**基于MySQL数据库商品交易系统**

**李星辰**

**2019年5月20日**

**摘要**

随着各阶段贸易活动电子化的飞速发展, 网络在商务活动业务流程的优势已经越发明显。集中于基于互联网的电子交易，强调企业利用互联网与外部发生交易与合作。文章以MySQL数据库为基础, 以在线商品交易的数据库设计为切入点, 重点研究了网络在线商品交易系统数据库前端后端的构建以及数据库表的设计, 旨在设计出一个基本功能完备的在线商品交易数据库系统, 为交易系统各个功能模块的实现提供信息的存储和查询支撑, 该方法能够显著地提升商品交易系统的安全性以及处理信息的高效性。

**关键词：商品交易系统，MySQL数据库，应用分析**

**Abstract**

With the rapid development of electronic trade activities in various stages, the advantages of network in business processes have become more and more obvious. It focuses on internet-based electronic transactions and emphasizes that enterprises use the Internet to conduct transactions and cooperation with the outside world. Article on the basis of the MySQL database, an online commodity trading as the breakthrough point, the database design of the research is mainly focused on the online commodity trading system front-end backend database building and the design of database table, the basic function to design a complete database of online commodity trading system, trading system each function module to the realization of the information store and query support, this method can significantly improve the security of commodities trading system and the efficiency of processing information.

**Keywords:** Commodity trading system, MySQL database, Application analysis

**目录**

[1. 系统概述 4](#_Toc3017_WPSOffice_Level1)

[2. 技术方案 4](#_Toc19660_WPSOffice_Level1)

[3. 系统设计 4](#_Toc28042_WPSOffice_Level1)

[4. 详细设计 5](#_Toc15347_WPSOffice_Level1)

[4.1. 模块 5](#_Toc28042_WPSOffice_Level2)

[4.1.1. 登录和注册 5](#_Toc19660_WPSOffice_Level3)

[4.1.2. 买家卖家选择 5](#_Toc1164_WPSOffice_Level2)

[4.1.3. 卖家商品处理 6](#_Toc28042_WPSOffice_Level3)

[4.1.4. 买家商品交易选择 7](#_Toc15347_WPSOffice_Level3)

[4.1.5. 聊天室 8](#_Toc25531_WPSOffice_Level3)

[4.2. 技术内容 8](#_Toc9249_WPSOffice_Level2)

[4.2.1. 关系数据模型 8](#_Toc1164_WPSOffice_Level3)

[4.2.2. 关系操作 8](#_Toc9744_WPSOffice_Level3)

[4.2.3. 数据库设计过程 8](#_Toc4174_WPSOffice_Level3)

[4.2.4. 数据库语句 9](#_Toc5410_WPSOffice_Level3)

[4.2.5. Websocket 技术 9](#_Toc27691_WPSOffice_Level3)

[4.3. 设计方法 9](#_Toc10238_WPSOffice_Level2)

[4.4. 数据库设计 10](#_Toc22170_WPSOffice_Level2)

[4.5. 交易逻辑设计 11](#_Toc27329_WPSOffice_Level2)

[4.6. 前端设计 11](#_Toc16295_WPSOffice_Level2)

[4.6.1. 登录和注册 11](#_Toc11694_WPSOffice_Level3)

[4.6.2. 买家卖家选择 12](#_Toc25148_WPSOffice_Level3)

[4.6.3. 卖家商品处理 12](#_Toc24687_WPSOffice_Level3)

[4.6.4. 买家商品交易选择 12](#_Toc5293_WPSOffice_Level3)

[4.6.5. 聊天室 12](#_Toc9249_WPSOffice_Level3)

[4.7. 后端设计 12](#_Toc8052_WPSOffice_Level2)

[4.7.1. 登录和注册 13](#_Toc1837_WPSOffice_Level3)

[4.7.2. 卖家商品处理 13](#_Toc10238_WPSOffice_Level3)

[4.7.3. 买家商品交易选择 14](#_Toc11555_WPSOffice_Level3)

[4.6.5. 聊天室 14](#_Toc22170_WPSOffice_Level3)

[5. 测试 14](#_Toc25531_WPSOffice_Level1)

[6. 结论 15](#_Toc1164_WPSOffice_Level1)

[7. 附录 15](#_Toc9744_WPSOffice_Level1)

[7.1. tradeDAO.java 15](#_Toc15694_WPSOffice_Level2)

