

天津大学

商城设计文档



学 部:	智能与计算学部
专 业:	软件工程
年 级:	2018 级
组长姓名:	张雨茜
组长学号:	3017207084
组员姓名:	陈旭扬
组员学号:	3018216221
组员姓名:	姜天杰
组员学号:	3018216226
组员姓名:	崔广睿
组员学号:	3018216285
授课教师:	李罡

2020 年 6 月 8 日

目录

1.引言	1
1.1 编写目的	1
1.2 背景	1
2 概要设计	1
2.1 系统目标	1
2.2 主要功能	1
2.3 运行要求	2
2.4 主要约束与限制	2
3 架构设计	2
3.1 数据流复审	2
3.1.1 静态数据	2
3.1.2 动态数据	3
3.2 总体架构	5
3.2.1 架构图	5
3.2.2 类图	6
4 详细设计	7
4.1 接口设计	7
4.1.1 管理员模块	7
4.1.2 用户模块	21
4.2 模块设计	40
5.数据库设计	58
5.1 E-R 图	58
5.2 数据库建表	58
5.2.1 活动表 activity	58
5.2.2 地址表 address	59
5.2.3 管理员表 admin	60
5.2.4 商品分类表 category	60
5.2.5 订单表 collection	61
5.2.6 评论表 comment	61
5.2.7 订单传送表 deliver	62
5.2.8 商品表 goods	63
5.2.9 图片路径表 imagepath	64
5.2.10 订货单表 indent	64
5.2.11 订单项目表 orderitem	65
5.2.12 购物车表 shopcart	66
5.2.13 用户表 user	67

1.引言

1.1 编写目的

本阶段完成商城系统的框架设计并明确系统的数据结构与软件总体结构。在软件设计阶段主要是把用户需求转化为软件的功能表示,目的是进一步细化软件设计阶段得出的软件总体概貌,把它加工成在程序细节上非常接近于源程序的软件表示。本软件设计书是系统设计的依据,是系统分析与设计人员的必读的参考用书。

该设计文档的编写目的旨在推动本项目的规范化,使设计人员遵循统一的详细设计书写规范,节省制作文档的时间,降低系统实现的风险,做到系统设计资料的规范性与全面性,以利于系统的实现、测试、维护、版本升级等。

1.2 背景

作为互联网战略的延伸,电子商务的热潮一度兴起,但是一直以来,很多企业网站都局限在信息发布等一些基本的功能,而越来越深入的实践表明,这些仅仅代表了企业网站的一个起点功能。随着企业网络环境的发展成熟,企业门户网站可以帮助企业把现有的业务系统整合起来,集中优势资源为客户提供个性化服务,并为企业创建一个良好的收入渠道。网上购物在线网站致力于提供产品展示及订购为核心的网上购物服务宣传自己商店的产品并将自己的产品展现给客户,让客户通过网站便能对自由的选择地购买产品。

2 概要设计

2.1 系统目标

1. 提供以产品展示及订购为核心的网上购物服务。
2. 宣传自己商店的产品并将自己的产品展现给客户。
3. 让客户通过网站便能对自由的选择地购买产品。

2.2 主要功能

对于客户,能够在商城中完成浏览商品、下单购买、收货、评价等功能。

对于管理员,能够宣传产品并将自己的产品展现给客户,能够完成订单的管理,并且能够适时举办活动。

程序采用 B/S 框架,采用 springboot 框架,采用面向对象的设计方法,PnP 的设计理念,在框架的组织下,程序员无需直接了解项目的流程等细节,只需要关心起负责的功能模块,对程序员的要求大大降低,也避免了程序员个人能力不足对项目造成的影响。大大提高

了系统的可维护性，降低了维护的风险。

基于框架的程序设计实现了模块-视图-控制器的模式（MVC），每个部分各司其职。采用了 IOC（控制反转）和 AOP（面向切面）的方法来降低了系统模块间的耦合性，采用工厂模式把三层的对象都使用配置文件配置起来，当启动服务器应用加载的时候，让一个类中的方法通过读取配置文件，把这些对象创建并存起来。

2.3 运行要求

- 1) 硬件环境 可用网卡 支持无线网络连接或网线连接上网的计算机及移动设备
- 2) 软件环境 操作系统： Microsoft Windows 7/8/10 Mac OS 支持环境： Java 8 , Java springboot , MySql 数据库

2.4 主要约束与限制

- 1) 编程语言限制：使用 Java 语言进行程序编写，使用程序设计 springboot 框架，通过 RESTful API 与这个应用进行交互。
- 2) 工具约束：采用 IntelliJ IDEA、 VS code、 Workbench 开发
- 3) 性能约束：要求可以支持不少于四万用户数据的存储。
- 4) 使用场景：适用于商城用户的购买商品等活动。。
- 5) 功能设计具体要求：对于客户，能够在商城中完成浏览商品、下单购买、收货、评价等功能。对于管理员，能够宣传产品并将自己的产品展现给客户，能够完成订单的管理，并且能够适时举办活动。

3 架构设计

3.1 数据流复审

3.1.1 静态数据

名称	简述	有关数据流
用户	使用本商城的用户	登录系统请求、注册用户请求、退出登录系统请求、修改信息请求、修改密码请求、购物车管理请求、商品检索请求、商品浏览请求、查看商品详细信息指令
管理员	本商城的管理人员	登录系统请求、退出登录系统请求、查看用户列表请求、删除用户指令、查看商品列表请求、添加商品指令、编辑商品信息指令、管理

		商品分类请求、管理活动请求、删除商品指令、订单管理指令、活动管理指令
--	--	------------------------------------

3.1.2 动态数据

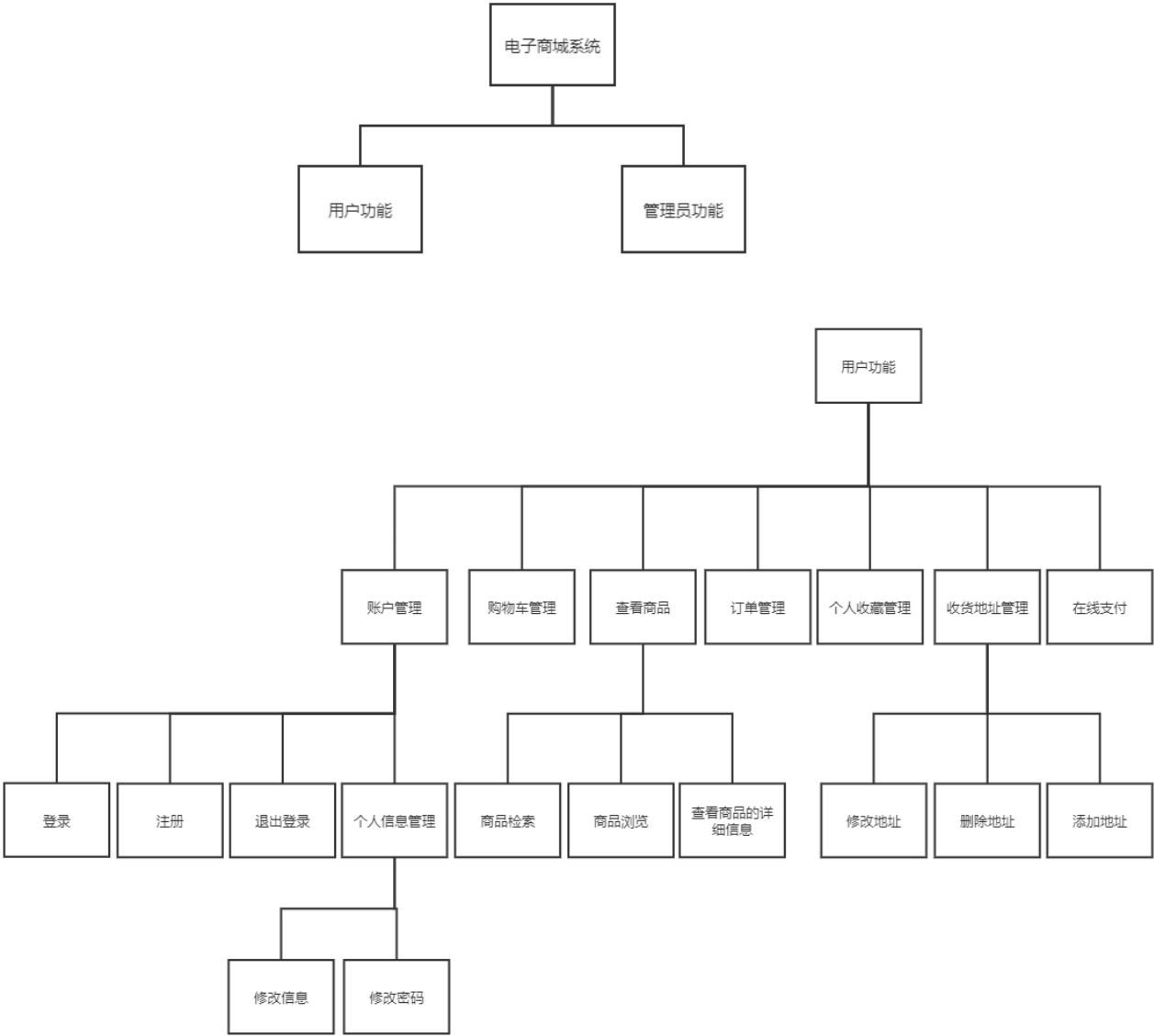
数据流名称	简述	定义	来源	去向
登录系统请求	用户使用账号和密码进行系统登录	登录请求=用户名+密码+验证码	用户	服务器
注册用户请求	用户使用用户名、密码、确认密码、邮箱、联系方式进行注册	注册请求=用户名+密码+确认密码+邮箱+联系方式	用户	服务器、数据存储
退出登录系统请求	用户进行退出系统	退出登录请求=用户名+密码	用户	服务器
修改信息请求	用户更新邮箱和手机号来修改信息	修改信息请求=邮箱+手机号	用户	服务器、数据存储
修改密码请求	用户更新密码来修改密码	修改密码请求=原密码+现密码	用户	服务器、数据存储
购物车管理请求	用户通过购物车管理查看每一个商品的图片，名称，价格，数量（可调）总价以及所有商品的总计，	购物车管理请求=图片+名称+价格+数量（可调）+商品总计	用户	服务器、数据存储
商品检索请求	用户在输入框进行搜索，查看对应商品的图片、名称、价格、记录个数	商品检索请求=图片+名称+价格+记录个数	用户	服务器
商品浏览请求	用户在首页进行浏览，查看商品的多级分类、图片、名称、价格、首页海报	商品浏览请求=多级分类信息+图片+名称+价格+首页海报	用户	服务器
查看商品详细信息指令	用户查看商品详细信息：图片，名称，价格，类别，剩余数量，优惠信息，商品说明，加入购物车的商品个数，收藏信息，商品描述	查看商品详细信息指令=图片+名称+价格+类别+剩余数量+优惠信息+商品说明+加入购物车的商品个数+收藏信息+商品描述	用户	服务器
订单管理	用户查看自己的订单详情界面	订单管理=订单类别+订单号+订单日期	用户	服务器
个人收藏管理	用户管理个人的收	收藏管理=商品列表	用户	服务器

