校园二手交易系统

**软件设计说明书**

**版本：1.0**

编写： 十人小组

**2019年7月**

|  |  |  |
| --- | --- | --- |
| 文档版本变更 | | |
| 更改人 | 变更内容 | 审批人 |
| 丁严威 | 设计说明 | 陈红峰 |
| 李成蹊 | 修改3.9用户申请 | 陈红峰 |
| 何龙翔 | 修改3.7留言功能 | 陈红峰 |
| 薛金龙 | 修改3.11 3.12 | 陈红峰 |

目录

[软件计划说明书 1](#_Toc1226)

[1引言 4](#_Toc1056)

[1.1标识 4](#_Toc29900)

[1.2系统概述 4](#_Toc23151)

[1.3文档概述 4](#_Toc14205)

[1.4编写目的 4](#_Toc6049)

[1.5参考资料 5](#_Toc28095)

[2总体设计 5](#_Toc12917)

[2.1需求概述： 5](#_Toc20244)

[2.2软件结构 7](#_Toc14624)

[2.2.1主要功能 7](#_Toc31411)

[2.2.2性能要求 7](#_Toc14475)

[2.2.3输入要求 7](#_Toc12860)

[2.2.4输出要求 7](#_Toc15020)

[2.2.5安全与保密性要求 8](#_Toc21317)

[3程序描述 8](#_Toc31853)

[3.1登陆注册 8](#_Toc19986)

[3.1.1功能 8](#_Toc16377)

[3.1.2性能 8](#_Toc2864)

[3.1.4输出项目 9](#_Toc1655)

[3.1.5流程图 9](#_Toc32464)

[3.1.6界面设计 10](#_Toc5227)

[3.17接口设计 10](#_Toc24596)

[3.18算法设计 10](#_Toc32348)

[3.2浏览模块 10](#_Toc27564)

[3.2.1功能 10](#_Toc10097)

[3.2.2性能 12](#_Toc5238)

[3.2.3输入项目 12](#_Toc10779)

[3.2.4输出项目 12](#_Toc23037)

[3.2.5流程图 13](#_Toc8980)

[3.2.6界面设计 13](#_Toc31752)

[3.2.7接口设计 13](#_Toc22831)

[3.2.8算法 13](#_Toc15264)

[3.3查看历史记录功能 14](#_Toc20688)

[3.3.1 功能 14](#_Toc7467)

[3.3.2 性能 14](#_Toc4102)

[3.3.3输入项目 14](#_Toc31898)

[3.3.4输出项目 14](#_Toc16546)

[3.3.5流程图 15](#_Toc2995)

[3.3.6界面设计 16](#_Toc1973)

[3.3.7接口设计 17](#_Toc21502)

[3.3.8算法设计 17](#_Toc26210)

[3.4个人信息功能 17](#_Toc24810)

[3.4.1 功能 17](#_Toc21994)

[3.4.2 性能 17](#_Toc7186)

[3.4.3输入项目 17](#_Toc8531)

[3.4.4输出项目 17](#_Toc29911)

[3.4.5流程图 18](#_Toc20770)

[3.4.6界面设计 20](#_Toc30125)

[3.4.7接口设计 20](#_Toc25081)

[3.4.8算法设计 20](#_Toc15552)

[3.5 用户发布产品和取消产品功能 20](#_Toc1308)

[3.5.1 功能 20](#_Toc15075)

[3.5.2 性能 20](#_Toc25130)

[3.5.3 输入项目 20](#_Toc29421)

[3.5.4 输出项目 21](#_Toc19341)

[3.5.5 流程图 21](#_Toc1252)

[3.5.6 界面设计 21](#_Toc9064)

[3.5.7 接口设计 22](#_Toc27567)

[3.5.8 算法设计 22](#_Toc23829)

[3.6订单 22](#_Toc18722)

[3.6.1功能 22](#_Toc10094)

[3.6.2性能 23](#_Toc15265)

[3.6.3输入项目 23](#_Toc28274)

[3.6.4输出项目 23](#_Toc20914)

[3.6.5流程图 23](#_Toc5897)

[3.6.6界面设计 25](#_Toc4910)

[3.6.7接口设计 25](#_Toc11764)

[3.6.8算法设计 25](#_Toc5365)

[3.7留言功能 26](#_Toc19241)

[3.7.1 功能 26](#_Toc29523)

[3.7.2 性能 27](#_Toc13616)

[3.7.3输入项目 27](#_Toc12407)

[3.7.4输出项目 27](#_Toc27704)

[3.7.5流程图 28](#_Toc8102)

[3.7.6界面设计 29](#_Toc17145)

[3.7.7接口设计 29](#_Toc2839)

[3.7.8算法设计 29](#_Toc7082)

[3.8评论功能 29](#_Toc29080)

[3.8.1功能 29](#_Toc20876)

[3.8.2 性能 29](#_Toc25193)

[3.8.3输入项目 29](#_Toc23510)

[3.8.4输出项目 29](#_Toc28919)

[3.8.5流程图 30](#_Toc23819)

[3.8.6界面设计 30](#_Toc18264)

[3.8.7接口设计 30](#_Toc10756)

[3.8.8算法设计 31](#_Toc21943)

[3.9用户申请成为卖家 31](#_Toc28205)

[3.9.1功能 31](#_Toc19984)

[3.9.2性能 31](#_Toc32455)

[3.9.3输入项目 32](#_Toc22257)

[3.9.4输出项目 32](#_Toc15667)

[3.9.5程序逻辑 32](#_Toc29979)

[3.9.6界面设计 33](#_Toc10381)

[3.9.7接口 33](#_Toc27436)

[3.9.8算法 33](#_Toc286)

[3.10 悬赏功能 33](#_Toc1993)

[3.10.1 功能 33](#_Toc13428)

[3.10.2 性能 33](#_Toc24028)

[3.10.3 输入项目 33](#_Toc16159)

[3.10.4 输出项目 34](#_Toc32726)

[3.10.5 流程图 35](#_Toc22879)

[3.10.6界面设计 36](#_Toc24104)

[3.10.7接口设计 36](#_Toc11855)

[3.10.8算法设计 36](#_Toc10823)

[3.11 审核功能 36](#_Toc6474)

[3.11.1 功能 36](#_Toc27376)

[3.11.2 性能 36](#_Toc928)

[3.11.3 输入项目 36](#_Toc5013)

[3.11.4 输出项目 37](#_Toc3290)

[3.11.5流程图 37](#_Toc16399)

[3.11.6界面设计 37](#_Toc14867)

[3.11.7接口设计 37](#_Toc32525)

[3.11.8算法设计 37](#_Toc3384)

[3.12管理员增删改查用户和产品的功能 38](#_Toc10461)

[3.12.1作为管理员可以对用户信息进行增删改查； 38](#_Toc11052)

[3.12.2作为管理员可以对被举报的违规产品进行删除 40](#_Toc28023)

[3.12.3作为管理员可以通知卖家产品下架 42](#_Toc12320)

[3.12.4作为管理员可以通知买家举报成功 44](#_Toc23544)

[3.13找回密码 46](#_Toc2980)

[3.13.1功能 46](#_Toc30454)

[3.13.2性能 46](#_Toc18997)

[3.13.3输入项目 46](#_Toc17477)

[3.13.4输出项目 46](#_Toc8285)

[3.13.5流程图 47](#_Toc31791)

[3.13.6界面设计 47](#_Toc27544)

[3.13.7接口设计 47](#_Toc23345)

[3.13.8算法设计 47](#_Toc16283)

[4数据字典 47](#_Toc29357)

[5数据库设计 49](#_Toc2744)

# 1引言

## 1.1标识

标识号：YH-ZQ-QP-122

名称：校园二手交易系统

缩略名：UstSystem

版本号：1.0.0

发布号：1.0.0.181220\_Alpha

## 1.2系统概述

对于用户而言，该系统是为了方便校园学生进行低价的有价值的物品转移。

校园二手交易系统可以更好的帮助用户买家查询到该系统的所有物品，并且可以浏览该物品的大概信息，以低廉的价格来获得对自己有用的物品。

校园二手交易系统可以更好的帮助用户卖家很好的出售自己无用的东西来补贴自己生活。

## 1.3文档概述

为了保证项目团队按时保质地完成项目目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，因此以文件化的形式，把对于在项目生命周期内的工作任务范围、各项工作的任务分解、项目团队组织结构、各团队成员的工作责任、团队内外沟通协作方式、开发进度、经费预算、项目内外环境条件、风险对策等内容做出的安排以书面的方式，作为项目团队成员以及项目干系人之间的共识与约定，项目生命周期内的所有项目活动的行动基础，项目团队开展和检查项目工作的依据。而且该文档还可以更好的对小组成员进行规范。

