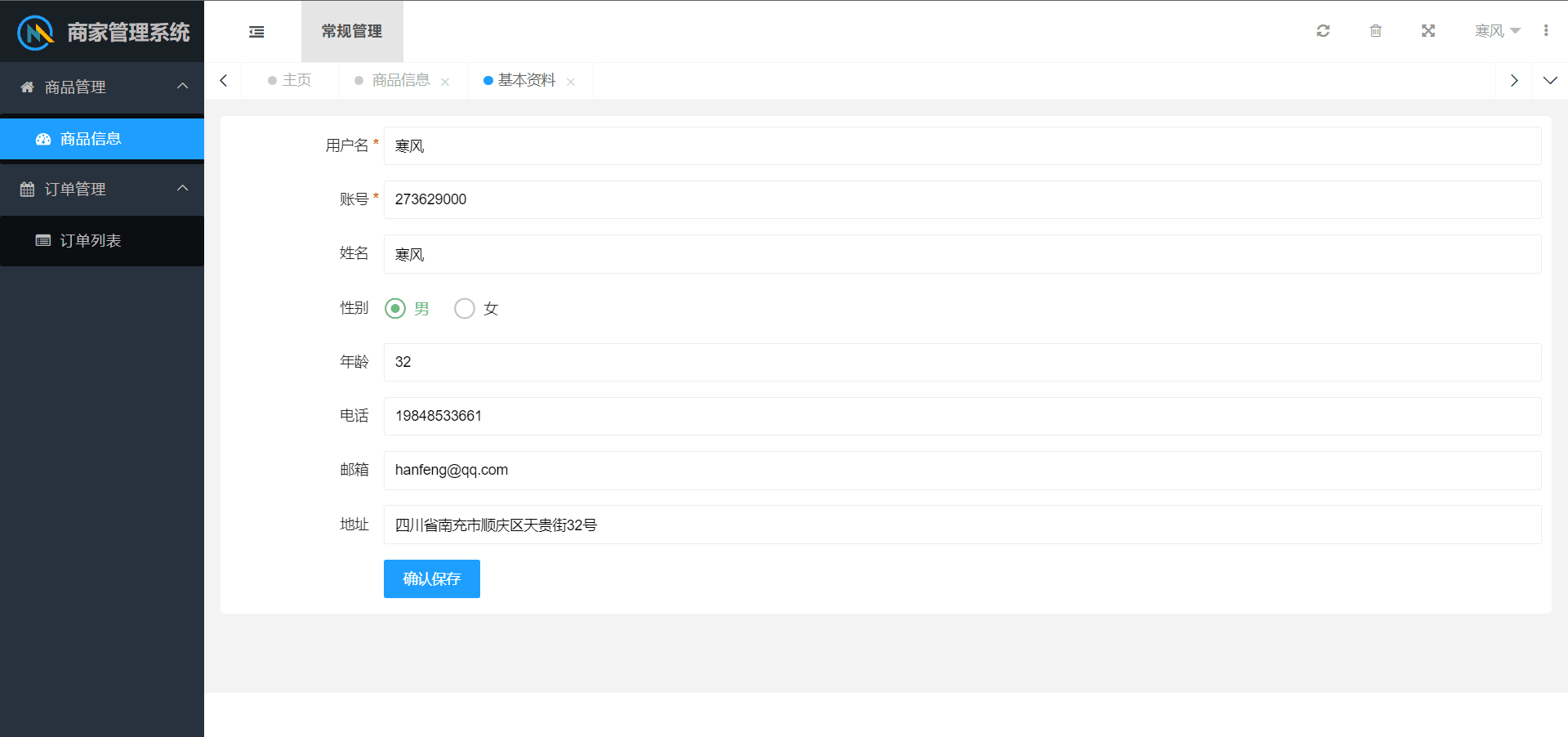
1. 管理员对于用户的管理

![B81TBR$5V](0`Z8W%P656(1](data:image/png;base64,)