[7.2. trade.java 17](#_Toc27836_WPSOffice_Level2)

[7.3. MyWebSocket.java 18](#_Toc18705_WPSOffice_Level2)

[7.4. ChatRoom.jsp 20](#_Toc9640_WPSOffice_Level2)

[7.5. check.jsp 23](#_Toc26317_WPSOffice_Level2)

[7.6. check2.jsp 25](#_Toc23440_WPSOffice_Level2)

[7.7. check3.jsp 26](#_Toc24036_WPSOffice_Level2)

[7.8. check4.jsp 28](#_Toc22909_WPSOffice_Level2)

[7.9. check5.jsp 29](#_Toc12707_WPSOffice_Level2)

[7.10. login.jsp 30](#_Toc10023_WPSOffice_Level2)

[7.11. login\_fail.jsp 31](#_Toc20859_WPSOffice_Level2)

[7.12. register.jsp 32](#_Toc12179_WPSOffice_Level2)

[7.13. tradeList.jsp 34](#_Toc16638_WPSOffice_Level2)

[7.14. tradeList2.jsp 38](#_Toc210_WPSOffice_Level2)

[7.15. transaction.jsp 41](#_Toc16268_WPSOffice_Level2)

[7.16. update.jsp 42](#_Toc14703_WPSOffice_Level2)

[7.17. update2.jsp 43](#_Toc22398_WPSOffice_Level2)

1. **系统概述**

系统的开发环境为：客户端操作系统平台为Mac OS X 平台。服务器端的操作系统为Mac OS X server。编程语言采用Java. jsp。数据库使用MySQL。

系统的运行环境为：使用Mac OS X 等操作系统；Safari 浏览器；Internet环境。

为实现卖家和买家对交易的自主掌控，商品交易系统要存储并处理大量的数据。所以，该系统以MySQL数据库进行挂接，智能罗列卖家设置的商品信息，按商品的价格和数量构建相应的信息表，为买家提供商品介绍展示，以此实现商品的自主化选择和交易。

1. **技术方案**

系统在开发之前要做好数据库的设计，主要包括数据的内容和数据的组织方式。数据库的设计目标是数据库设计的重要环节。科学合理的数据库，要避免冗余数据，对某个记录的定位要准确迅速，且维护方便。对数据库的设计,组织数据库尤为重要,主要指在一个活多个表中存储数据并建立关系。

1. **系统设计**

系统采用模块化设计, 方便进行日后的扩展和改进。系统可以实现在线商品的浏览和交易, 具体包括用户注册和登录、卖家和买家身份选择、买家商品选择、卖家商品信息的插入删除和更改；在商品交易过程中, 买家可以选择利用平台浏览挑选商品并实现交易，卖家可以对商品信息进行的维护与管理。交易流程如下。



**图1**

1. **详细设计**

**4.1. 模块**

详细模块包括四个，分别是注册和登录模块，买家卖家选择模块，卖家商品的插入、删除和更改模块，买家商品选择模块。

**4.1.1. 登录和注册**

登陆界面包括“用户名”和“密码”的输入框，包括“登陆”和“重置”两个功能。输入完用户名和密码之后点击登陆则进入该用户的账户界面；如果输入错误则可以点击重置按钮，清空两个输入框。最后，有注册提示，如果是首次使用的用户则需要先注册账号，根据点击超链接“这里”进行注册操作。



**图2**

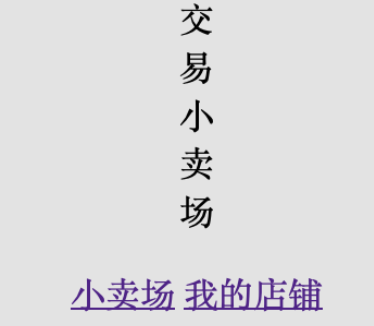
进入注册界面后可以看到三个输入框，分别是：用户名、输入密码、再次确认密码。功能按键也包括两个：注册、重置。重置的功能与上面相同。填好三个输入框之后点击注册，则已经有对应用户信息被保存，可以在登录界面直接登录。



**图3**

**4.1.2. 买家卖家选择**

在注册好并登录上账户之后，进入商品交易的账户界面。界面标题为“交易小卖场”，包括卖家界面和买家界面，分为两个超链接。一个是，“小卖场”，相当于买家界面，用户可以在公开的商品信息平台选择自己想购买的商品；另一个“我的店铺”，相当于卖家界面，用户可以自己根据自身的商品信息调整数据表，更改在公开平台上要卖出的商品。