编写本计划书的目的是明确系统开发过程中对各项工作任务的项目目标，开发团推以及人员安排、工作分解，项目进度和开发费用估算等问题用文档的形式记录下来，以便项目可以顺利的开发。

## 1.4编写目的

编写详细设计说明书是软件开发过程必不可少的部分，其目的是为了使开发的人员在完成需求规格说明书的基础上完成概要设计规定的各个功能块的具体实现的设计工作。通过对所产生的功能模块进行过程描述，开发一个可以直接转化成程序语言的软件标识，这样可以使软件设计人员很好的了解软件的功能，并对软件的所有模块有一个更深的了解

## 1.5参考资料

[1]GJB438B-2009 军用软件开发文档通用要求

[2]Y .Daniel Liang著李娜译 ,JAVA语言程序设计．北京：机械工业出版社 2012

[3]刘先锋 ,数据库系统原理与应用.武汉：华中科技大学出版社2012

[4]谢希仁,计算机网络（第五版）．北京：电子工业出版社 2012

# 2总体设计

## 2.1需求概述：

(1).登陆，注册，找回功能

作为用户可以注册来登陆进入系统；

作为用户可以登陆到主页浏览产品；

作为用户可以通过邮箱来找回密码；

作为管理员可登陆到管理界面；

(2)浏览功能

作为用户可以在主页浏览产品来寻找想要产品；

作为用户可以通过关键字搜寻想要找的产品；

作为用户可以举报违规产品；

作为用户可以查看单独物品详细信息；

作为用户可以查看卖家所卖东西的记录；

作为用户可以查看东西的评价；

(3)历史记录功能

作为用户可以查看自己的卖东西的记录；

作为用户可以查看自己买的东西的记录；

作为用户可以确认订单；

(4)个人信息修改功能

作为用户可以修改自己的个人信息；

作为用户可以查看自己的个人信息；

(5)用户发布产品和取消产品功能

作为用户可以发布产品；

作为用户可以取消已发布产品；

作为用户可以修改自己发布的产品的信息；

作为用户可以查看自己产品的状态；

(6)购买产品功能

作为用户可以把产品加入购物车来继续浏览；

作为用户可以点击生成来生成订单；

作为用户可以点击来取消订单；

作为用户可以点击来确认订单；

(7)信息功能

作为用户可以作为买家对卖家进行商品相关的询问；  
作为用户可以作为卖家对买家留言回复；  
作为用户可以作为买家对卖家留言回复；

(8)评价功能

作为用户可以作为买家，再购买商品后对卖家的商品进行评价；  
作为用户可以对曾经购买过的商品的评价进行修改；

(9)用户申请成为卖家

作为用户可以向系统（管理员）提出申请成为卖家；

只有经过审批后具有卖家资格的用户才有权利发布想要出售的商品；

(10)悬赏功能

作为用户可以发布自己想要的二手货的信息；

系统通过关键词匹配是否有满足要求的二手货；

若有，则向其发送通知及信息；

若无，则向其发送无的通知，（该悬赏未解决，积压）；

在某商家发布新二手货时，若与积压的悬赏匹配，则向用户发送通知和信息。

(11)管理员审核用户能否成为卖家功能

作为管理员可以通过点击审核的按钮来审核用户；

作为管理员可以看到所有被审核过的用户

审核界面；

(12)管理员增删改查用户和产品功能

作为管理员可以对用户信息进行增删改查；

作为管理员可以搜索用户；

作为管理员可以对被举报的违规产品进行下架；

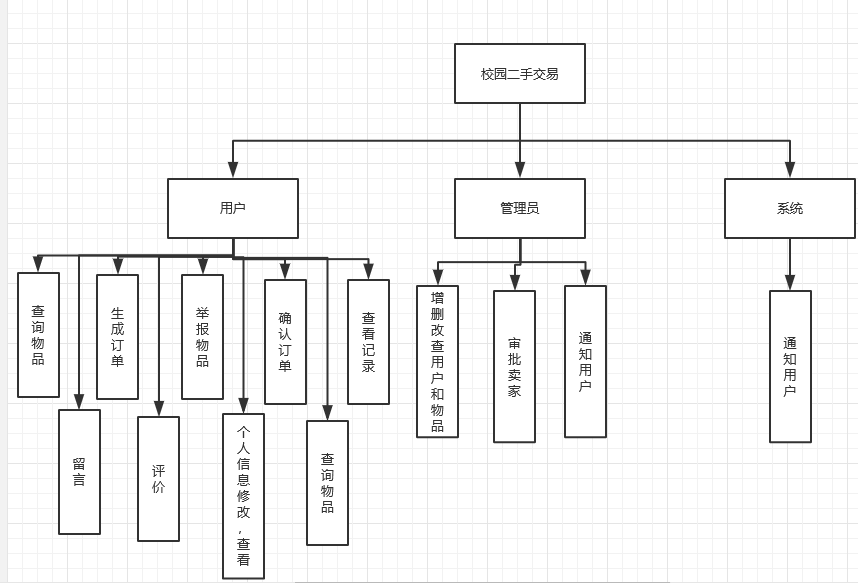
作为管理员可以对被举报的违规产品进行恢复；

作为管理员可以看到所有被举报的产品

作为管理员可以看到所有被下架的产品

作为管理员可以看到所有被恢复的产品

## 2.2软件结构



### 2.2.1主要功能

供用户进行买卖物品，供管理员高效管理用户和产品。

### 2.2.2性能要求

用户提供的信息即使反映在网站的页面上。信息存储在数据中，并且能被及时正确的刷新。

### 2.2.3输入要求

数据完整，详实。

### 2.2.4输出要求

简洁，快速，实时。

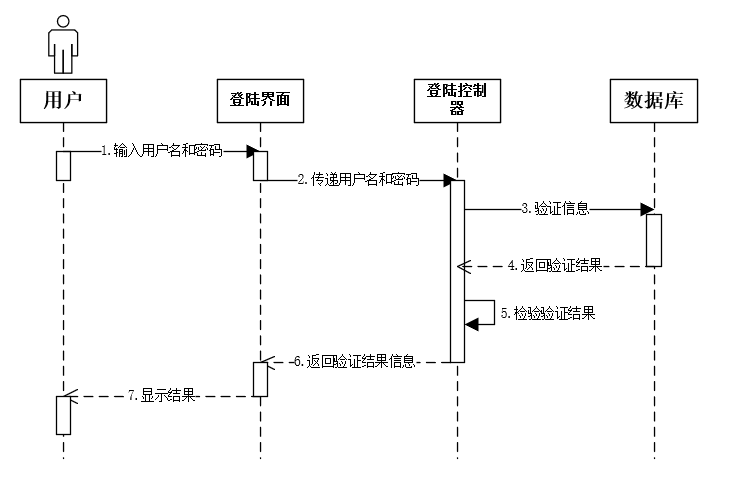
### 2.2.5安全与保密性要求

仅供内部使用。

# 3程序描述

## 3.1登陆注册

### 3.1.1功能



登陆可分为用户登陆和管理员登陆，在登陆模块的描述中，将管理员和用户都看作系统的用户，他们根据自己的账户和密码登陆系统，才可进行以后的操作。若用户名和密码输入错误，不匹配，系统将会提示输入错误，并要求用户重新输入。新用户输入用户名密码进行注册。