管理员能够对于用户进行删除用户等操作，也就是当用户自行进行注销后的操作，同时也能够对于用户的信息进行修改

**商户：**

（1）商户对个人资料的修改

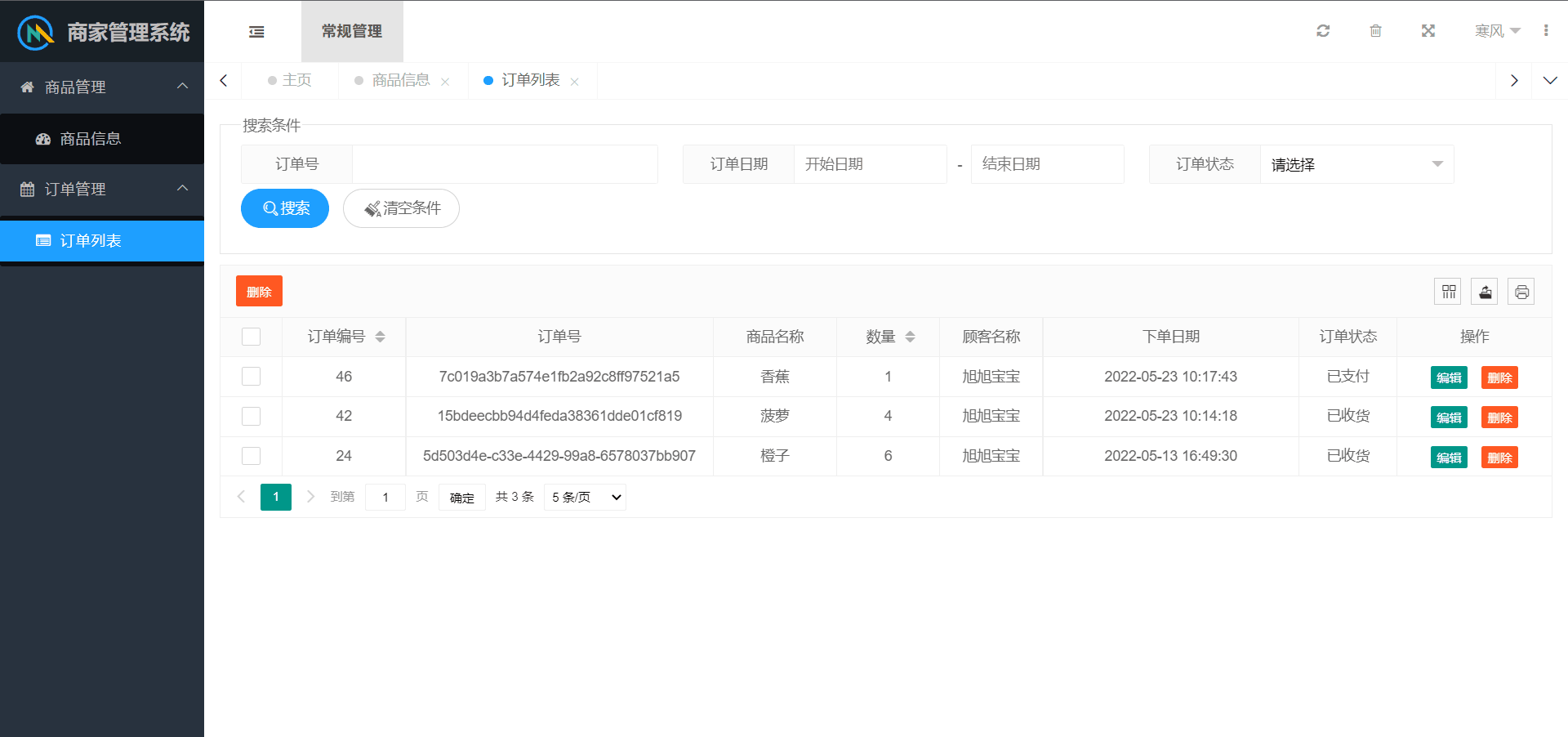


商户也可以与管理员相同的对个人资料进行管理，也是通过对下方确认保存按钮，对于修改的信息进行管理，同时，商户只能对商品和订单进行管理，无法对用户进行管理

1. 商户对商品信息管理

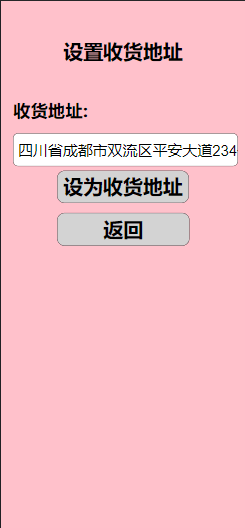


1. 商户对订单信息管理



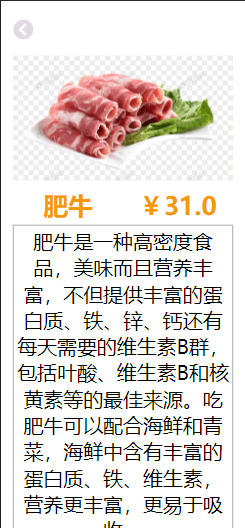
**购买用户：**

1. 个人信息等修改



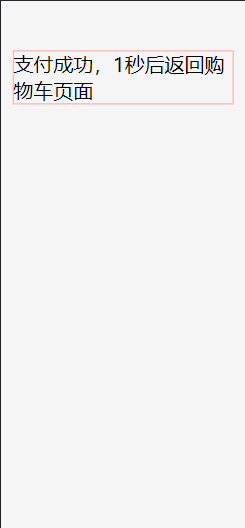
个人信息、订单信息、收货地址等可通过该界面进行修改，当然，钱包不能自行修改其金额