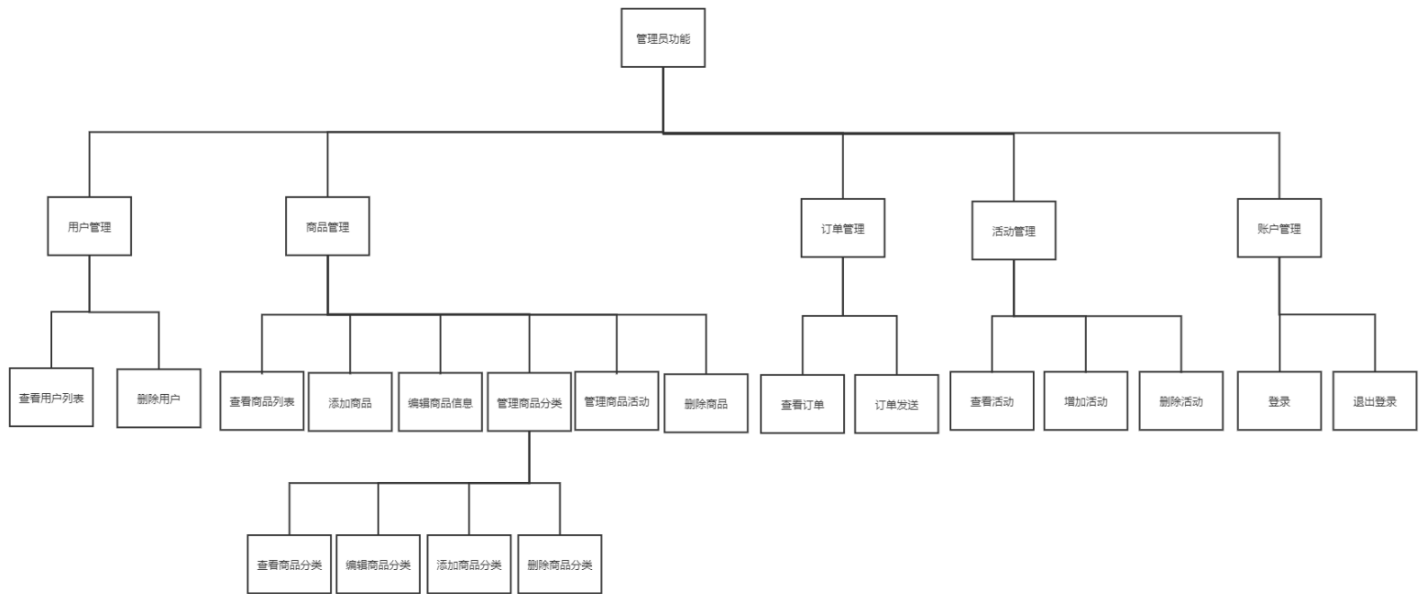
	藏商品界面			
修改地址	用户修改自己存储好的收货地址	修改地址=新的地址+用户 ID	用户	服务器
添加地址	用户添加新的收货地址	添加地址=新的地址+用户 ID	用户	服务器
删除地址	用户删除自己已有的收货地址	删除地址=地址信息+用户 ID	用户	服务器
在线支付	用户支付自己的未支付订单	在线支付=支付方式+订单 ID	用户	服务器
登录系统请求	管理员使用账号和密码进行系统登录	登录请求=登录用户名+密码	管理员	服务器
查看用户信息	管理员申请查看用户列表的响应	用户列表=用户信息列表	管理员	服务器
请求删除用户信息	管理员申请删除用户信息成功的响应	成功信息=删除成功	管理员	数据库系统数据存储
查看商品信息	管理员申请查看商品列表的响应	商品列表=商品信息列表	管理员	服务器
添加商品请求	管理员申请增加商品信息	商品信息=商品 id+商品价格+商品数量	管理员	数据库系统数据存储、服务器
编辑商品信息请求	管理员申请编辑商品信息	商品信息=商品 id+商品价格+商品数量	管理员	数据库系统数据存储、服务器
查看商品分类信息请求	管理员申请查看商品分类的响应	商品分类=商品分类信息列表	管理员	服务器
编辑商品分类请求	管理员申请编辑商品分类信息	商品分类=商品分类信息列表	管理员	数据库系统数据存储、服务器
添加商品分类请求	管理员申请增加商品分类信息	商品分类=商品分类信息列表	管理员	数据库系统数据存储、服务器
删除商品分类请求	管理员申请删除商品分类信息	商品分类=商品分类信息列表	管理员	数据库系统数据存储
商品活动管理	管理员申请管理商品活动的响应	商品活动=活动列表	管理员	数据库系统数据存储、服务器
删除商品请求	管理员申请删除商品信息	商品信息=商品 id+商品价格+商品数量	管理员	数据库系统数据存储
查看订单	管理员查看所有订单信息	查看订单=订单列表	管理员	服务器、数据库系统数据存储
订单发货指令	管理员完成订单的发货	订单发货=订单信息	管理员	服务器、数据库系统数据存储
查看活动请求	管理员申请查看活动信息	查看活动=活动列表	管理员	服务器、数据库系统数据存储
增加活动指令	管理员添加活动的信息	添加活动=活动信息	管理员	服务器、数据库系统数据存储
删除活动指令	管理员删除某一活动的信息	删除活动=活动信息	管理员	服务器、数据库系统数据存储

登录系统请求	用户使用账号和密码进行系统登录	登录请求=用户名+密码+验证码	管理员	服务器
退出登录系统请求	用户进行退出系统	退出登录请求=用户名+密码	管理员	服务器