### 3.1.2性能

用户的密码传输对安全性能要求很高，数据传输不能采用明文。系统对账户和密码的匹配并返回信息的时间不超过2s。

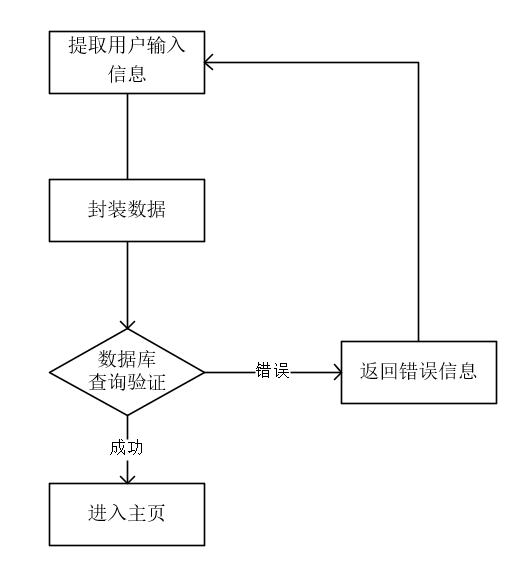
3.1.3输入项目

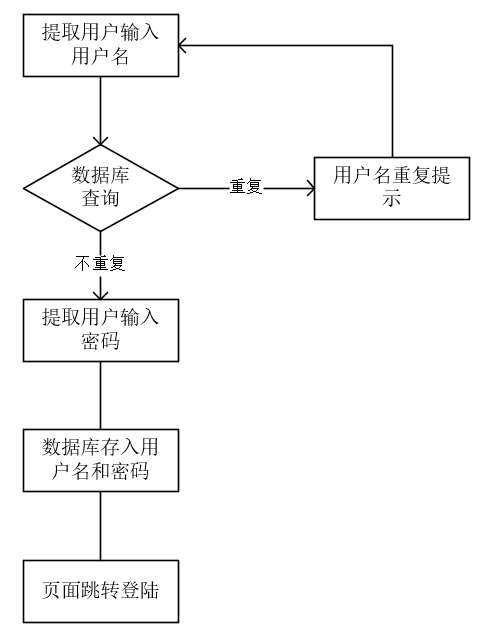
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 用户名 | Varchar |  |  |  | 主 |
| 密码 | Varchar |  |  |  |  |

### 3.1.4输出项目

根据数据库中存在的表中信息来查询，判断用户名、密码是否正确，正确则跳转到主页面，若用户名与密码不匹配则提示错误。

### 3.1.5流程图





### 3.1.6界面设计

从登陆界面可以点击注册进入注册界面。

从登陆界面可以点击忘记密码进入忘记密码的界面。

从注册界面可以返回登陆界面。

### 3.17接口设计

### 3.18算法设计

登陆时该模块通过读取前台信息，提取用户名及密码，将密码进行MD5加密，采用PreparedStatement生成SELECT语句解决SQL注入问题，到数据库查询信息，然后将结果返回给前台。

注册时该模块通过读取前台信息，提取用户名，采用PreparedStatement生成SELECT语句解决SQL注入问题，到数据库查询信息，然后将结果返回给前台，如果不重复再读取密码，将密码进行MD5加密，连同用户名存储进数据库

## 3.2浏览模块

### 3.2.1功能



浏览主页，查询商品



查看商品详细信息，举报商品



查看卖家详细信息

用户可以在主页浏览产品和寻找想要的产品；

可以通过关键字搜寻想要找的产品；

可以举报违规产品；

可以查看单独物品详细信息；

可以查看卖家所卖东西的记录；

可以查看东西的评价。

**3.2.2性能**

查询支持模糊查询；所有网页要求在5秒以内处理完成

**3.2.3输入项目**

所查询商品的关键信息

**3.2.4输出项目**

返回的商品关键信息

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 允许空 | 主/外键 |
| user\_id | int | 11 |  |  |
| goods\_name | varchar | 50 |  |  |
| description | Varchar | 255 |  |  |
| goods\_status | int | 11 |  |  |
| goods\_id | int | 11 |  | 主 |

### 3.2.5流程图



**3.2.6界面设计**

从首页在搜索框输入关键字，点击搜索，进入搜索页面

从首页或搜索页面，点击商品，进入商品详细信息页面

从商品详细信息页面，点击举报，进入举报页面

从首页或搜索页面或商品详细信息页面，点击卖家，进入卖家详细信息页面

**3.2.7接口设计**

**3.2.8算法**

## 3.3查看历史记录功能

### 3.3.1 功能

用户可以查看自己买东西的记录、用户可以查看自己卖东西的记录、用户可以查看卖家卖东西的记录。

### 3.3.2 性能

查询记录精确、单词查询任务要在3秒内完成，系统响应要在3秒以内。保证10台机器一起运行不会有高延迟。

### 3.3.3输入项目

账号、密码、点击相应的查看记录按钮。

### 3.3.4输出项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| GoodsName | String | 64 |  |  |  |
| GoodsID | int | 4 |  |  |  |
| Number | Int | 4 |  |  |  |
| Description | String | 64 |  |  |  |

### 3.3.5流程图



图1 用户查看卖家或卖家记录流程图



图2 用户查看卖家记录流程图

### 3.3.6界面设计

见流程图

### 3.3.7接口设计

### 3.3.8算法设计

顾客点击对应的按钮，到数据库中查询是否有买/卖记录。如果没有记录，则返回没有记录的体型。如果有，后端根据用户名返回对应的记录。

## 3.4个人信息功能

### 3.4.1 功能

用户可以查看、修改自己的个人信息

### 3.4.2 性能

查看个人信息精确、单词查看个人信息任务要在3秒内完成，系统响应要在3秒以内。保证10台机器一起运行不会有高延迟。修改同步到数据库，修改精确，单词修改任务要在3秒的完成，系统响应要在3秒以内，保证式十台机器一起运行不会有高延迟，保证修改不发生冲突。

### 3.4.3输入项目

账号、密码、点击修改/查看按钮。

### 3.4.4输出项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| User-name | String | 64 |  |  |  |
| User-ID | int | 4s |  |  |  |
| Phone | Int | 11 |  |  |  |
| E-mail | String | 64 |  |  |  |
| User-number | int | 10 |  |  |  |

### 3.4.5流程图



图3 用户查看修改个人信息流程图

### 3.4.6界面设计

见流程图

### 3.4.7接口设计

### 3.4.8算法设计

用户点击查看按钮，根据用户名到数据库中查看用户相关信息，由后台返回相应数据显示在前端。用户点击修改个人信息按钮，用户填入相应的数据再点击确认修改按钮，此时系统询问用户是否确认修改，如果点击确认，则修改同步至数据库，否则修改不生效。

## 3.5 用户发布产品和取消产品功能

### 3.5.1 功能

当用户去发布产品的时候，将提供给用户发布产品界面，并且在其中填写相关信息，优先度较低。用户可以在发布产品的时候和自己的发布产品界面取消自己发布的产品。

### 3.5.2 性能

软件响应时间不超过1s，系统对用户的请求响应时间不超过1s，用户获取系统的响应不超过1s，用户发布界面以后发布产品界面在1s内进行更新，显示用户最新更新的。

### 3.5.3 输入项目

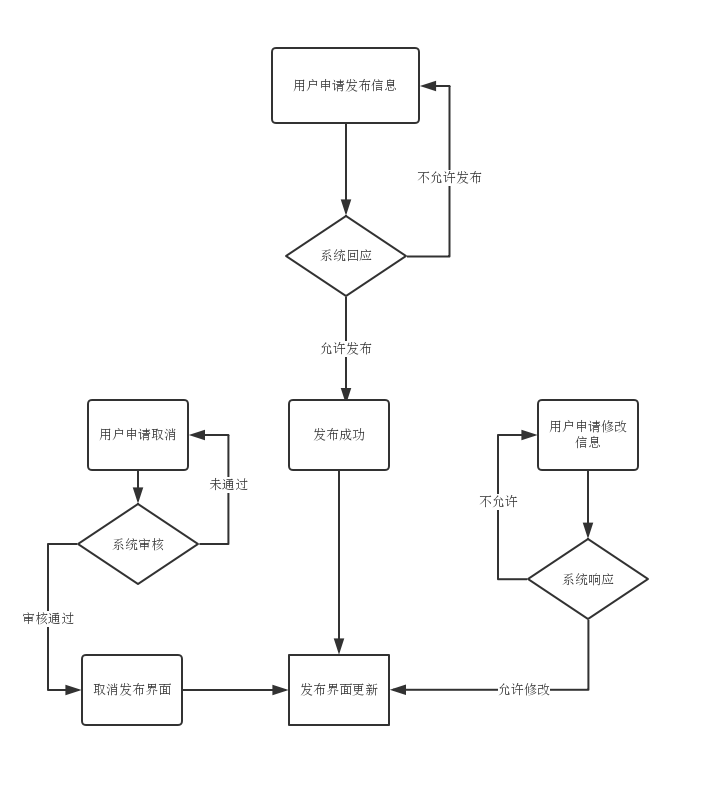
用户输入产品名字、描述、通过数据库的封装大盘以下表格中

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 产品名字 | varchar | 255 |  |  |  |
| 产品描述 | varchar | 255 |  |  |  |
| 产品价格 | double | 10 |  |  |  |