**图4**

**4.1.3. 卖家商品处理**

卖家商品处理包括商品信息的插入、更新和删除。进入“我的店铺”之后显示出商品信息表，包括商品名称、商品数量、商品价格三个信息，对应终端数据表。有三个对表操作按钮，包括添加、更新、删除。

添加，即在数据表中插入商品的名称、数量和价格三个信息；更新，即更改已有商品信息；删除，即删掉数据表中的商品的三条信息。



**图5**

点击“添加”按钮，出现三条输入框用来输入商品三条信息，输入完成后点击“添加”则插入成功，“重置”按钮功能同上，“关闭”按钮用来隐藏三条输入框。



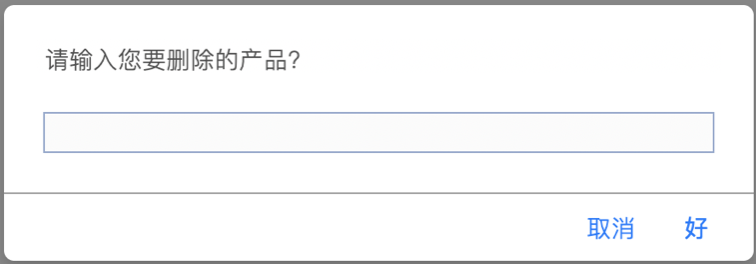
**图6**

点击“更新”按钮，用来更新已有信息，三条输入框用来输入该商品的新信息，输入完之后点击更新则更新成功。“重置”和“关闭”的功能同上。



**图7**

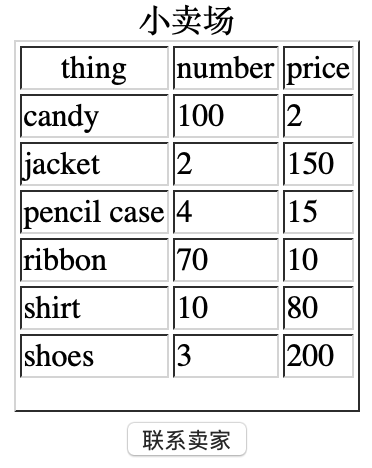
点击“删除”，弹出输入框，输入要删除的商品名称，点击“好”，则删除成功。



**图8**

**4.1.4. 买家商品交易选择**

买家界面也会出现商品信息表，还有商品图片对应显示，用户了解公开平台上所有商品的种类、数量、价格和实物图片。界面最后按键“联系卖家”，可以进入聊天室跟卖家交流。

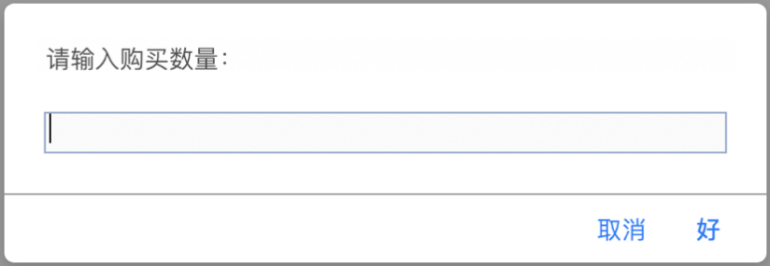


**图9**



**图10**

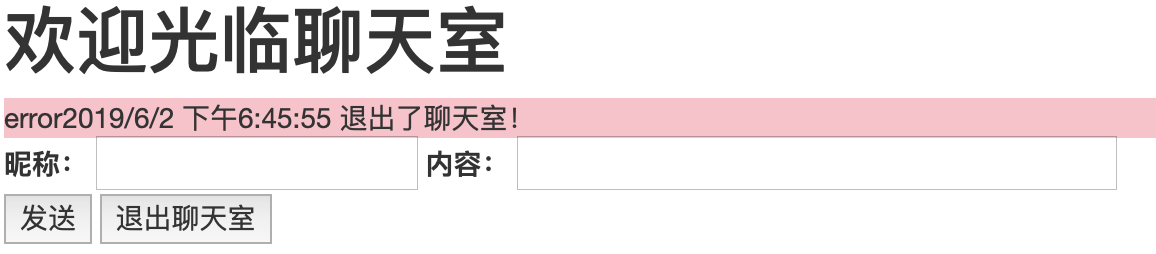
用户可以点击商品的实物图片，根据提示，在输入框里输入想要购买的数量，点击“好”，就能实现交易，此时上表中的对应商品的数量会减少，减少的数量就是输入的购买商品的数目。