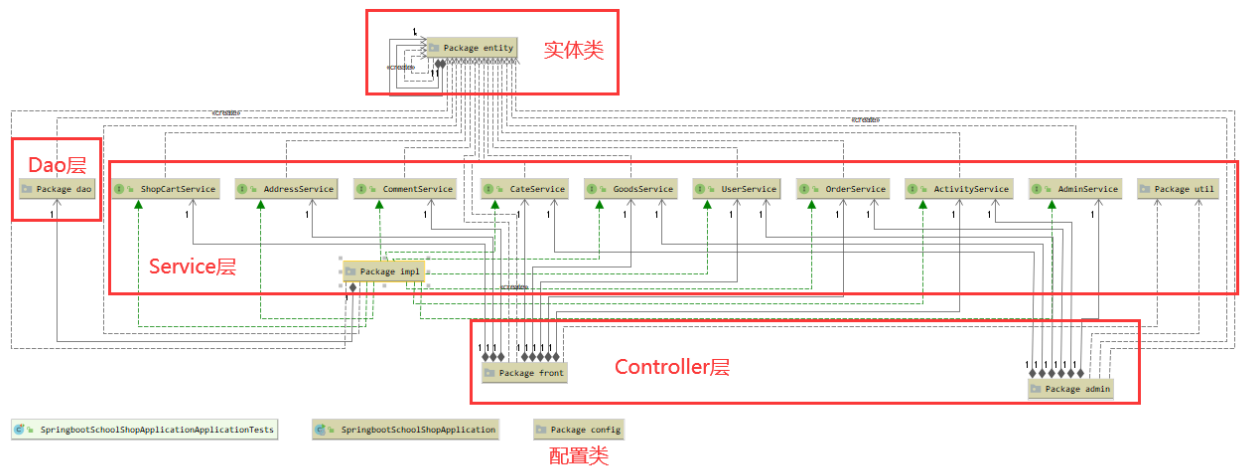
3.2 总体架构

3.2.1 架构图





3.2.2 类图



4 详细设计

4.1 接口设计

4.1.1 管理员模块

4.1.1.1 登陆请求：

URL: </shop/admin/login>

Method: GET

入参: 无

返回实例:

```
{
  "code": 100,
  "msg": "登录",
  "info": {
    "url": "/confirmLogin"
  }
}
```

4.1.1.2 确认身份

URL: </shop/admin/confirmlogin>

Method: POST

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	adminName	admin	名称
<input checked="" type="checkbox"/>	password	11111111	密码

返回实例:

```
{
  "code": 100,
  "msg": "登录",
  "info": {
    "result": "0",
    "reason": "用户名或密码错误",
    "url": "/confirmLogin"
  }
}
```

4.1.1.3 退出登录

[URL: /shop/admin/logout](/shop/admin/logout)

Method: GET

入参: 无

返回实例:

```
{
  "code": 100,
  "msg": "注销",
  "info": {
    "result": "1",
    "url": "/admin/login"
  }
}
```

4.1.1.4 获取用户信息

[URL: /shop/admin/user/show](/shop/admin/user/show)

Method: GET

入参: 无

返回实例:

```
{
  "code": 100,
  "msg": "请求成功!",
  "info": {
    "url": "/admin/user/showAllUsers"
  }
}
```

4.1.1.5 查询用户信息

[URL: /shop/admin/user/showAllUsers](/shop/admin/user/showAllUsers)

Method: GET

入参: 无

返回实例:

```

{
  "code": 100,
  "msg": "查询成功!",
  "info": {
    "userList": [
      {
        "userid": 8,
        "username": "zzz",
        "password": "25f9e794323b453885f5181f1b624d0b",
        "regtime": "2020-05-24T04:40:04.000+0000",
        "email": "123456789@163.com",
        "telephone": "13820178909"
      },
      {
        "userid": 9,
        "username": "lalal",
        "password": "bc11f06afb9b27070673471a23ecc6a9",
        "regtime": "2020-05-24T04:41:39.000+0000",
        "email": "zzz@163.com",
        "telephone": "13456789123"
      }
    ]
  }
}

```

4.1.1.6 删除用户信息

URL: </shop/admin/user/delete/{userid}>

Method: DELETE

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	userid	1	用户ID

返回实例:

```

"code": 100,
"msg": "删除成功!",
"info": {
  "result": "1",
  "url": "/admin/user/showAllUsers"
}

```

4.1.1.7 查询商品信息

URL: </shop/admin/goods/showAllDoods>

Method: GET

入参: 无

返回实例:

```

{
  "code": 100,
  "msg": "查询成功!",
  "info": {
    "employees": [
      {
        "goodsid": 94,
        "goodsname": "测试商品2",
        "price": 23,
        "num": 34,
        "uptime": "2019-05-10T11:44:28.000+0000",
        "category": 2,
        "detailcate": "测试",
        "activityid": 1,
        "description": "这是测试商品2",
        "imagePaths": null,
        "fav": false,
        "activity": null,
        "newPrice": null
      },
      {↔},
      {↔},
      {↔}
    ]
  }
}

```

4.1.1.8 查询商品分类信息

[URL: /shop/admin/goods/showCategory](/shop/admin/goods/showCategory)

Method: GET

入参: 无

返回实例:

```
{
  "code": 100,
  "msg": "查询成功!",
  "info": {
    "categoryList": [
      {
        "cateid": 1,
        "catename": "数码"
      },
      {
        "cateid": 2,
        "catename": "服饰"
      },
      {
        "cateid": 3,
        "catename": "家电"
      },
      {
        "cateid": 4,
        "catename": "书籍"
      },
      {
        "cateid": 5,
        "catename": "其他"
      }
    ]
  }
}
```

4.1.1.9 增加商品信息

[URL: /shop/admin/goods/add](#)

Method: GET

入参: 无

返回实例:

```

{
  "code": 100,
  "msg": "添加商品",
  "info": {
    "msg": "",
    "categoryList": [
      {
        "cateid": 1,
        "catename": "数码"
      },
      {
        "cateid": 2,
        "catename": "服饰"
      },
      {
        "cateid": 3,
        "catename": "家电"
      },
      {
        "cateid": 4,
        "catename": "书籍"
      },
      {
        "cateid": 5,
        "catename": "其他"
      }
    ]
  },
  "url": "/admin/goods/addGoodsSuccess"
}

```

4.1.1.10 修改商品信息

[URL: /shop/admin/goods/update](/shop/admin/goods/update)

Method: POST

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	goodsId	93	编号

返回实例:

```

{
  "code": 100,
  "msg": "更新成功!",
  "info": {
    "result": "1",
    "url": "/admin/show"
  }
}

```

4.1.1.11 删除商品

[URL: /shop/admin/goods/delete/{goodsid}](/shop/admin/goods/delete/{goodsid})

Method: DELETE

入参：

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	goodsId	93	编号

返回实例：

```
{
  "code": 100,
  "msg": "删除成功！",
  "info": {
    "result": "1",
    "url": "/admin/show"
  }
}
```

4.1.1.12.添加商品成功

[URL: /shop/admin/goods/addGoodsSuccess](/shop/admin/goods/addGoodsSuccess)

Method: POST

入参：

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	goodsName	测试商品	名称
<input checked="" type="checkbox"/>	price	99	价格
<input checked="" type="checkbox"/>	num	100	数量
<input checked="" type="checkbox"/>	category	1	分类

返回实例：

```
"code": 100,
"msg": "商品添加成功！",
"info": {
  "result": "1",
  "url": "/admin/goods/add"
}
```

4.1.1.13 增加分类信息

[URL: /shop/admin/goods/addCategory](/shop/admin/goods/addCategory)

Method: POST

入参：

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	cateId	5	编号
<input checked="" type="checkbox"/>	cateName	其他	名称

返回实例:

```
{
  "code": 100,
  "msg": "添加分类",
  "info": {
    "msg": "",
    "categoryList": [
      {
        "cateid": 1,
        "catename": "demoData"
      },
      {
        "cateid": 2,
        "catename": "服饰"
      },
      {
        "cateid": 3,
        "catename": "家电"
      },
      {
        "cateid": 4,
        "catename": "书籍"
      },
      {
        "cateid": 5,
        "catename": "其他"
      }
    ]
  },
  "url": "/admin/goods/addCategoryResult"
}
```

4.1.1.14 增加商品分类

[URL: /shop/admin/goods/addCategoryResult](#)

Method: POST

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	catename	其他	名称

返回实例:

```
{
  "code": 200,
  "msg": "添加分类",
  "info": {
    "result": "0",
    "reason": "分类已存在",
    "url": "/admin/goods/addCategory"
  }
}
```


入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	catename	测试	名称

返回实例

```
{
  "code": 100,
  "msg": "添加分类",
  "info": {
    "result": "1",
    "url": "/admin/goods/addCategory"
  }
}
```

4.1.1.15 更新分类信息

[URL: /shop/admin/goods/saveCate](/shop/admin/goods/saveCate)

Method: POST

入参:

Query Params			
	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	catename	测试	

返回实例:

```
{
  "code": 100,
  "msg": "更新分类",
  "info": {
    "result": "1",
    "url": "/admin/goods/addCategory"
  }
}
```

入参:

Query Params			
	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	catename	其他	

返回实例:

```
{
  "code": 200,
  "msg": "名字已经存在",
  "info": {
    "result": "0",
    "url": "/admin/goods/addCategory"
  }
}
```

4.1.1.16 删除分类信息

URL: </shop/admin/goods/deleteCate>

Method: DELETE

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	catename	其他	

返回实例

```
{
  "code": 100,
  "msg": "删除成功",
  "info": {
    "result": "1",
    "url": "/admin/goods/addCategory"
  }
}
```