1. 主页能够进行搜索



比如输入西瓜进行搜索，会跳出西瓜的界面,我们还可以通过点击上方的蔬菜，肉类等对食材进行分类查看，同时也可点击打开进行细看介绍，购买等

1. 购买以及确认收货

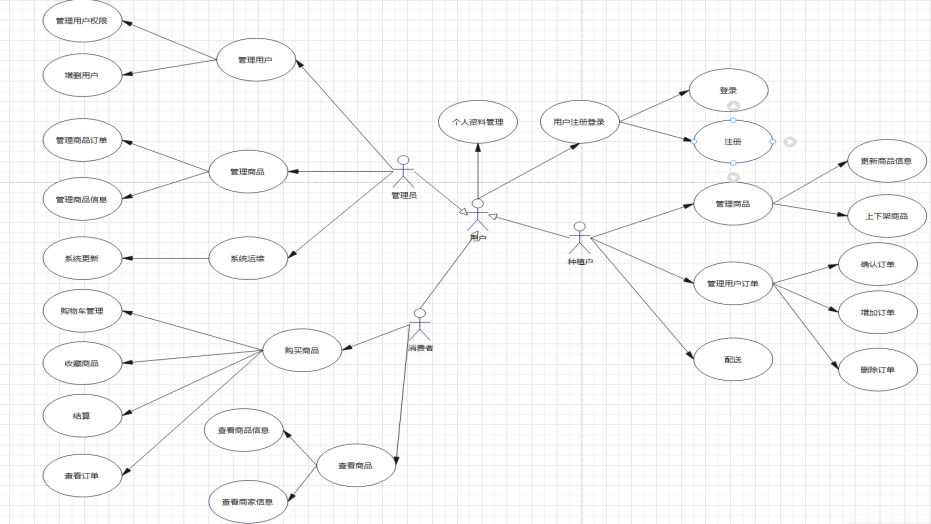


以上就是在进行购买的时候，我们的结算以及确认订单

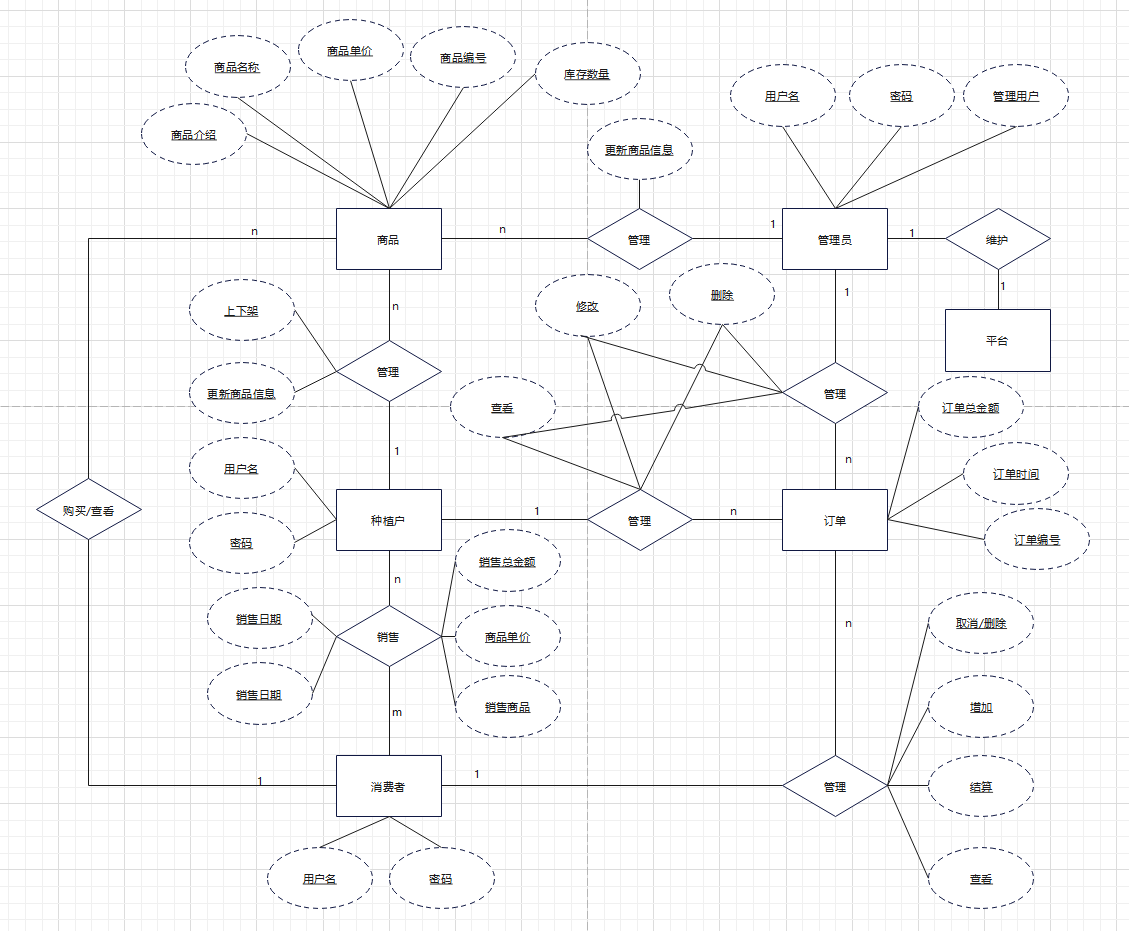
**最后，我们在网页的登录端商户和管理员是公用的，只是通过他们的数据库中表的不同进行区分，这样的设计是由于我们开发经营不足造成的，但是由于时间上的问题，我们不得不继续进行下去，只能以期后续进行修改**