### 3.5.4 输出项目

根据数据库中存在表的信息来自动生成一个订单，提取用户输入的数据来生成订单，并且进行更新。若出现错误，系统自动弹出错误提示框。

### 3.5.5 流程图



### 3.5.6 界面设计

用户在发布界面点击发布按钮，系统生成后，跳转到主页并且显示刚才发布的产品信息。

用户在自己的发布界面可以修改自己发布产品的信息，修改后自动回到发布界面。

### 3.5.7 接口设计

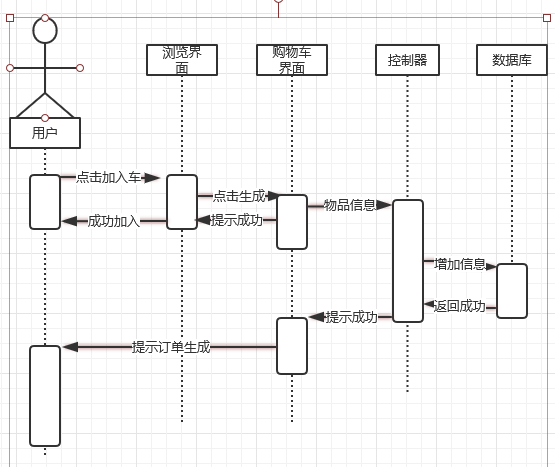
该模块通过publish接口中的update方法访问数据库，提供连接数据库的功能，在用户发布产品的时候会调用此模块。

### 3.5.8 算法设计

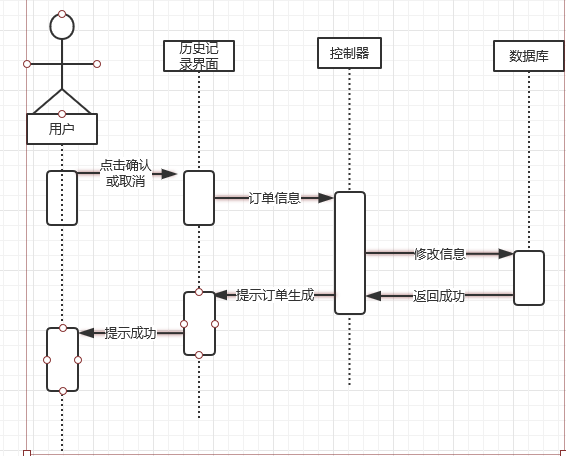
该模块通过访问数据库对数据库中的信息进行添加修改封装成UPDATE语句，对数据库中信息进行修改，将信息发挥到发布界面上。若用户输入错误，弹出错误提示框。

## 3.6订单

### 3.6.1功能



选择想要的物品加入购物车提示成功，并确认生成订单返回成功。



点击确定或取消就能把信息传给数据库，并提示。

### 3.6.2性能

在0.5秒之内加到购物车中，确认订单在1秒内生成数据到数据库。

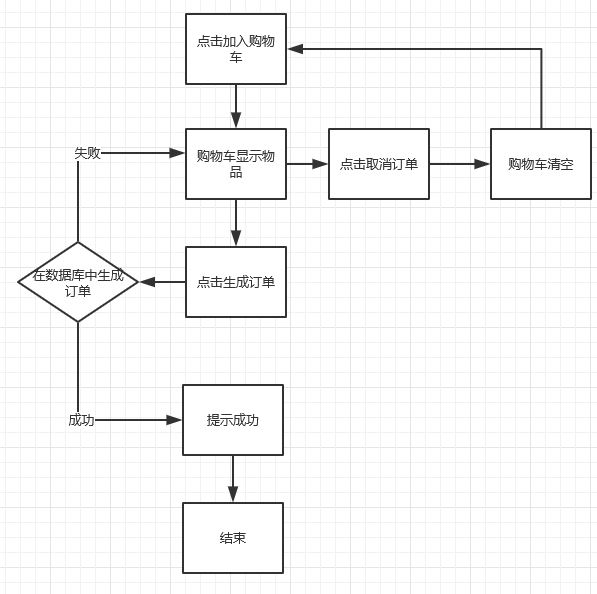
### 3.6.3输入项目

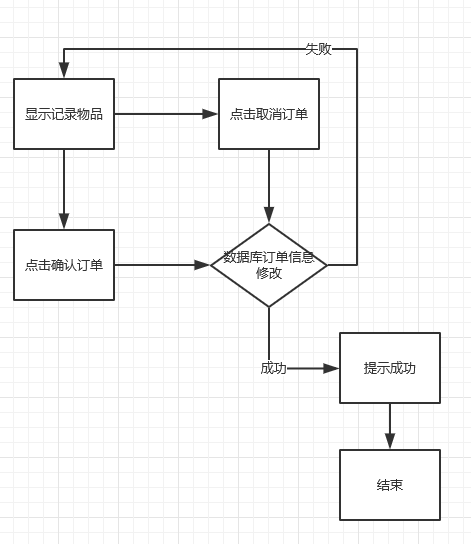
无

### 3.6.4输出项目

根据操作来提示成功与否。

### 3.6.5流程图





### 3.6.6界面设计

在浏览界面点击加入购物车购物车中刷新商品

在购物车界面中点击生成订单，提示成功与否

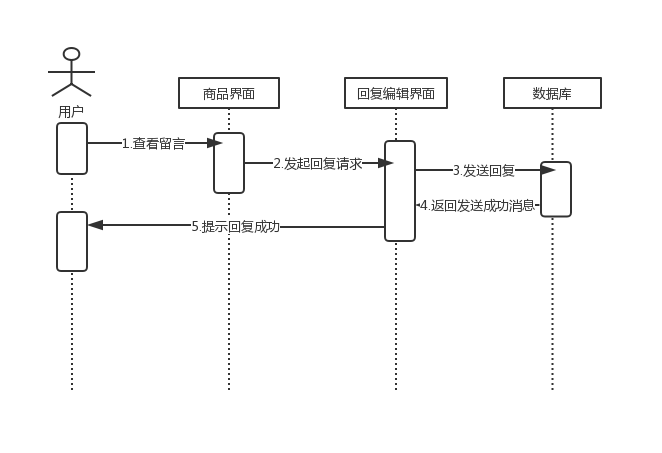
在历史记录点击确认或取消，提示成功与否

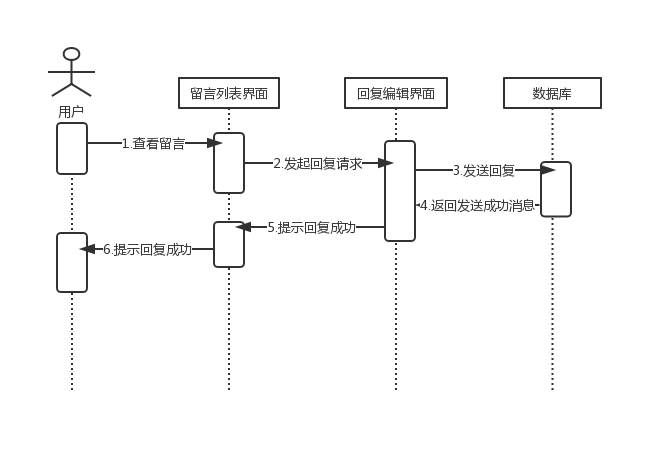
### 3.6.7接口设计

### 3.6.8算法设计

## 3.7留言功能

### 3.7.1 功能





顾客浏览商品时可以给相应的商家留言。用户可以在留言列表查看他人对自己的留言并对其进行回复。

### 3.7.2 性能

顾客留言可以精确到具体商家，用户回复可以精确到个人。

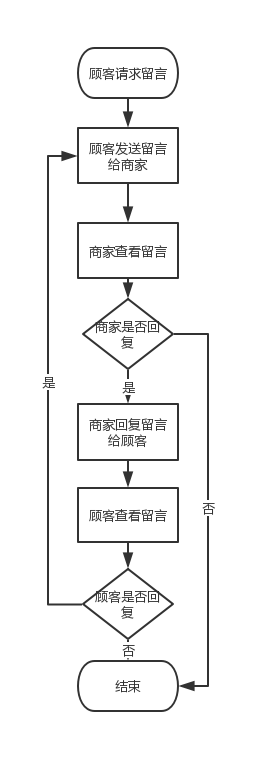
### 3.7.3输入项目

用户向留言对象留言的内容以及留言对象。

### 3.7.4输出项目

收到的留言内容以及发送方。

### 3.7.5流程图



### 3.7.6界面设计

留言：从商品界面点击“联系商家”跳转到编辑留言界面，编辑完成后点击发送，在提示发送成功后返回商品界面；

回复：从留言列表里点击具体的留言跳转相应留言的回复界面，编辑完成后点击发送，在提示发送成功后返回留言列表界面。

### 3.7.7接口设计

留言模板需要与数据库交互的接口，通过dao包中的HelongxiangDAO方法来实现与数据库的连接。

### 3.7.8算法设计

用户编辑完留言发送后，数据库在information表中插入新数据，存储发送方与接收方的id，留言内容，留言日期，是否已读。

在用户进入留言列表后，在数据库在information表中搜索接收人id为用户id的information，读取其中的信息，以发送方为分类在留言列表显示消息发送方，消息是否已读，最后一条消息的发送时间，是否有未读消息，以及回复按钮。