4.1.1.17 查询订单

URL: </shop/admin/order/send>

Method: GET

入参: 无

返回实参

```
{
  "code": 100,
  "msg": "查询订单",
  "info": {
    "sendOrder": [],
    "url": "redirect:/admin/order/send"
  }
}
```

4.1.1.18 发货

URL: </shop/admin/order/sendDoods>

Method: POST

入参:

	KEY	VALUE
<input checked="" type="checkbox"/>	orderID	2

返回实参

```
{
  "code": 100,
  "msg": "发货",
  "info": {
    "url": "redirect:/admin/order/send"
  }
}
```

4.1.1.19 查询待收货订单

URL: </shop/admin/order/receive>

Method: GET

入参: 无

返回实参

```
{
  "code": 100,
  "msg": "查询订单",
  "info": {
    "sendOrder": [],
    "url": "redirect:/admin/order/send"
  }
}
```

4.1.1.20 查询完成订单

URL: </shop/admin/order/complete>

Method: GET

入参: 无

返回实参

```
{
  "code": 100,
  "msg": "查询订单",
  "info": {
    "sendOrder": [],
    "url": "redirect:/admin/order/send"
  }
}
```

4.1.1.21 查询活动

URL: </shop/admin/activity/show>

Method: GET

入参: 无

返回实参

```
{
  "code": 100,
  "msg": "获取活动信息",
  "info": {
    "activityList": [
      {
        "activityid": 2,
        "activityname": "开业大酬宾",
        "activitydes": "开业大酬宾, 全场9折。",
        "discount": 9.2,
        "fullprice": 100,
        "reduceprice": 8,
        "fullnum": 200,
        "reducenum": 20
      }
    ],
    "url": "/admin/activity/showActivity"
  }
}
```

4.1.1.22 获取活动信息

[URL: /shop/admin/activity/showActivity](/shop/admin/activity/showActivity)

Method: GET

入参: 无

返回实参

```
{
  "code": 100,
  "msg": "获取活动信息",
  "info": {
    "activityList": [
      {
        "activityid": 2,
        "activityname": "开业大酬宾",
        "activitydes": "开业大酬宾, 全场9折。",
        "discount": 9.2,
        "fullprice": 100,
        "reduceprice": 8,
        "fullnum": 200,
        "reducenum": 20
      }
    ]
  }
}
```

4.1.1.23 添加活动

URL: </shop/admin/activity/add>

Method: GET

入参: 无

返回实参

```
{
  "code": 100,
  "msg": "请求成功",
  "info": {
    "url": "/admin/activity/addResult"
  }
}
```

4.1.1.24 增加活动

URL: </shop/admin/activity/addResult>

Method: POST

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	activityName	测试活动	名称
<input checked="" type="checkbox"/>	activityDes	12345	宣传语
<input checked="" type="checkbox"/>	discount	9	折扣
<input checked="" type="checkbox"/>	fullPrice	100	满减价格
<input checked="" type="checkbox"/>	reducePrice	10	减少价格
<input checked="" type="checkbox"/>	fullnum	1231	最多上限
<input checked="" type="checkbox"/>	reduceNum	12	最低下限

返回实例:

```
{
  "code": 100,
  "msg": "添加商品活动",
  "info": {
    "result": "1",
    "url": "/admin/activity/show"
  }
}
```

4.1.1.25. 更新活动

URL: </shop/admin/activity/update>

Method: POST

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	goodsId	93	商品编号
<input checked="" type="checkbox"/>	activityId	2	活动编号

返回实例:

```
{
  "code": 100,
  "msg": "更新商品活动",
  "info": {
    "result": "1",
    "url": "/admin/activity/show"
  }
}
```

4.1.1.26. 删除活动

URL: </shop/admin/activity/delete>

Method: DELETE

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	activityID	2	
<input checked="" type="checkbox"/>	activityName	<input type="text" value="开业大酬宾"/>	

返回实例:

```
{
  "code": 100,
  "msg": "删除商品活动",
  "info": {
    "result": "1",
    "url": "/admin/activity/show"
  }
}
```

4.1.2 用户模块

4.1.2.1 登陆请求

URL: /shop/login

Method: GET

入参: 无

返回实例:

```
1  {
2      "code": 100,
3      "msg": "请求成功",
4      "info": {
5          "uri": "login"
6      }
7  }
```

4.1.2.2 登陆确认

URL: /loginconfirm

Method: POST

入参:

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	username	陈旭扬	账号
<input checked="" type="checkbox"/>	password	123456789	密码
<input checked="" type="checkbox"/>	confirmlogo		验证码

返回实例

```
{
  "code": 100,
  "msg": "请求成功",
  "info": {
    "user": {
      "userid": 8,
      "username": "陈旭扬",
      "password": "25f9e794323b453885f5181f1b624d0b",
      "regtime": "2020-05-23T11:30:21.000+0000",
      "email": "123456789@qq.com",
      "telephone": "123456"
    },
    "uri": "main"
  }
}
```

4.1.2.3 注册请求:

URL: /shop/register

Method: GET

入参: 无

返回实例

```
1  {
2      "code": 100,
3      "msg": "请求成功",
4      "info": {
5          "uri": "registerresult"
6      }
7  }
```

4.1.2.4 注册结果

URL: /shop/registerresult

Method: POST

入参:

	KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/>	userid	155	用户ID	
<input checked="" type="checkbox"/>	username	789456123	用户名	
<input checked="" type="checkbox"/>	password	789456123	用户密码	
<input checked="" type="checkbox"/>	regtime	2020/5/23 19:28	注册时间	
<input checked="" type="checkbox"/>	email	1520619515@qq.com	注册邮箱	
<input checked="" type="checkbox"/>	telephone	123456789	电话	

返回实例:

```
1  {
2      "code": 100,
3      "msg": "请求成功",
4      "info": {
5          "uri": "login"
6      }
7  }
```

4.1.2.5 访问主页

URL: /shop/main

methon: GET

入参: 通用参数

返回实例:


```

{
  "code": 100,
  "msg": "请求成功",
  "info": {
    "colGoods": [
    ],
    "bookGoods": null,
    "digGoods": [
      {
        "goodsid": 93,
        "goodsname": "java",
        "price": 60,
        "num": 1,
        "uptime": "2020-05-23T11:43:04.000+0000",
        "category": 1,
        "detailcate": "测试",
        "activityid": 1,
        "description": "这是测试商品1",
        "imagePaths": [
          {
            "pathid": 114,
            "goodid": 93,
            "path": "java.jpg"
          },
          {
            "pathid": 115,
            "goodid": 93,
            "path": "java.jpg"
          }
        ]
      }
    ]
  }
}

```

4.1.2.6 获取收货地址

URL: /shop/info/address

Method: GET

请求参数: 通用参数

返回实例:

```
4  {
5    "code": 100,
6    "msg": "请求成功",
7    "info": {
8      "addressList": [
9        {
10         "addressid": 14,
11         "userid": 8,
12         "province": "山西省",
13         "city": "晋中市",
14         "county": "祁县",
15         "detailaddr": "",
16         "conname": "",
17         "contel": ""
18       }
19     ],
20     "uri": "address"
21   }
22 }
```

4.1.2.7 获取个人信息

URL: /shop/information

Method: GET

请求参数: 通用参数

返回实例:

```

{
  "code": 100,
  "msg": "请求成功",
  "info": {
    "user": {
      "userid": 8,
      "username": "1520619515",
      "password": "c613779f79361ad4d8489381d45100f4",
      "regtime": "2020-05-17T14:36:41.000+0000",
      "email": "1520619515@qq.com",
      "telephone": "456"
    },
    "uri": "information"
  }
}

```

4.1.2.8 修改信息

URL: /shop/saveInfo

Method: POST

请求参数:

	KEY	VALUE
<input checked="" type="checkbox"/>	name	陈旭扬
<input checked="" type="checkbox"/>	email	123456789@qq.com
<input checked="" type="checkbox"/>	telephone	123456

返回实例:

```

4 {
5   "code": 100,
6   "msg": "更新成功",
7   "info": {
8     }
9 }
10

```

4.1.2.9 添加收货地址

URL: /shop/saveAddr

请求方法: Post

请求参数:

	KEY	VALUE	DESCRIPTION	...	Bulk Ed
<input checked="" type="checkbox"/>	province	山西	省份		
<input checked="" type="checkbox"/>	city	吕梁	市		
<input checked="" type="checkbox"/>	county	离石区	县		
<input checked="" type="checkbox"/>	detailaddr	xxxxxxx	收货地址		
<input checked="" type="checkbox"/>	conname	xxx	姓名		
<input checked="" type="checkbox"/>	contel	1597894655	电话号码		

返回实例:

```
// http://localhost:8080/shop/saveInfo?name=%E3
{
  "code": 100,
  "msg": "更新成功",
  "info": {
  }
}
```