三、总体设计

3.1用例图

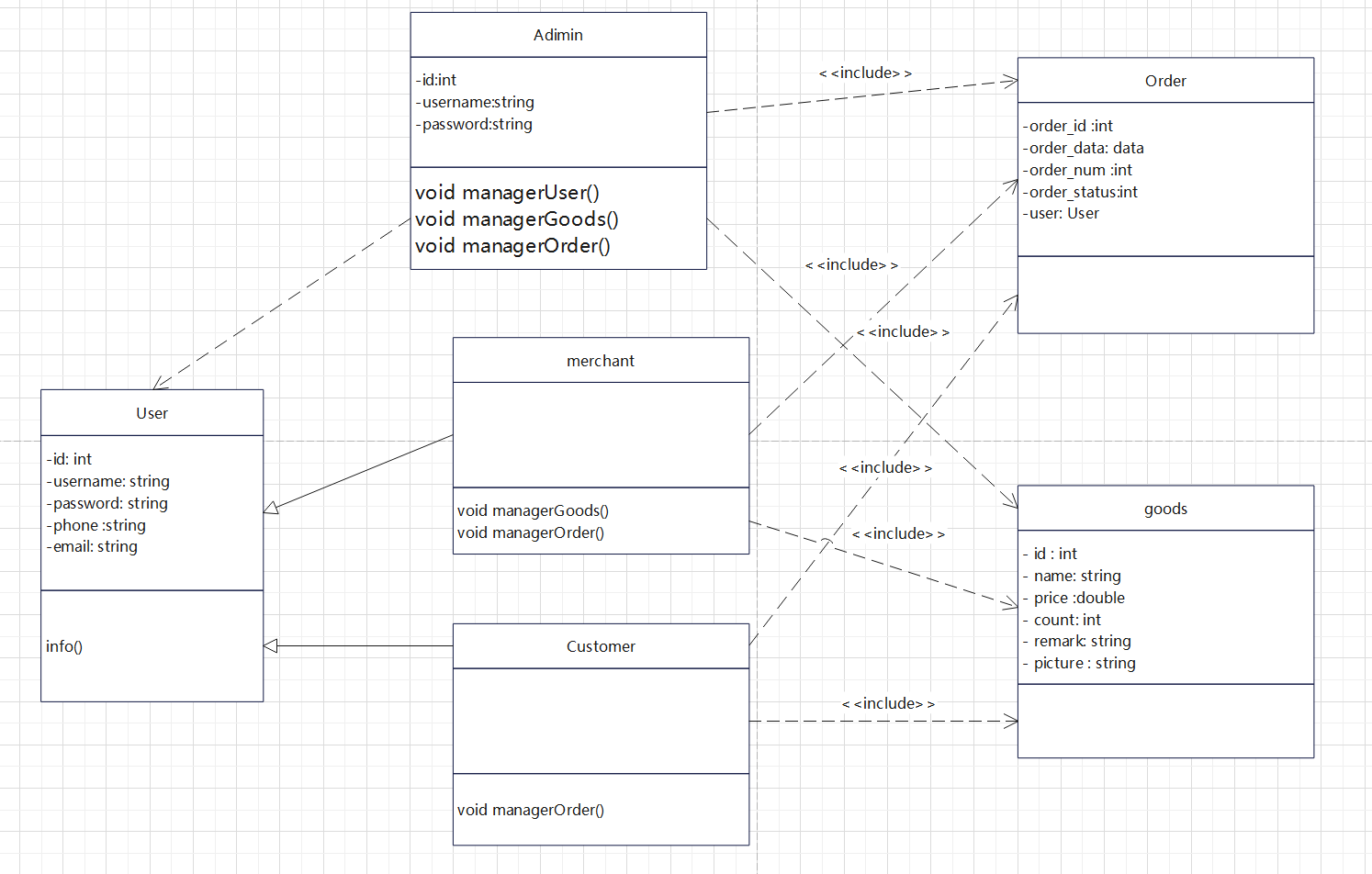


3.2实体图关系图



四、接口设计

4.1类图设计

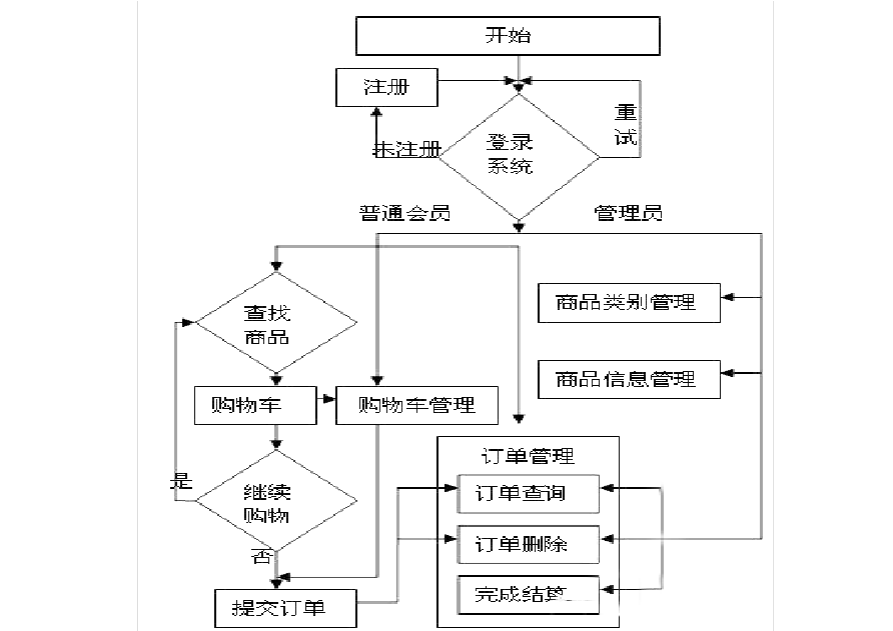


4.2数据库结构设计

我们的数据库采用的是关系型数据库mysql，将我们的数据分别存放进不同的表中，同时我们的数据库由于技术力的不足，我们无法通过采用关系型数据库和分布式数据库结合使用，这也和我们最初的设计里面有一些冲突出现