在用户进入回复界面时，数据库在information表搜索接受方为用户id且发送方为所指定的用户id，或者接受方为所指定的用户id且发送方为用户id的information，将搜素结果按id顺序依次显示对于的发送方，发送时间以及消息内容。

## 3.8评论功能

### 3.8.1功能

当买家购买商品后，可以进行评论。也可对以前购买的商品评论、

### 3.8.2 性能

买家的评论可以被商家看到，并且商家不能更改。

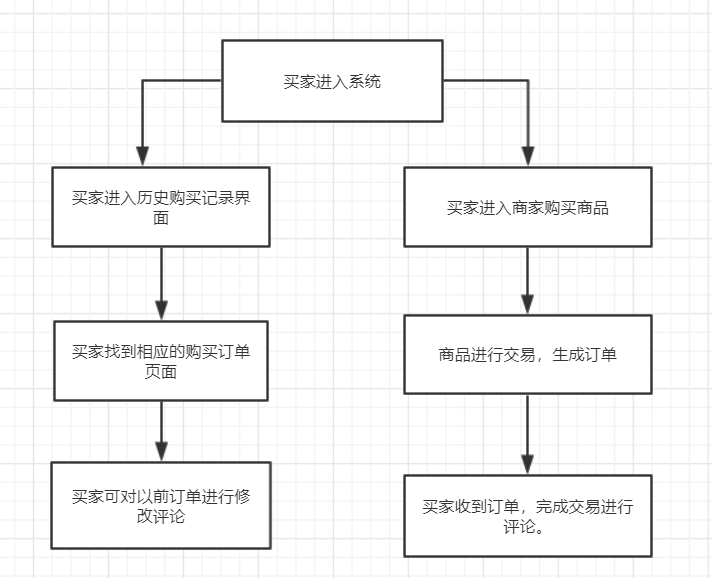
### 3.8.3输入项目

|  |  |  |  |
| --- | --- | --- | --- |
| 用户登入需要输入账号密码 | 用户输入想购买商品的信息 | 用户输入想要对此商品的评论信息 | 用户输入想要修改以前购买商品的评论信息 |

### 3.8.4输出项目

|  |  |  |  |
| --- | --- | --- | --- |
| 系统提示账号密码正确，用户进入系统 | 系统列举出用户需要的商品信息 | 系统提示用户评论成功 | 系统提示用户评论修改成功 |

### 3.8.5流程图



### 3.8.6界面设计

买家进入系统点击自己喜欢的商品页面，商家生成订单页面，买家可以进行相关评论。当买家进入历史购买记录时，可以进行评论的修改。

### 3.8.7接口设计

**private** **static** **final** String ***url*** = "jdbc:mysql://localhost:3306/university?useUnicode=true&characterEncoding=UTF-8&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&useSSL=false&serverTimezone=UTC&autoReconnect=true";

**private** **static** **final** String ***user*** = "root";

**private** **static** **final** String ***password*** = "li3660857";

String sql="update goods set evaluation ="+comment+" where goods\_id="+goods\_id;

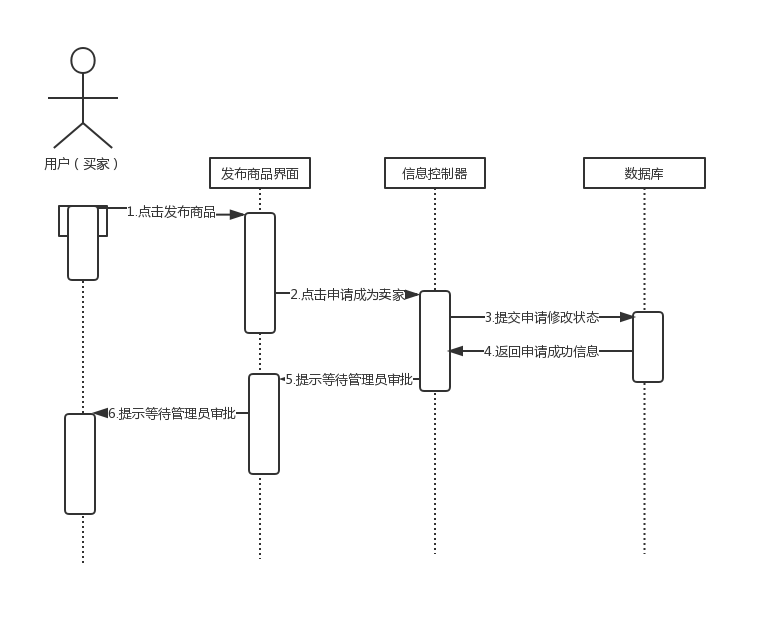
DAO 和servlet 中的类

### 3.8.8算法设计

使用了迭代算法，将数据库中用户、管理员等的相关属性，当用户购买商品时，用户的权限被赋予了管理员，管理员可以根据用户的相关操作，进行对商家相关功能的评论，管理员无权限对用户的评论信息进行相关的删除和更改。

## 3.9用户申请成为卖家

### 3.9.1功能



处理用户想要申请成为卖家的情况，非卖家用户可提交申请，并等待管理员审核批准。

### 3.9.2性能

提交申请后会立即反馈“等待管理员审核”

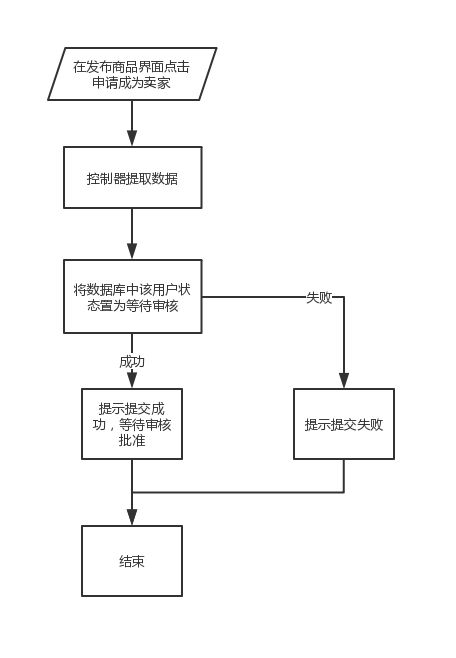
### 3.9.3输入项目

申请成为买家按钮

### 3.9.4输出项目

提示信息“等待管理员审批”

### 3.9.5程序逻辑



### 3.9.6界面设计

对买家用户，在登陆后点击发布商品按钮，进入发布商品界面，在发布商品界面，点击申请成为卖家，会，会向数据库发起连接，若成功则界面提示等待管理员审核，若失败则提示申请失败，请重新申请。

### 3.9.7接口

申请功能首先调用Servlet包中LichengxiServlet进行处理，通过session读取user\_id，之后将其传给DAO包中LichengxiDAO，连接数据库。

### 3.9.8算法

用户提交成为卖家的申请，控制器根据通过session读取该用户user\_id属性信息，进入数据库修改用户表数据，根据user\_id检索到将该用户将其状态user\_status置为等待审核状态，修改成功会返回给用户以等待管理员审核的提示。

## 3.10 悬赏功能

### 3.10.1 功能

当用户发现没有需要的二手商品的时候，可以在该系统上进行发布悬赏信息，系统会保存用户发布的二手信息。如果有人在之后发布相应的商品的时候，系统可以根据关键词进行匹配，然后将相关的商品信息推送给该用户。