4.1.2.10 删除收货地址

URL: /shop/deleteAddr

请求方法: Post

请求参数:

	KEY	VALUE	DESCRIPTION	...
<input checked="" type="checkbox"/>	addressid	1	地址ID	••

返回实例:

```
{
  "code": 100,
  "msg": "删除成功",
  "info": {
  }
}
```

4.1.2.11 添加地址

URL: /shop/insertAddr

请求方法: Post

请求参数:

	KEY	VALUE	DESCRIPTION	...	Bulk Ed
<input checked="" type="checkbox"/>	province	山西	省份		
<input checked="" type="checkbox"/>	city	吕梁	市		
<input checked="" type="checkbox"/>	county	离石区	县		
<input checked="" type="checkbox"/>	detailaddr	xxxxxxx	收货地址		
<input checked="" type="checkbox"/>	conname	xxx	姓名		
<input checked="" type="checkbox"/>	contel	1597894655	电话号码		

返回实例:

```
2 // http://localhost:8080/shop/saveInfo?name=%E3
3
4 {
5   "code": 100,
6   "msg": "更新成功",
7   "info": {
8
9   }
10 }
```

4.1.2.12 获取订单详情

URL: /shop/info/list

Method: GET

请求参数: 通用参数

返回实例:

```

4  {
5      "code": 100,
6      "msg": "请求成功",
7      "info": {
8          "orderList": [
9              {
10                 "orderid": 35,
11                 "userid": 8,
12                 "ordertime": "2020-05-23T11:30:21.000+0000",
13                 "oldprice": 60.0,
14                 "newprice": 60.0,
15                 "ispay": false,
16                 "issend": false,
17                 "isreceive": false,
18                 "iscomplete": false,
19                 "addressid": 14,
20                 "goodsInfo": [
21                     {
22                         "goodsid": 97,
23                         "goodsname": "golong",
24                         "price": 60,
25                         "num": 1,
26                         "uptime": "2019-05-13T02:22:58.000+0000",
27                         "category": 1,
28                         "detailcate": "测试",
29                         "activityid": 1,
30                         "description": "这是测试商品",
31                         "imagePaths": null,

```

4.1.2.13 获取喜欢商品

URL: </shop/info/favorite>

Method: GET

请求参数: 通用参数

返回实例:

```

{
  "code": 100,
  "msg": "请求成功",
  "info": {
    "pageInfo": {
      "total": 0,
      "list": [

      ],
      "pageNum": 1,
      "pageSize": 0,
      "size": 0,
      "startRow": 0,
      "endRow": 0,
      "pages": 0,
      "prePage": 0,
      "nextPage": 0,
      "isFirstPage": true,
      "isLastPage": true,
      "hasPreviousPage": false,
      "hasNextPage": false,
      "navigatePages": 5,
      "navigatepageNums": [

      ],
      "navigateFirstPage": 0,
      "navigateLastPage": 0,
      "firstPage": 0,
      "lastPage": 0
    },
    "uri": "favorite"
  }
}

```

4.1.2.14 注销用户

URL: /shop/logout

Method: GET

请求参数: 通用参数

返回实例:

```

1 {
2   "code": 100,
3   "msg": "请求成功",
4   "info": {
5     "reason": "注销",
6     "uri": "login"
7   }
8 }

```

4.1.2.15 修改密码

URL: /shop/savePsw

Method: POST

请求参数

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	Psw	123456789	新密码

返回实例

```
{
  "code": 100,
  "msg": "修改密码成功",
  "info": {
  }
}
```

4.1.2.16 添加购物车

URL:/addCart

Method:POST

入参: ShopCart

属性	说明	类型	字段约束	字段类型
userId	用户编号	Int(自增)	Not null	PK、FK
goodsid	商品编号	Int(50)	Not null	FK
cateDate	日期	datetime	Not null	
goodsNum	商品数量	Int	Not null	

返回实例:

```
1 {
2   "code": 200,
3   "msg": "添加失败，请再次登录",
4   "info": {
5     "请求路径": "http://localhost:8081/shop/login"
6   }
7 }
```



```

1  {
2    "code": 100,
3    "msg": "添加成功",
4    "info": {
5      "请求路径": "http://localhost:8081/shop/shopcart"
6    }
7  }

```

4.1.2.17 展示购物车

URL:/showcart

Method:GET

入参: 通用参数

返回实例:

```

1  {
2    "code": 200,
3    "msg": "展示失败，请再次登录",
4    "info": {
5      "请求路径": "http://localhost:8081/shop/login"
6    }
7  }

```

```

1  {
2    "code": 100,
3    "msg": "展示成功",
4    "info": {
5      "请求路径": "http://localhost:8081/shop/shopcart"
6    }
7  }

```

4.1.2.18 购物车数据

URL:/cartjson

Method:GET

入参: 通用参数

返回示例

```

1  {
2    "code": 200,
3    "msg": "请先登录",
4    "info": {}
5  }

```

```

1  {
2    "code": 100,
3    "msg": "查询成功",
4    "info": {
5      "shopcart": []
6    }
7  }

```

4.1.2.19 删除购物车

URL:/deleteCart/{goodsid}

Method:DELETE

入参: Integer : goodsid

返回示例

```
{
  "code": 200,
  "msg": "请先登录",
  "info": {
    "请求路径": "http://localhost:8081/shop/login"
  }
}
```

```
{
  "code": 100,
  "msg": "删除成功",
  "info": {
    "请求路径": "http://localhost:8081/shop/shopcart"
  }
}
```

4.1.2.20 更新购物车

URL:/update

Method:GET

入参: Integer goodsid,Integer num

返回示例

若未登录:

```
1 {
2   "code": 200,
3   "msg": "请先登录",
4   "info": {
5     "请求路径": "http://localhost:8081/shop/login"
6   }
7 }
```

```
1 {
2   "code": 100,
3   "msg": "更新购物车成功",
4   "info": {
5     "请求路径": "http://localhost:8081/shop/shopcart"
6   }
7 }
```

4.1.2.21 查看详细商品信息

URL:/detail

Method:GET

入参: Integer goodsid

返回示例

```
1 {  
2     "code": 200,  
3     "msg": "该商品不存在",  
4     "info": {  
5         "请求路径": "http://localhost:8081/shop/main"  
6     }  
7 }
```

```
{  
  "code":200,  
  "msg":"查询成功,跳转到详细信息界面",  
  "info":{  
    "commentList":[  
      {  
        "commentid":2,  
        "userid":3,  
        "goodsid":93,  
        "point":4,  
        "content":"商品特别好,我非常喜欢",  
        "commenttime":"2019-05-13T02:48:51.000+0000",  
        "username":"admin"  
      }  
    ],  
    "请求路径":"http://localhost:8081/shop/detail",  
    "goodsInfo":{  
      "cate":{  
        "cateid":1,  
        "catename":"数码"  
      },  
      "image":[  
        {  
          "pathid":114,  
          "goodid":93,  
          "path":"3a25测试商品1新公司录用offer.png"  
        },  
        {  
          "pathid":115,  
          "goodid":93,  
          "path":"6a7b测试商品1原公司解约通知.png"  
        }  
      ],  
      "goods":{  
        "goodsid":93,  
        "goodsname":"测试商品1",  
        "price":10,  
        "num":29,  
        "uptime":"2019-05-10T11:43:04.000+0000",  
        "category":1,  
        "detailcate":"测试",  
        "activityid":1,  
        "description":"这是测试商品1",  
        "imagePaths":null,  
        "fav":false,  
      }  
    }  
  }  
}
```

```
        "activity":{
            "activityid":1,
            "activityname":"测试活动",
            "activitydes":"1231",
            "discount":1,
            "fullprice":123,
            "reduceprice":123,
            "fullnum":1231,
            "reducenum":12
        },
        "newPrice":null
    }
}
}
```

4.1.2.22 搜索商品

URL:/search

Method:GET

入参: Integer page, String keyword

<input checked="" type="checkbox"/>	page	1
<input checked="" type="checkbox"/>	keyword	demoData

返回示例

```
{
  "code":100,
  "msg":"查询成功",
  "info":{
    "请求路径":"http://localhost:8081/shop/search",
    "pageInfo":{
      "total":0,
      "list":[
      ],
      "pageNum":0,
      "pageSize":16,
      "size":0,
      "startRow":0,
      "endRow":0,
      "pages":0,
      "prePage":0,
      "nextPage":0,
      "isFirstPage":false,
      "isLastPage":true,
      "hasPreviousPage":false,
      "hasNextPage":false,
      "navigatePages":5,
      "navigatepageNums":[
      ],
      "navigateFirstPage":0,
      "navigateLastPage":0,
      "lastPage":0,
      "firstPage":0
    },
    "keyword":"demoData"
  }
}
```