五、详细设计

5.1程序流程图



六、实现

6.1测试要点

# 6.1.1功能测试

功能测试，是对产品的各功能进行验证。测试要点有：

1. 链接测试  
   （1）所有的链接是否按指示链接到该链接的页面  
   （2）所链接的页面是否存在  
   （3）保证没有孤立页面
2. 表单测试  
   （1）验证服务器能正确保存这些数据  
   （2）后台运行的程序能正确解释和使用这些信息
3. Cookies测试  
   （1）Cookies是否起作用  
   （2）Cookies是否按预定的时间进行保存  
   （3）刷新对Cookies有什么影响
4. 设计语言测试  
   （1）不同版本的设计语言会引起客户端、服务端严重的问题
5. 数据库测试  
   （1）数据一致性错误：提交表单信息不正确  
   （2）输出错误
6. 相关功能性测试  
   （1）删除/增加某一项：是否对其他项造成影响，这些影响是否都正确  
   （2）列表默认值检查  
   （3）检查按钮功能是否正确：新建、编辑、删除、关闭、返回、保存、导入、上一页、下一页、页面跳转、重置（常见错误）  
   （4）字符串长度检查：超出长度  
   （5）字符类型检查  
   （6）标点符号检查：空格、各种引号、Enter键  
   （7）特殊字符：常见%、“、”  
   （8）中文字符：是否乱码  
   （9）检查信息完整：查看信息，查看所填信息是否完整更新；更新信息，更新信息与添加信息是否一致  
   （10）信息重复：需唯一信息处，比如重复的名字或ID、重名是否区分大小写、加空格  
   （11）检查删除功能：不选择任何信息，按Delete，看如何处理；选择一个或多个进行删除；多页选、翻页选删除；删除是否有提示  
   （12）检查添加和修改是否一致：添加必填项，修改也该必填；添加为什么类型，修改也该什么类型  
   （13）检查修改重名：修改时把不能重名的项改为已存在的内容  
   （14）重复提交表单：一条已经成功提交的记录，返回后再提交  
   （15）检查多次使用返回键：返回到原来页面，重复多次  
   （16）搜索检查：存在或不存在内容，看搜索结果是否正确；多个搜索条件，同时输入合理和不合理条件；特殊字符  
   （17）输入信息的位置  
   （18）上传下载文件检查：功能是否实现，上传：上传文件是否能打开、格式要求、系统是否有解释信息、将不能上传的文件格式修改后缀为可上传的文件格式；下载：下载是否能打开、保存、格式要求  
   （19）必填项检查：必填项未填写；是否有提示，如加\*；对必填项提示返回后，焦点是否自动定位到必填项  
   （20）快捷键检查：是否支持快捷键Ctrl+C、Ctrl+V、backspace；对不允许做输入的字段（如：下拉选项），对快捷方式是否也做了限制  
   （21）Enter键检查：输入结束后按Enter键，系统如何处理  
   （22）刷新键检查：按浏览器刷新键如何处理  
   （23）回退键检查：按浏览器回退键如何处理  
   （24）空格检查：输入项输入一个或多个空格  
   （25）输入法半角全角检查：比如，浮点型，输入全角小数点“。”或“. ”，如4. 5；全角空格  
   （26）密码检查：输入加密方式的极限字符；密码尽可能长  
   （27）用户检查：不同种类管理员用户的不同权限，是否可以互相删除、管理、编辑；一般用户的权限；注销功能，老用户注销再注册，是否为新用户  
   （28）系统数据检查：数据随业务过程、状态的变化保持正确，不能因为某个过程出现垃圾数据，也不能因为某个过程而丢失数据。  
   （29）系统可恢复性检查：以各种方式把系统搞瘫，测试系统是否可以迅速恢复  
   （30）确认提示检查：系统更新、删除操作：是否有提示、取消操作；提示是否准确；事前、事后提示  
   （31）数据注入检查：对数据库注入，特殊字符，对SQL语句进行破坏  
   （32）时间日期检查：时间、日期、时间验证：日期范围是否符合实际业务；对于不符合实际业务的日期是否有限制  
   （33）多浏览器验证