**图11**

**4.1.5. 聊天室**

聊天室是买家联系卖家的交流平台。在聊天室界面里，用户可以自行定义昵称和发送内容。点击发送按钮，内容会输出，点击退出聊天室按钮，关闭连接。



**图12**

**4.2. 技术内容**

**4.2.1. 关系数据模型**

本设计使用SQL软件，建立关系数据模型。

在关系数据库系统中，关系是关系数据模型的核心。关系数据库系统中最主要的数据结构就是关系。在实际表示中，关系可以用一个表来直观的表示，通常表是以一种矩形数据行/列的形式表示。表的每一列表示关系的一个属性，每列的名字即为一个属性名，每一行表示一个记录代表一个物理实体。关系数据库中，所有的数据都是通过表来进行存储的，可以说没有表数据就无法进行存储和表示。

我们可以看出模型的主键和候选键。其中，主键：独立的标识存储在表中的每个记录或潜在记录的一组列（一组中可能只包含一个值）；候选键：包含独立的标识数据库中每一行的值的任意一个列或一组列。

**4.2.2. 关系操作**

选择、投影、连接三种关系操作。

选择：选择操作从表中返回特定行的集合，要指定限制条件，可以将条件写在WHERE子句中；

投影：限制了查询返回的列，允许列出所要查看的列，在SELECT列表中列出需要查看的列的名称；

连接：结合多个表的关系操作，为了查询特定数据而将两个或更多个表连接起来形成新的报表或视图，以便能够比较和对比数据库不同表中的数据，通过在WHERE子句中设定特定的条件，或者使用JOIN关键字通过表之间相关的列来连接不同表中的数据，在其中设定想指定的条件。

**4.2.3. 数据库设计过程**

(1) 信息的收集

(2) 确定数据

(3) 建立实体-关系模型（用来说明数据库中实体的等级和属性）

(4) 进行规范化

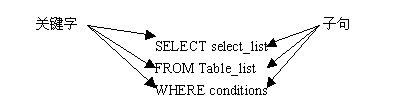
(5) 编写组建数据库的SQL代码，同时使用查询等操作进行一些检测

(6) 根据测试结果对已完成的数据库进行检查，在觉得有问题或需要完善的地方进行必要修改和完善。

**4.2.4. 数据库语句**

SQL是一门格式自由的语言，既不要求在一行放置的单词数，也不要求断行的地方，同样也不区分大小写，还包含了括号的使用。

每条SQL语句均由一个关键字开头，如SELECT、CREATE、INSERT、UPDATE等都描述了SQL语句将要执行的动作。每个关键字后都跟了一个子句，子句可以指定语句产生作用的数据，也可以提供语句作用的更详细的情况。子句可以包含表名、字段名，还可以包含表达式、常量及其它的一些关键字（如AND、NOT、OR等）。



**图13**

**4.2.5. Websocket 技术**

传统的HTTP协议是无状态的，每次请求（request）都要由客户端（如浏览器）主动发起，服务端进行处理后返回response结果，而服务端很难主动向客户端发送数据；这种客户端是主动方，服务端是被动方的传统Web模式对于信息变化不频繁的Web应用来说造成的麻烦较小，而对于涉及实时信息的Web应用却带来了很大的不便，如带有即时通信、实时数据、订阅推送等功能的应用。

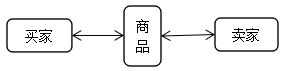
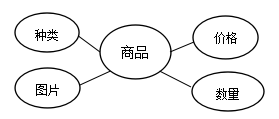
WebSocket协议是基于TCP的一种新的网络协议。是长连接，实现双向通信，具有底层socket的特点，实现真正意义上的推送功能；也是HTML5的技术之一，具有巨大的应用前景；并且节约带宽，节省服务器资源。但是少部分浏览器不支持，浏览器支持的程度与方式也有区别。

在本商品交易系统中的在线聊天室用的就是WebSocket技术。在实现websocket连线过程中，需要通过浏览器发出websocket连线请求，然后服务器发出回应，这个过程通常称为“握手” (handshaking)。在 WebSocket API，浏览器和服务器只需要做一个握手的动作，然后浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