4.1.2.22 收藏商品

URL:/collect
Method:GET
入参: Integer goodsid

	KEY	VALUE
<input checked="" type="checkbox"/>	goodsid	94

返回示例

```
␣{  
  "code":200,  
  "msg":"收藏失败",  
  "info":␣{  
    "请求路径":"http://localhost:8081/shop/collect"  
  }  
}  
  
␣{  
  "code":100,  
  "msg":"收藏成功",  
  "info":␣{  
    "请求路径":"http://localhost:8081/shop/collect",  
    "favorite":␣{  
      "userid":1,  
      "goodsid":94,  
      "collecttime":"2020-05-23T01:50:59.625+0000"  
    }  
  }  
}
```

4.1.2.23 取消收藏

URL:/deleteCollect
Method:GET
入参: Integer goodsid

	KEY	VALUE
<input checked="" type="checkbox"/>	goodsid	93

返回示例

```
⊖{
  "code":200,
  "msg":"取消收藏失败",
  "info":⊖{
    "请求路径":"http://localhost:8081/shop/collect"
  }
}

⊖{
  "code":100,
  "msg":"取消收藏成功",
  "info":⊖{
    "请求路径":"http://localhost:8081/shop/collect"
  }
}
```

4.1.2.24 商品种类分类查询

URL:/category

Method:GET

入参: String cate， Integer page

	KEY	VALUE
<input checked="" type="checkbox"/>	cate	demoData
<input checked="" type="checkbox"/>	page	1

返回示例

```
⊖{
  "code":100,
  "msg":"查询成功",
  "info":⊖{
    "cate":"demoData",
    "请求路径":"http://localhost:8081/shop/category",
    "pageInfo":⊖{
      "total":0,
      "list":⊖[

    ],
    "pageNum":1,
    "pageSize":0,
    "size":0,
    "startRow":0,
    "endRow":0,
    "pages":0,
    "prePage":0,
    "nextPage":0,
    "isFirstPage":true,
    "isLastPage":true,
    "hasPreviousPage":false,
    "hasNextPage":false,
    "navigatePages":5,
    "navigatepageNums":⊖[

    ],
    "navigateFirstPage":0,
    "navigateLastPage":0,
    "firstPage":0,
    "lastPage":0
  }
}
```

4.1.2.25 评论

URL:/comment

Method:POST

入参: Comment comment

commentid	2
userid	1
goodsid	93
point	4

content	demoData
commenttime	2020/5/23 10:18
username	demoData

返回示例

```
{
  "code": 200,
  "msg": "评论失败",
  "info": {
    "请求路径": "http://localhost:8081/shop/comment"
  }
}
```

```
{
  "code": 200,
  "msg": "评论失败",
  "info": {
    "请求路径": "http://localhost:8081/shop/comment"
  }
}
```

4.1.2.26 展示订单

URL:/order

Method:GET

入参: 通用参数

返回示例

```
{
  "code": 200,
  "msg": "用户不存在",
  "info": {
    "请求路径": "http://localhost:8081/shop/login"
  }
}
```



```
{
  "code":100,
  "msg":"展示订单成功",
  "info":{
    "address":{
      "addressid":7,
      "userid":1,
      "province":"北京市",
      "city":"北京市市辖区",
      "county":"东城区",
      "detailaddr":"12312",
      "conname":"123123",
      "contel":"2131231321"
    }
  ],
  "totalPrice":120,
  "oldTotalPrice":120,
  "goodsAndImage":{
    "goodsid":93,
    "goodsname":"测试商品1",
    "price":10,
    "num":12,
    "uptime":"2019-05-10T11:43:04.000+0000",
    "category":1,
    "detailcate":"测试",
    "activityid":1,
    "description":"这是测试商品1",
    "imagePaths":{
      {
        "pathid":114,
        "goodid":93,
        "path":"3a25测试商品1新公司录用offer.png"
      },
      {
        "pathid":115,
        "goodid":93,
        "path":"6a7b测试商品1原公司解约通知.png"
      }
    }
  ],
  "fav":false,
  "activity":{
    "activityid":1,
    "activityname":"测试活动",
    "activitydes":"1231",
    "discount":1,
    "fullprice":123,
    "reduceprice":123,
    "fullnum":1231,
    "reducenum":12
  },
  "newPrice":120
}
  ],
  "请求路径":"http://localhost:8081/shop/order"
}
```

4.1.2.27 购买

URL:/orderFinish

Method:GET

入参:

Query Params		
	KEY	VALUE
<input checked="" type="checkbox"/>	oldPrice	500
<input checked="" type="checkbox"/>	newPrice	200
<input checked="" type="checkbox"/>	isPay	true
<input checked="" type="checkbox"/>	addressid	7

返回示例

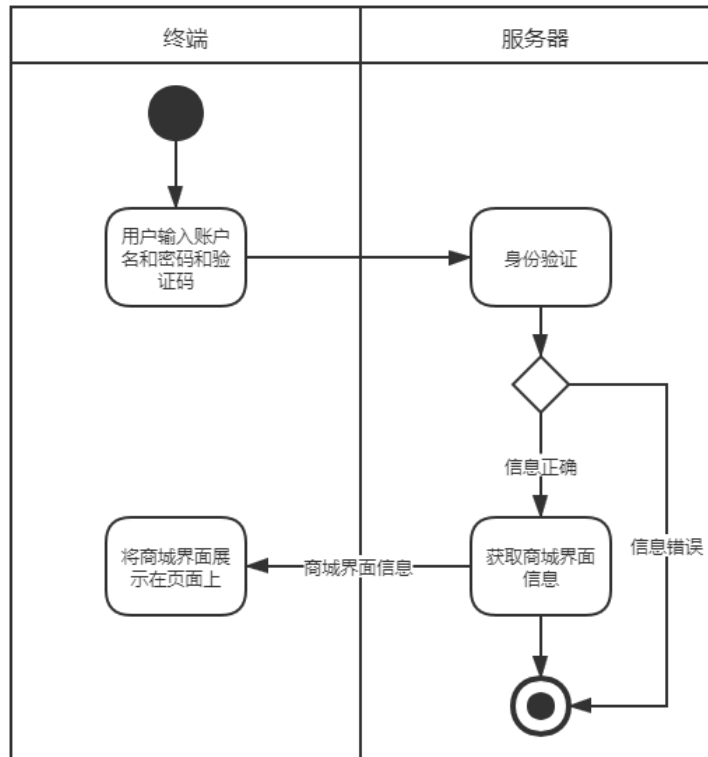
```
{
  "code":100,
  "msg":"购买成功",
  "info":{
    "请求路径":"http://localhost:8081/shop/order"
  }
}
```