# 6.1.2易用性测试（界面测试）

1. 整体界面测试  
   （1）给用户的整体感：舒适感；凭感觉能找到想要找的信息；设计风格是否一致
2. 控件测试  
   （2）各控件的功能
3. 多媒体测试  
   （1）图形要有明确的用途，图片、动画排列有序且目的明确  
   （2）图片按钮链接有效，并且链接的属性正确（比如是新建窗口打开、当前页面打开）  
   （3）背景图片应该与字体颜色和前景颜色相搭配  
   （4）检查图片的大小和质量：一般jpg、gif、png；不影响图片质量的情况下能使图片的大小减小到30kb以下  
   （5）gif动画是否设置了正确的循环模式，颜色是否正常  
   （6）Flash、Silverlight元素是否正常
4. 导航测试  
   （1）站点地图和导航条：位置是否合理；页面结构
5. 内容测试  
   （2）提供信息的正确性、准确性、相关性
6. 容器测试  
   （1）DIV  
   （2）表格：作为控件，设置是否正确；长宽是否足够。作为较早的网页布局方式，考虑浏览器窗口尺寸的变化；内容动态增加或删除对界面的影响

# 6.1.3兼容性测试

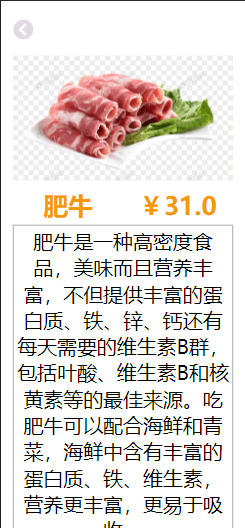
1. 平台测试：windows；unix；macintosh；linux
2. 浏览器测试：不同厂商的浏览器对Java、Javascript、ActiveX、plug-ins或不同的HTML的规格不同的支持；框架和层次结构在不同浏览器也不同的显示

# 6.1.4安全性测试

1. 安全性测试要求：  
   （1）能够对密码试探工具进行防范  
   （2）能够防范对Cookie攻击的常用手段  
   （3）敏感数据保证不用明文传输  
   （4）能防范通过文件名猜测和查看html文件内容获取重要信息  
   （5）能保证在网站收到工具后在给定时间内恢复，重要数据丢失不超过1小时

# 6.1.5性能测试

（1）压力测试：实际破坏一个Web应用系统，测试系统的反应，测试系统的限制和故障恢复能力  
（2）负载测试：在某一负载级别上的性能，包括某个时刻同时访问Web的用户数量、在线数据处理的数量  
（3）强度测试：测试对象在性能行为异常或极端条件下（如资源减少或用户过多）的可接受性，以此验证系统软硬件水平  
（4）数据库容量测试：通过存储过程往数据库表中插入一定数量的数据，看是否能及时显示  
（5）预期指标的性能测试：在需求分析和设计阶段会提出一些性能指标，对于预先确定的性能要求要首先进行测试  
（6）独立业务性能测试：对核心业务模块做用户并发测试，包括同一时刻进行完全一样的操作、同一时刻使用完全一样的功能  
（7）组合业务性能测试：模拟多用户的不同操作，最接近实际用户使用情况，按用户实际的实际使用人数比例来模拟各个模块的组合并发情况  
（8）疲劳强度性能测试：系统稳定运行情况下，以一定负载压力来长时间运行系统的测试  
6.2测试结果和总结



6.2.1测试功能完善

经测试功能后，基本功能大致完成

6.2.2

总结，在制作项目的时候，发现了很多问题，比如，老师可能还未讲到我们就必须完成，又或者老师讲了，但是在做的时候，从理论到实际做的时候会有很大的差距，甚至网上有时候找都找不到，最后就不得不去询问同学完成。同时在完成项目的时候我们也遇到了很多很多的bug，即使我们现在做的只是一个简单的项目，甚至有时候会出现在项目完成后，但是遇到逻辑上的bug，这也是最难修改的地方，我们有时候不得不进行全部推倒重来，总之，完成整个项目的时候我们收获良多，也提升了自己的能力