**4.3. 设计方法**

概念设计、逻辑设计、物理设计三种设计方法。由上到下，先要概念设计，接着逻辑设计，再是物理设计，一级一级设计。

1. 概念设计：对用户要求描述的一个小卖场，设计注册、登记操作，进入店铺浏览、购买商品，通过对其中商品的分类、聚集和概括，交易双方记录，建立抽象的概念数据模型。建立E-R模型，如下图：



**图14** **商品实体集的E-R图** **图15**

（2）逻辑设计：主要工作是将小卖场的概念数据模型设计成数据库的一种逻辑模式，即适应于某种特定数据库管理系统所支持的逻辑数据模式。

买家、卖家进行交易，将图3商品实体集的E-R图转化为如下关系模式：

商品：种类、价格、数量。

商品实体集转化为数据库模式如下：

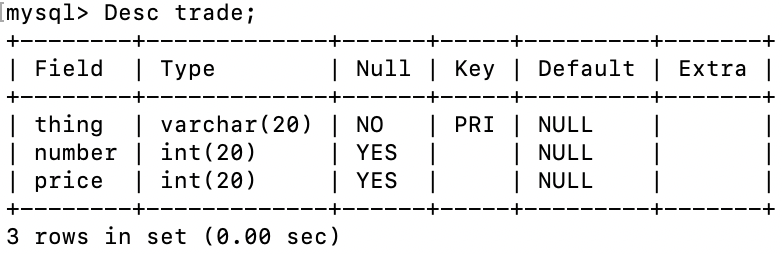
|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 字段值约束 |
| 商品种类 | nvarchar | 8 | NOT NULL |
| 商品价格 | money | 4 | NOT NULL |
| 商品数量 | char | 8 | NULL |

**表1**

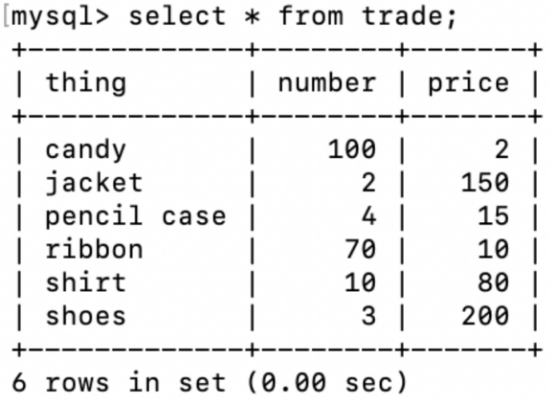
（3）物理设计：根据特定数据库管理系统所提供的多种存储结构和存取方法等依赖于具体计算机结构的各项物理设计措施，对具体的应用任务选定最合适的物理存储结构(包括文件类型、索引结构和数据的存放次序与位逻辑等)、存取方法和存取路径等。

**4.4. 数据库设计**

首先建立一个数据库，在该数据库内建两个数据表，分别是商品交易信息表和用户登录表。其中商品交易信息包括商品的名称、数量、价格。商品名称是主键，类型为varchar，长度为20，不为空；商品数量的类型为int，长度为20；商品价格的类型为int，长度为20。