4.2 模块设计

登录

功能描述：用户使用账号和密码进行系统登录

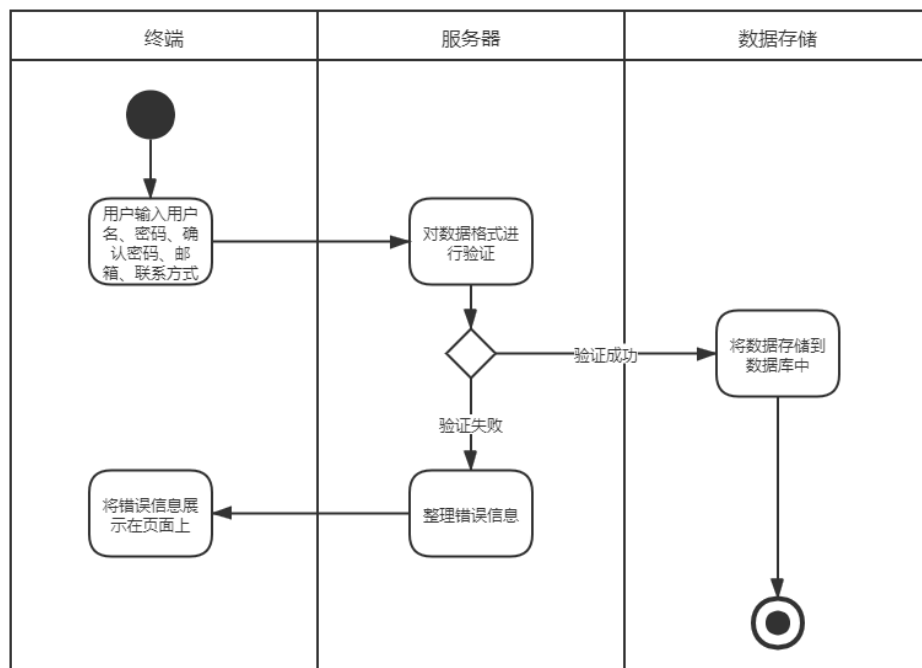
程序逻辑：



注册

功能描述：用户使用用户名、密码、确认密码、邮箱、联系方式进行注册

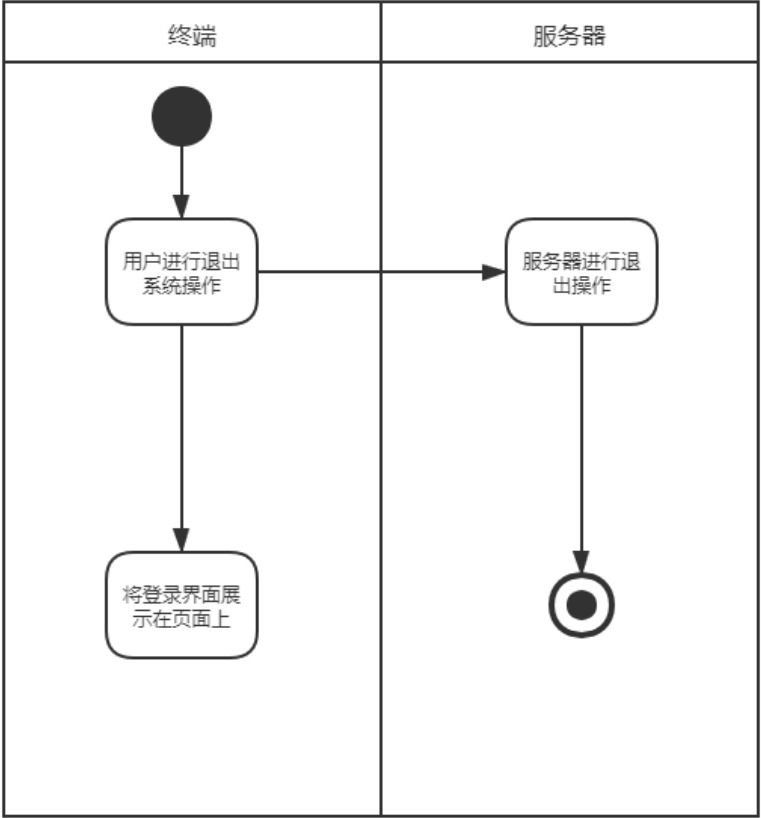
程序逻辑：



退出登录

功能描述：用户进行退出系统

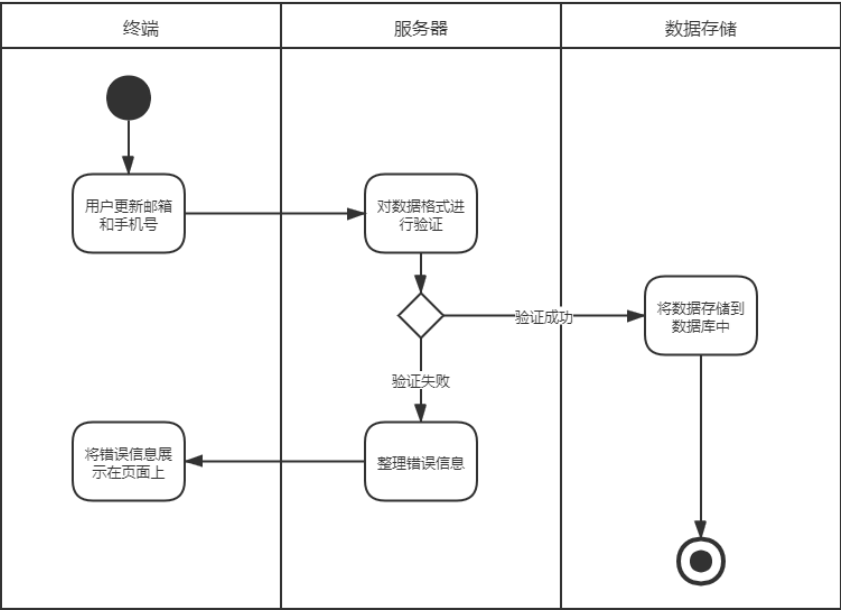
程序逻辑：



修改信息

功能描述：用户更新邮箱和手机号来修改信息

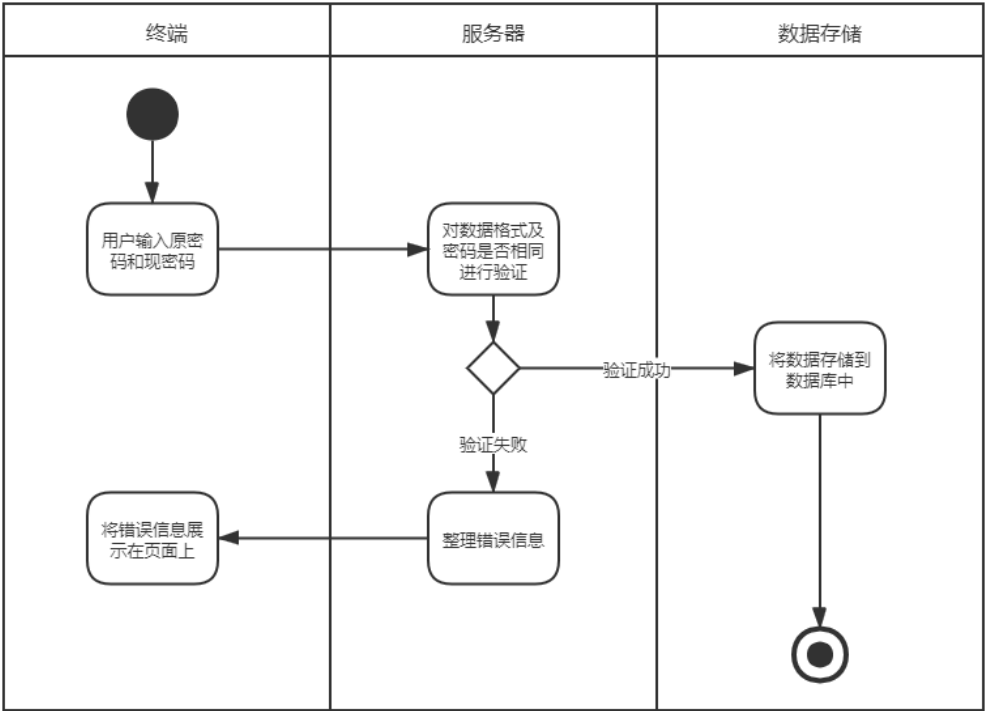
程序逻辑：



修改密码

功能描述：用户更新密码来修改密码

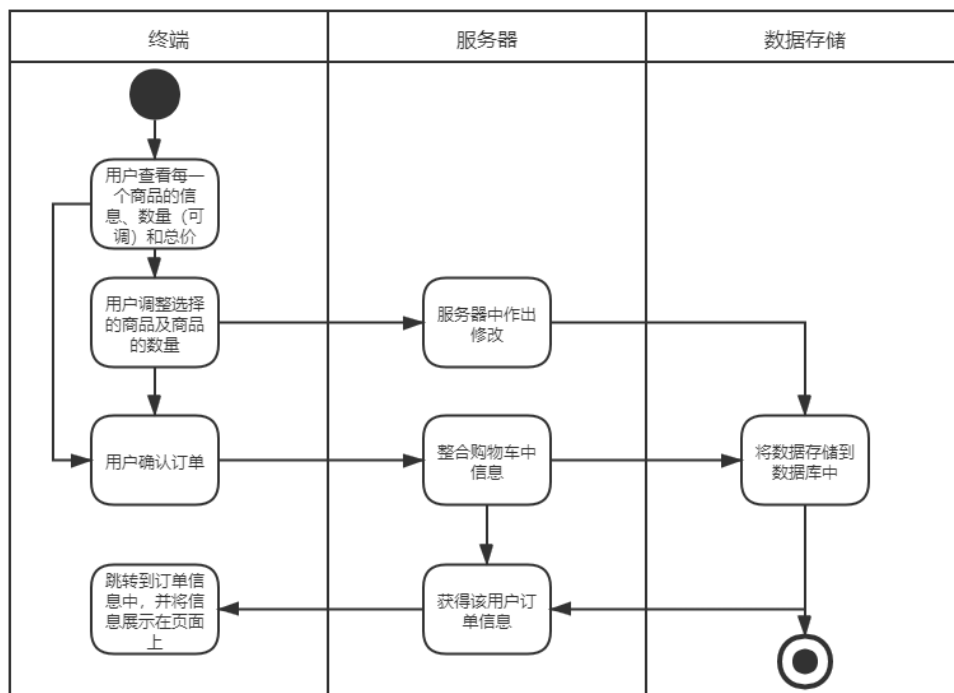
程序逻辑：



购物车管理

功能描述：用户通过购物车管理查看每一个商品的图片，名称，价格，数量（可调）总价以及所有商品的总计。