### 3.10.2 性能

用户可以在该系统中发布所需要商品的信息，该功能的性能依赖于实现的算法。

### 3.10.3 输入项目

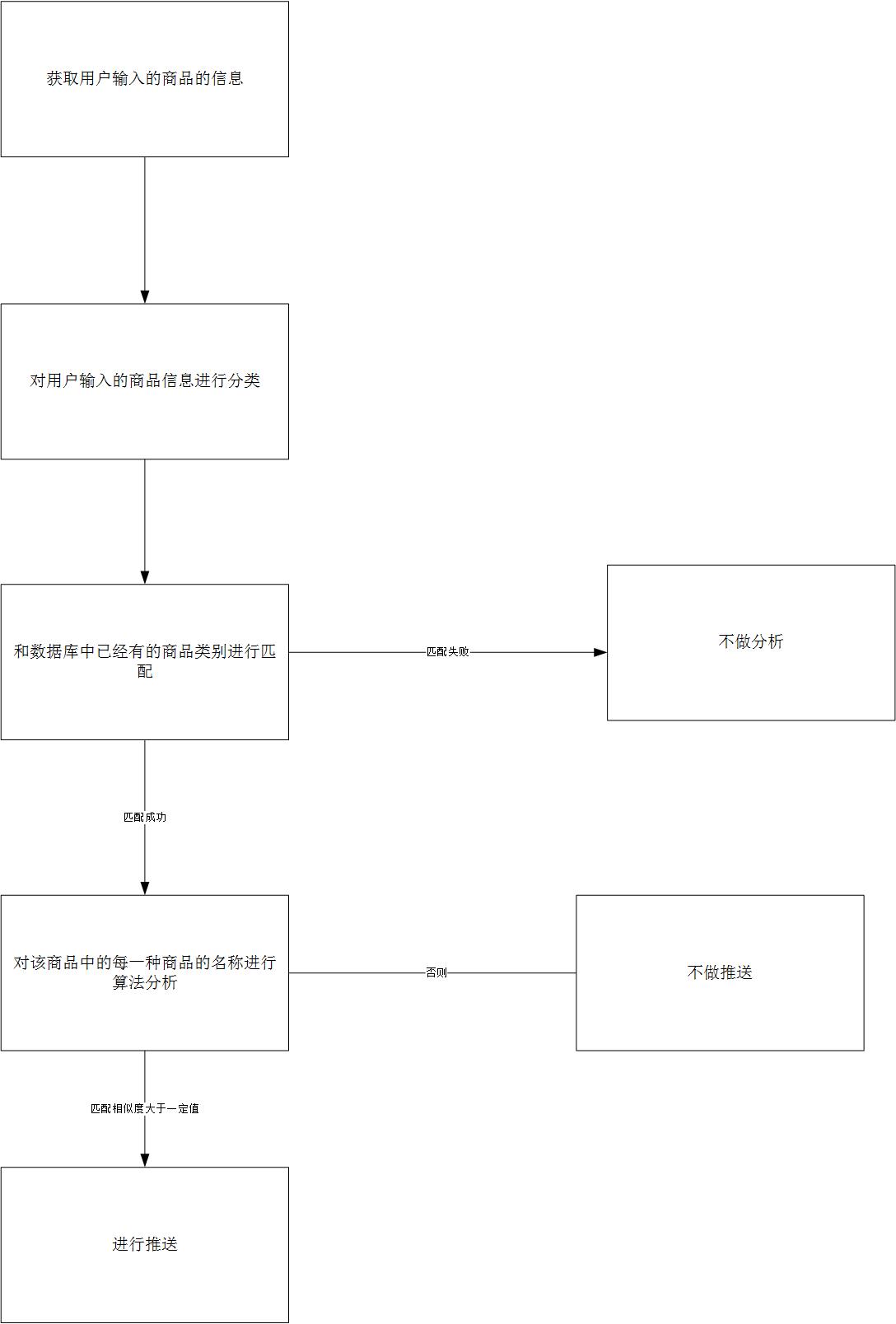
用户通过输入请求商品的名字和请求商品的类别，通过数据库封装在下面的表格中。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否为空 | 主/外键 |
| query\_id | int | 11 | 否 | 是 |
| user\_id | int | 11 | 否 | 否 |
| query\_name | varchar | 50 | 否 | 否 |
| category\_id | int | 11 | 否 | 否 |

### 3.10.4 输出项目

系统根据输入的商品查看是否有相关的产品上架，如果有的话就给用户返回响应的商品

### 3.10.5 流程图



### 3.10.6界面设计

### 3.10.7接口设计

### 3.10.8算法设计

该模块通过系统存储用户请求的商品分类，然后通过用户上架商品的分类进行匹配，如果是该分类的话，就在该分类中找寻是否有和请求相匹配的商品，通过聚类算法分析两种商品的匹配程度，当匹配程度大于一定值的时候，我们就认为这两种商品是同一种商品，之后系统会把两种商品认为是同一种商品，并推送给用户。

### 3.11 审核功能

### 3.11.1 功能

作为一个用户，当他想从买家变成卖家的时候，他会向系统发出申请，而管理员可以收到用户发出的申请，管理员可以对用户发出的申请作出审核，来决定该用户时候可以成为一个用户。

### 3.11.2 性能

审核需要系统管理员登录到该系统中，安全系数比较高，基本性能要求为系统响应时间不能超过1S.

### 3.11.3 输入项目

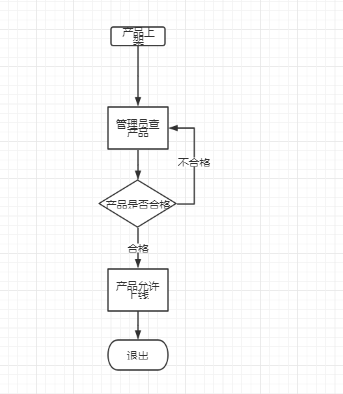
用户通过输入请求商品的名字和请求商品的类别，通过数据库封装在下面的表格中。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 是否为空 | 主/外键 |
| query\_id | int | 11 | 否 | 是 |
| user\_id | int | 11 | 否 | 否 |
| query\_name | varchar | 50 | 否 | 否 |
| category\_id | int | 11 | 否 | 否 |

### 3.11.4 输出项目

管理员根据输入的商品查看是否有相关的产品上架，如果有的话就审核产品，若全部通过则审核成功。

### 3.11.5流程图



### 3.11.6界面设计

管理员通过浏览用户，管根据输入的商品查看是否有相关的产品上架，如果有的话就审核产品，若全部通过则审核成功。点击审核按钮则审核成功，并且之后可以点击查看所有被审核过的用户

### 3.11.7接口设计

首先获取要被审核的用户id，通过调用managementServlet.java里的verify方法，接着verify方法调用xuejinlongDao.java里的verify方法来更新数据库里的用户状态，接着再通过调用managementServlet.java里的listUsers方法来显示用户。