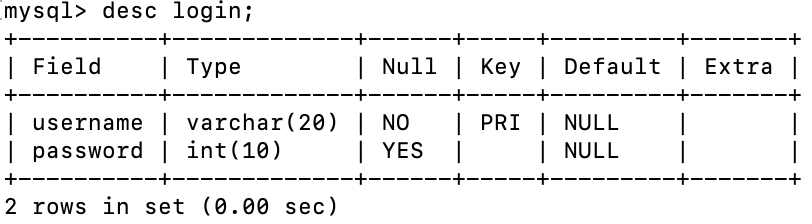


**图16**

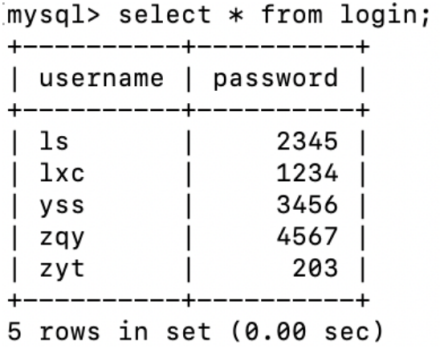


**图17：商品交易信息表**

用户登录信息表包括用户名和密码。用户名是主键，类型为varchar，长度为20，密码数据类型为int，长度为20。



**图18**



**图19：用户登录信息表**

**4.5. 交易逻辑设计**

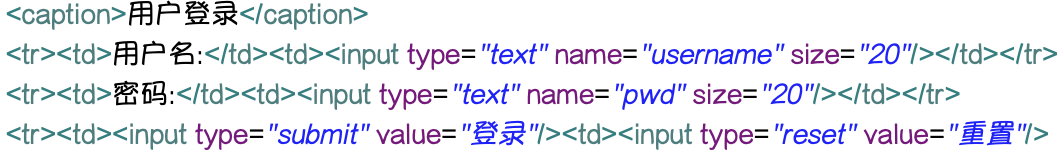
在交易过程中，如果卖家需要更新商品信息，在我的店铺中添加、更改或删除商品的数量和价格。如果买家需要购买商品，在买家界面也会出现商品信息表，用户可以在该界面浏览商品信息。此外，还有商品图片对应显示，方便用户了解商品详情。当用户产生购买意愿时，可以选择点击意愿商品的图片，输入购买数量即可实现交易。

**4.6. 前端设计**

前端主要包括主要模块中的几个界面的显示。

**4.6.1. 登录和注册**

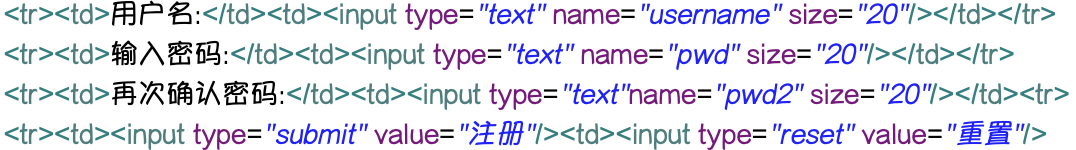
登录界面的显示如下：“text”规定了可以键入的输入框；size只的是用户名和密码的长度为20。要注意的是每个表中只能有一个“submit”。



“这里”规定了一个超链接，跳入注册的register.jsp文件，实现用户注册。



注册界面显示如下：与登录界面类似。在此界面下还规定了一个isValid函数，在输入框键入不符合要求时弹出提示：用户名不能为空、两次输入的密码不同、用户密码不能为空。