### 3.11.8算法设计

该模块通过上架的产品，管理员对其审核，若产品合格，则认为审核通过，如果产品不合格，则认为审核不通过。

## 3.12管理员增删改查用户和产品的功能

### 3.12.1作为管理员可以对用户信息进行增删改查；

#### 3.12.1.1功能

管理员在登录系统后，可以进行对用户信息的增加、删除、修改、以及查看。

#### 3.12.1.2性能

响应时间不能超过1S，同时允许多个管理员进行操作。

#### 3.12.1.3输入项目

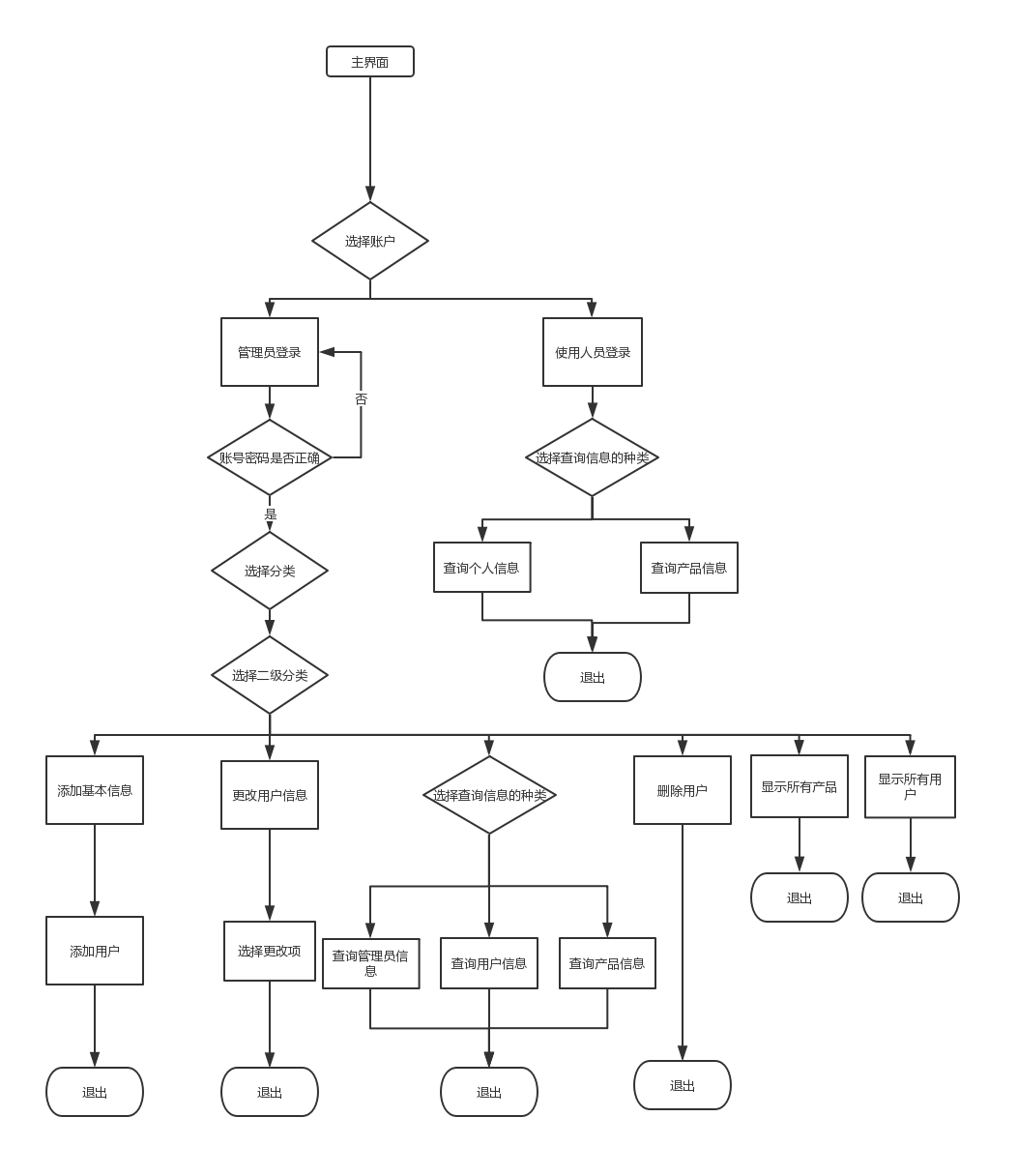
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 管理员账号 | Int | 11 |  | 不允许 | 主 |
| 管理员密码 | Varchar | 50 |  | 不允许 |  |

#### 3.12.1.4输出项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 用户id | Int | 11 |  | 不允许 | 主 |
| 用户姓名 | Varchar | 50 |  | 不允许 |  |
| 用户密码 | Varchar | 50 |  | 不允许 |  |
| 电话 | Int | 11 |  | 不允许 |  |
| 邮箱 | Varchar | 50 |  | 不允许 |  |
| 用户状态 | Int | 11 |  | 不允许 |  |

根据用户的请求，管理员进行对用户信息的增加、删除、修改、以及查看。

#### 3.12.1.5流程图



#### 3.12.1.6界面设计

如上述流程图所示，点击主界面，选择账户类型，如果是管理员，进入管理员界面，进行对应的增删改查操作，比如进入到添加用户界面，弹出录入姓名、学号、电话等输入框，如果添加成功则跳转到显示用户界面。

#### 3.12.1.7接口设计

对于增加用户，首先通过addUser.jsp中form表单获取前台用户信息数据，点击add按钮调用managementServlet.java里的addUser方法，通过将数据封装成一个user对象后，接着调用xuejinlongDao.java里的adduser方法，将user信息数据写入数据库里，然后在managementServlet.java里的adduser方法调用listusers方法，添加成功后在界面上显示所有的用户。

对于删除用户，首先获取要被删除用户的id，通过点击delete按钮来调用managementServlet.java里的deleteUser方法，根据此id来调用xuejinlongDao.java里的deleteUser方法，将这个用户从数据库里删除，接着在managementServlet.java里调用listUser方法来显示所有用户。

对于修改用户，首先通过editUser.jsp中form表单获取前台用户信息数据，点击edit按钮调用managementServlet.java里的editUser方法，通过将数据封装成一个user对象后，接着调用xuejinlongDao.java里的update方法，将user信息数据从数据库里更新，然后在managementServlet.java里的edituser方法调用listusers方法，添加成功后在界面上显示所有的用户。

对于查找和删除用户，首先在前台获取搜索框中的userId，通过search按钮调用managementServlet.java里的searchUser方法，之后调用xuejinlongDao.java里的findUserById方法来找到这个用户，在managementServlet.java的searchUser里设置attribute值，从而在search.jsp里获取到这个attribute值后将该用户的信息显示出来。

#### 3.12.1.8算法设计

管理员登陆系统后，获得用户的请求，比如获得用户的增加请求，系统给用户弹出信息录入界面，用户输入姓名、学号、电话等信息，如果输入非法字符，则提示输入错误，全部正确则录入成功，修改、删除、查看算法设计类似。

### 3.12.2作为管理员可以对被举报的违规产品进行下架和恢复

#### 3.12.2.1功能

管理员在登录系统后，可以进行对被举报的违规产品进行下架和恢复

#### 3.12.2.2性能

响应时间不能超过1S，同时允许多个管理员进行操作。

#### 3.12.2.3输入项目

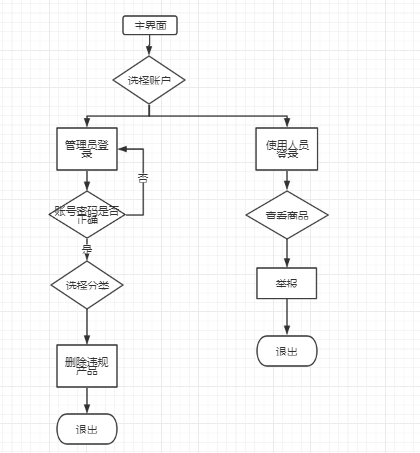
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 管理员账号 | Int | 11 |  | 不允许 | 主 |
| 管理员密码 | Varchar | 50 |  | 不允许 |  |

#### 3.12.2.4输出项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 产品id | Int | 11 |  | 不允许 | 主 |
| 产品名字 | Varchar | 50 |  | 不允许 |  |
| 描述 | Varchar | 50 |  | 不允许 |  |
| 产品状态 | Int | 11 |  | 不允许 |  |
| 商品id | int | 11 |  | 不允许 |  |

用户举报违规产品后，管理员可以对其进行下架或恢复。

#### 3.12.2.5流程图



#### 3.12.2.6界面设计

如上述流程图所示，点击主界面，选择账户类型，如果是用户，进入用户界面，举报产品，接着跳入举报成功界面，管理员进入管理者界面，点击违规产品，将被举报的产品下架或恢复。

#### 3.12.2.7接口设计

对于下架功能，首先获取到要被下架的产品id，通过点击下架按钮来调用managementServlet.java里的xiajia方法，接着调用xuejinlongDao.java里的findGoodById来找到这个产品，接着继续调用xuejinlongDao.java里的xiajia方法来更新数据库，从而下架成功，最后调用managementServlet.java里的listBeiJuBao方法显示所有被举报的产品。

同样对于恢复功能，首先获取到要被下架的产品id，通过点击下架按钮来调用managementServlet.java里的recover方法，接着调用xuejinlongDao.java里的findGoodById来找到这个产品，接着继续调用xuejinlongDao.java里的recover方法来更新数据库，从而下架成功，最后调用managementServlet.java里的listBeiJuBao方法显示所有被举报的产品。