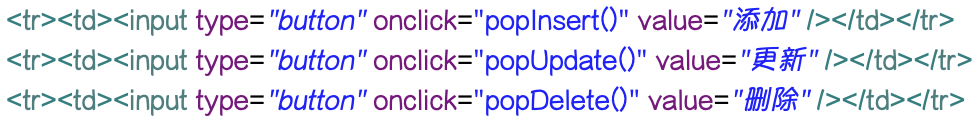
**4.6.2. 买家卖家选择**

这个界面主要是两个超链接选择卖家身份或者买家身份。跳转界面。

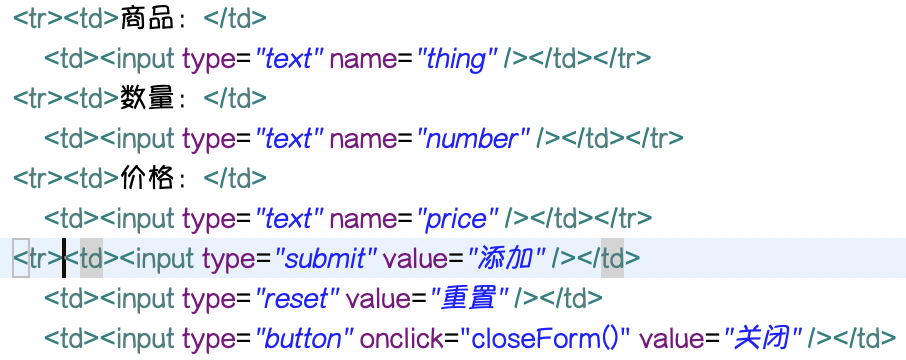


**4.6.3. 卖家商品处理**

卖家商品信息界面包括数据表trade，商品信息的处理包括插入、更新和删除，用按钮显示。

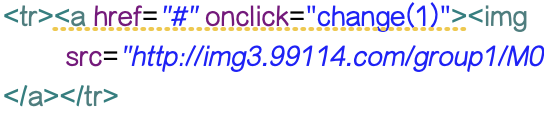


插入和更新按钮的响应是对应的表格，需要输入三个商品信息，删除按钮的响应是弹出的输入框，出入删除商品的名称。



**4.6.4. 买家商品交易选择**

在小卖场中显示内容有终端商品信息数据表和对应商品图片。数据表连接方式同上。图片用以下方式把网页路径存下来。点击图片会有响应，弹出输入框，并提示输入购买商品数目。



下面还有一个按键“联系卖家”，可以实现进入商家聊天室。

**4.6.5. 聊天室**

在聊天室中显示的内容有标题，昵称和内容（可以自行输入），还有两个按键“发送”和“退出聊天室”。

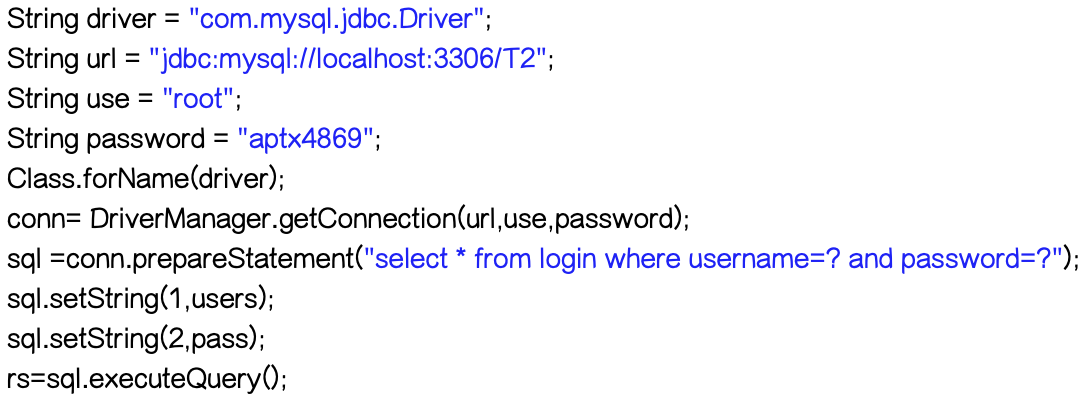


**4.7. 后端设计**

后端主要包括主要模块中几个界面中的数据表和运算。

**4.7.1. 登录和注册**

登录界面登录成功之后跳转到终端的数据表，显示商品信息。用以下语句建立终端的表格与工程文件的联系。如果没有正确登录则跳转到登录失败界面；如果可以正确登录，则跳转到交易界面。



在此介绍在jsp页面中通过表格显示数据库的实现，建立终端联系的java函数。首先，我们在web项目的src文件夹写入链接数据库的相关的类，他们分别是：链接数据库的类，数据的实体类。数据库实体类，在此类中主要含有数据库的字段，包名为cn.edu.bzu.entity，实体类名为trade.java，在类中主要规定数据表中的三个表头。在JAVA Web中，实体类里面的某一个类，相当于是数据库里的一张表，一个类里面的某个字段相当于表的列名。

数据库操作类：下面以对数据库的查询功能书写代码，包名cn. edu. bzu. dao，类名为tradeDAO.java。创建方法public List readFirstTitle()，选择返回值是List是因为一个标题含有名称、数量、价格三个商品信息，这些信息存在集合中，便于管理操作；创建变量，链接数据库，在这个web project使用的是JDBC技术链接的数据库。

与数据库建立链接分为四步：1.注册数据库驱动；2.获取数据库连接；3.获取statement对象；4.关闭资源。

注册界面用同样的方法与终端数据库建立联系，并在另一个数据表login中插入用户的注册信息。



**4.7.2. 卖家商品处理**

在添加信息输入框表格里键入商品的三个信息之后，点击添加。用以下语句实现在数据表中插入信息。



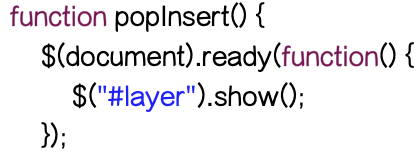
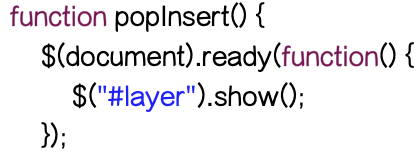
在更新信息输入框表格里键入商品的三个信息之后，点击更新。用以下语句实现在数据表中更改信息。

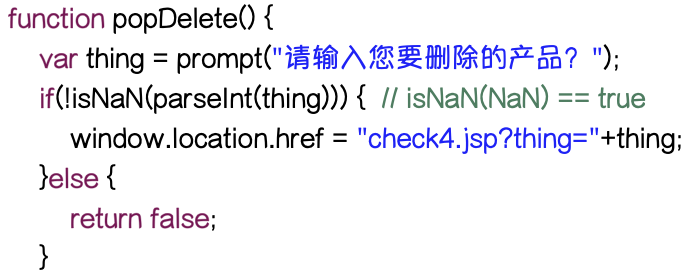


同理，在删除信息弹出输入框里键入商品的名称之后，点击确定。用以下语句实现在数据表中删除信息。



在完成插入、更新和删除操作之后，用以下函数进行数据表的更新。





**4.7.3. 买家商品交易选择**

在弹出框里输入购买数量之后，用下图显示的方式更新数据，跳转到更新界面，并把传入商品减少的数目传入update2.jsp。



在update2.jsp里利用下面语句完成数量计算，即减少商品数量（输入购买商品的数量），并实现数据表更新。



**4.6.5. 聊天室**

HTML5推出的WebSocket，真正实现了Web的实时通信，使B/S模式具备了C/S模式的实时通信能力。WebSocket的工作流程是这样的：浏览器通过JavaScript向服务端发出建立WebSocket连接的请求，在WebSocket连接建立成功后，客户端和服务端就可以通过TCP连接传输数据。因为WebSocket连接本质上是TCP连接，不需要每次传输都带上重复的头部数据，所以它的数据传输量比轮询和Comet技术小了很多。聊天室功能就是WebSocket在Java Web中的实现。

**5. 测试**

商品交易系统功能测试就是通过点击相应功能、输入相关数据和条件进行测试,。测试数据库的数据维护功能通过输入数据, 测试能否正常保存, 即对实现的功能分别做测试。我们主要通过测试来描述功能是否正常运行, 发现系统功能中存在的不能处理的情况, 发现系统错误, 是通过对设计功能情况进行测试, 包括功能点、测试结果和结论等。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 测试步骤 | 测试结果 |
| 1 | 注册用户 | 测试能否正常添加新用户 | 能正常添加用户 |
| 2 | 登陆用户 | 测试以注册用户能否进入小卖场 | 以注册用户能进入小卖场 |

**表2：登陆和注册模块功能用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 测试步骤 | 测试结果 |
| 1 | 小卖场连接 | 测试是否能成功进入买家界面 | 能成功进入买家界面 |
| 2 | 我的店铺连接 | 测试是否能成功进入卖家界面 | 能成功进入卖家界面 |

**表3：买家卖家选择模块功能用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 测试步骤 | 测试结果 |
| 1 | 商品添加 | 测试是否能成功添加商品 | 能成功添加商品 |
| 2 | 商品更新 | 测试是否能成功更新商品 | 能成功更新商品 |
| 3 | 商品删除 | 测试是否能成功删除商品 | 能成功删除商品 |

**表4：卖家商品处理模块功能用例**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 测试项目 | 测试步骤 | 测试结果 |
| 1 | 图片响应 | 测试图片是否能成功响应 | 图片能成功响应 |
| 2 | 交易实现 | 测试是否能成功实现交易 | 能成功实现交易 |

**表5：买家商品交易选择模块功能用例**

MySQL应用于商品交易系统的优势主要分为以下几方面, 一是免费开源优势, 如果再使用Mac OS操作系统, 可以减少购买操作系统和数据库的开销。二是多种平台支持的优势, MySQL可以与多个平台进行有效的连接, 实现信息资源的共享。三是中小型数据库灵活稳定的优势, 在设计MySQL程序的时候, 加入了其中没有的一些补充条件, 更加的适用于在中小型数据库中使用。商品交易系统通常要保存用户信息、商品信息和交易信息, 数据并不是很庞大, 因此,使用MySQL来管理数据非常合适，也最终能实现目的效果。

**6. 结论**

MySQL数据库是一种开放源代码的关系型数据库系统, 具有高可靠性、高适应性和高速度的特点。基于MySQL数据库的商品交易系统运行以来, 整体比较稳定, 流程清晰明了, 检索查询方便快捷, 通过多次测试, 实现了基本的商品浏览和买卖的功能。