#### 3.12.2.8算法设计

用户举报违规产品，管理员看到后，决定是否确实违规，如果着实违规，则删除产品，如果没有违规，则恢复该产品。

### 3.12.3**作为管理员可以显示被举报、下架、恢复正常的产品**

#### 3.12.3.1功能

管理员在登录系统后，可以显示被举报、下架、恢复正常的产品。

#### 3.12.3.2性能

响应时间不能超过1S，同时允许多个管理员进行操作。

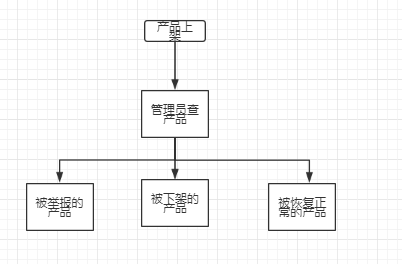
#### 3.12.3.3输入项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 管理员账号 | Int | 11 |  | 不允许 | 主 |
| 管理员密码 | Varchar | 50 |  | 不允许 |  |

#### 3.12.3.4输出项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 产品id | Int | 11 |  | 不允许 | 主 |
| 产品名字 | Varchar | 50 |  | 不允许 |  |
| 描述 | Varchar | 50 |  | 不允许 |  |
| 产品状态 | Int | 11 |  | 不允许 |  |
| 商品id | int | 11 |  | 不允许 |  |

#### 3.12.3.5流程图



#### 3.12.3.6界面设计

如上述流程图所示，点击主界面，选择账户类型，管理员进入管理者界面，点击查看商品，可以选择查看被举报的产品的界面，也可以进入到被下架的界面，同样可以进入到被恢复正常的界面。

#### 3.12.3.7接口设计

对于显示被举报的产品，通过调用managementServlet.java里的listBeiJuBao方法，接着调用xuejinlongDao.java里的findAllBeiJuBao方法获取所有被举报的产品，然后设置一个attribute值供listBeiJuBao.jsp获取从而显示所有被举报的产品。

对于显示被下架的产品，通过调用managementServlet.java里的listXiaJia方法，接着调用xuejinlongDao.java里的findAllXiaJia方法获取所有被下架的产品，然后设置一个attribute值供listXiaJia.jsp获取从而显示所有被下架的产品。

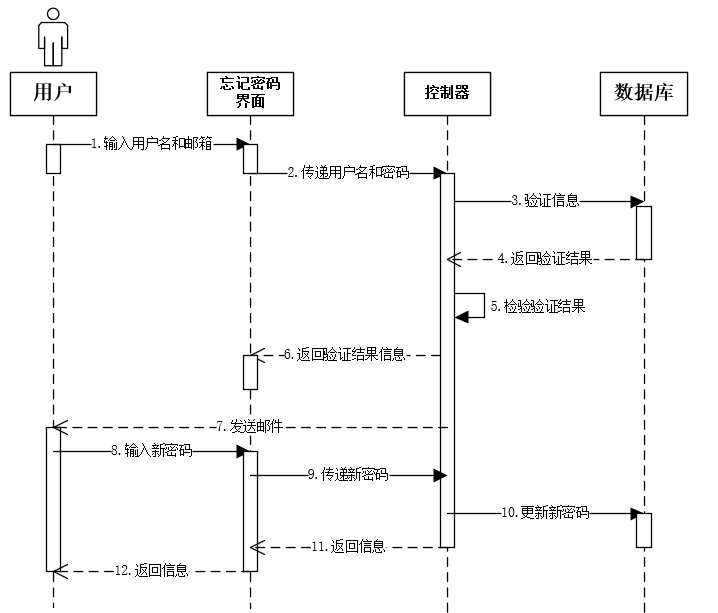
对于显示被恢复正常的产品，通过调用managementServlet.java里的listNormal方法，接着调用xuejinlongDao.java里的findAllNormal方法获取所有被恢复正常的产品，然后设置一个attribute值供listNormal.jsp获取从而显示所有被恢复正常的产品。

#### 3.12.3.8算法设计

如果有被举报的产品，则将被举报的产品在被举报的界面上显示出来；如果有被下架的产品，则将被下架的产品在被下架的界面上显示出来；如果有被恢复正常的产品，则将被恢复正常的产品在被恢复正常的界面上显示出来；

## 3.13找回密码

### 3.13.1功能



用户可以在忘记密码界面输入用户名和事先绑定的邮箱，系统给指定邮箱发送更改密码的邮件，用户通过邮件给的链接进入密码更改页面，然后返回登陆界面。

### 3.13.2性能

发送邮件的时间不超过20s，更改后的密码同样需要进行加密传输与存储。

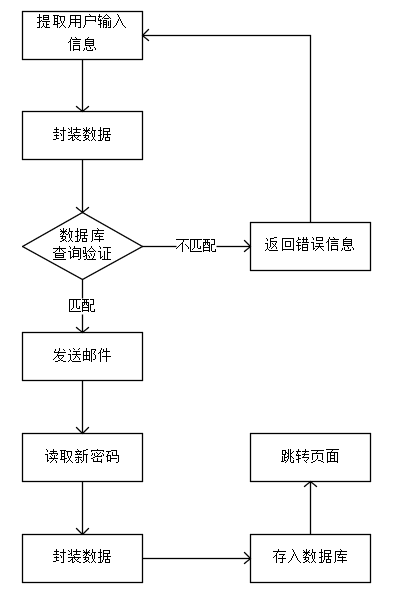
### 3.13.3输入项目

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 类型 | 长度 | 小数点 | 允许空 | 主/外键 |
| 用户名 | Varchar |  |  |  | 主 |
| 邮箱 | Varchar |  |  |  |  |
| 密码 | Varchar |  |  |  |  |

### 3.13.4输出项目

如果用户名和邮箱不能进行匹配则返回错误信息，反之返回发送邮件成功的信息；如果更改密码成功则返回成功信息并跳转页面。

### 3.13.5流程图



### 3.13.6界面设计

从忘记密码界面可以点击返回按钮返回到登陆界面。

### 3.13.7接口设计

### 3.13.8算法设计

通过邮箱给的POP3/SMTP/IMAP服务发送邮件。

# 4数据字典

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 允许为空 | PK | 默认值 | 字段说明 |
| admin\_id | int(11) | NO | auto\_increment |  |  |
| admin\_number | int(11) | NO |  |  |  |
| admin\_password | varchar(50) | NO |  |  |  |
| category\_id | int(11) | NO | auto\_increment |  |  |
| category\_name | varchar(50) | NO |  |  |  |
| goods\_id | int(11) | NO | auto\_increment |  |  |
| user\_id | int(11) | NO |  |  |  |
| goods\_name | varchar(50) | NO |  |  |  |
| description | varchar(255) | NO |  |  |  |
| goods\_status | int(11) | NO |  |  | 0代表正常上架，1代表被举报，2代表正在交易的，3代表已交易 |
| category\_id | int(11) | NO |  |  | 分类id |
| price | double(10,2) | NO |  |  |  |
| information\_id | int(11) | NO | auto\_increment |  |  |
| buyer\_id | int(11) | YES |  |  |  |
| seller\_id | int(11) | YES |  |  |  |
| content | varchar(255) | NO |  |  |  |
| sendtime | date | NO |  |  |  |
| nextinfor\_id | int(11) | YES |  |  |  |
| information\_status | int(11) | NO |  |  |  |
| order\_id | int(11) | NO | auto\_increment |  |  |
| user\_id | int(11) | NO |  |  | 买家的id |
| orderitem\_id | int(11) | NO | auto\_increment |  |  |
| order\_id | int(11) | NO |  |  |  |
| goods\_id | int(11) | NO |  |  | 物品id，用来查找物品信息 |
| orderitem\_status | int(11) | NO |  |  | 0，代表买家，卖家未确认，1代表一方未确认，2代表都确认 |
| query\_id | int(11) | NO | auto\_increment |  |  |
| user\_id | int(11) | NO |  |  |  |
| query\_name | varchar(50) | NO |  |  |  |
| category\_id | int(11) | NO |  |  |  |
| user\_id | int(11) | NO | auto\_increment |  |  |
| user\_name | varchar(50) | NO |  |  |  |
| user\_password | varchar(50) | NO |  |  |  |
| phone | int(11) | NO |  |  |  |
| user\_number | int(11) | NO |  |  |  |
| email | varchar(50) | NO |  |  |  |
| user\_status | int(11) | NO |  |  | 0代表买家，1代表卖家。 |

# 5数据库设计