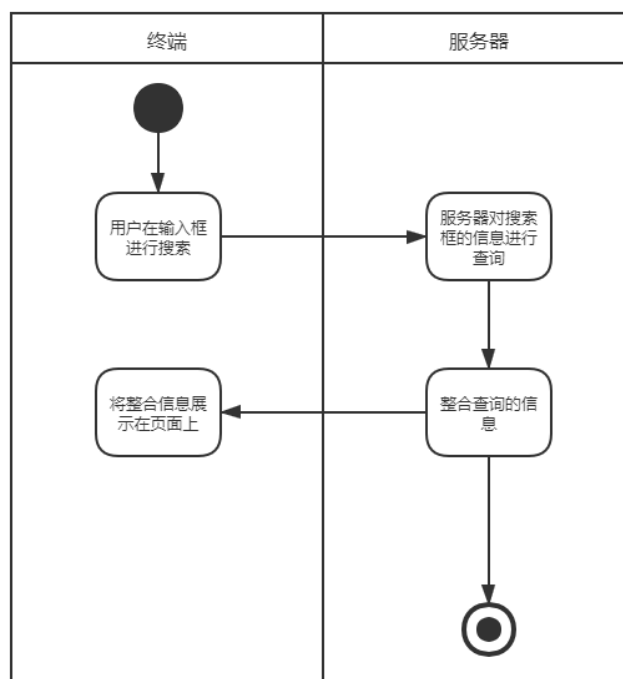
程序逻辑：



商品检索

功能描述：用户在输入框进行搜索，查看对应商品的图片、名称、价格、记录个数

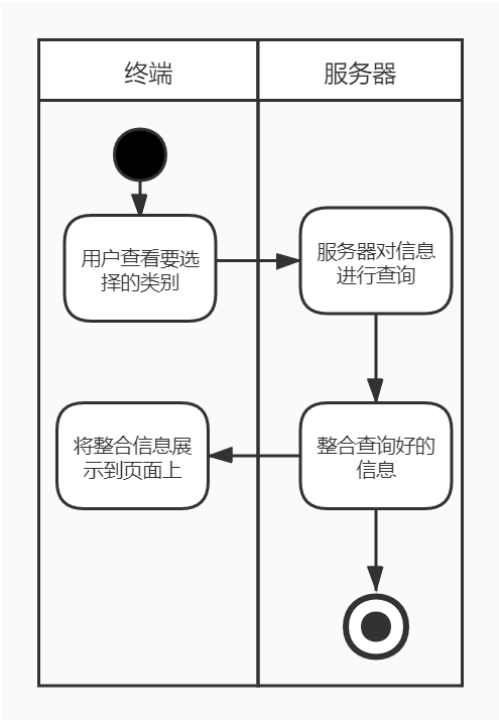
程序逻辑：



商品浏览

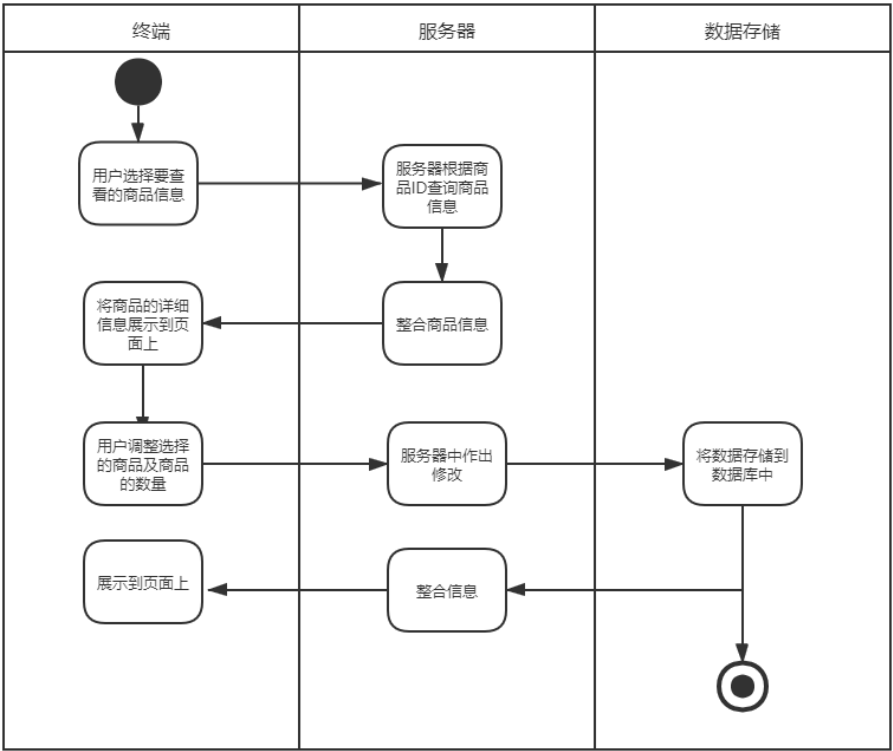
功能描述：用户可以进行商品浏览，用户可以进入每一个单独分类的列表查看商品。用户可

以对每一个商品进行收藏，同时可以查看每一个商品的详细信息。
程序逻辑：



查看商品详细信息

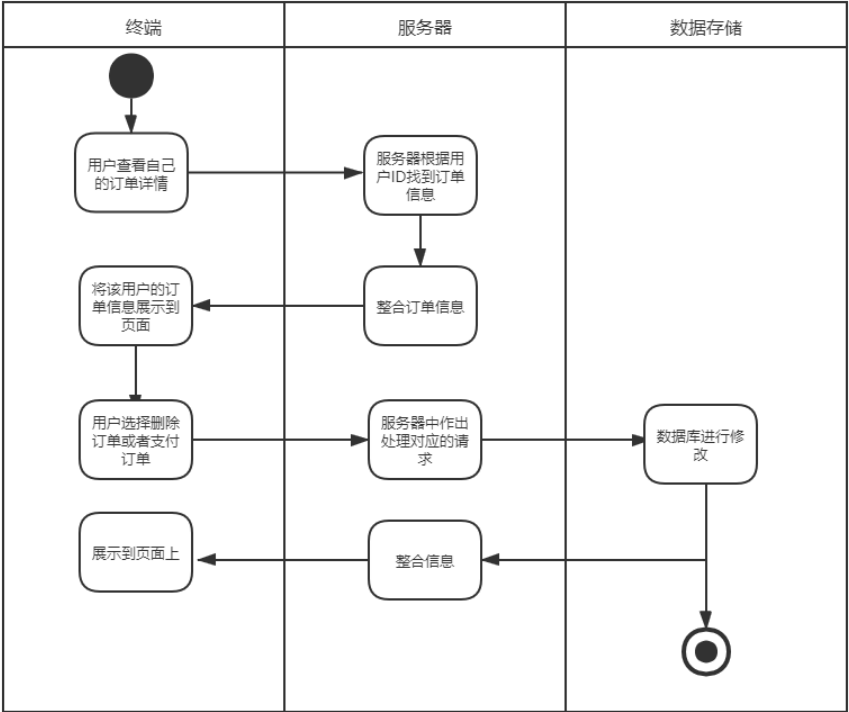
功能描述：用户查看商品的详细信息已经用户可以进行的各类操作
程序逻辑：



订单管理

功能描述：用户对自己的订单进行管理

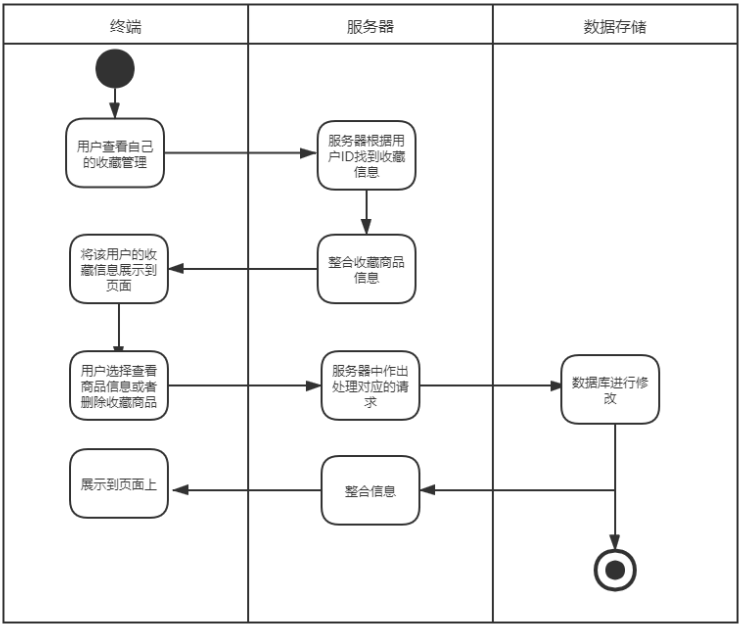
程序逻辑：



个人收藏管理

功能描述：收藏中存放着用户收藏的商品列表，可以查看详细信息，可以删除该收藏

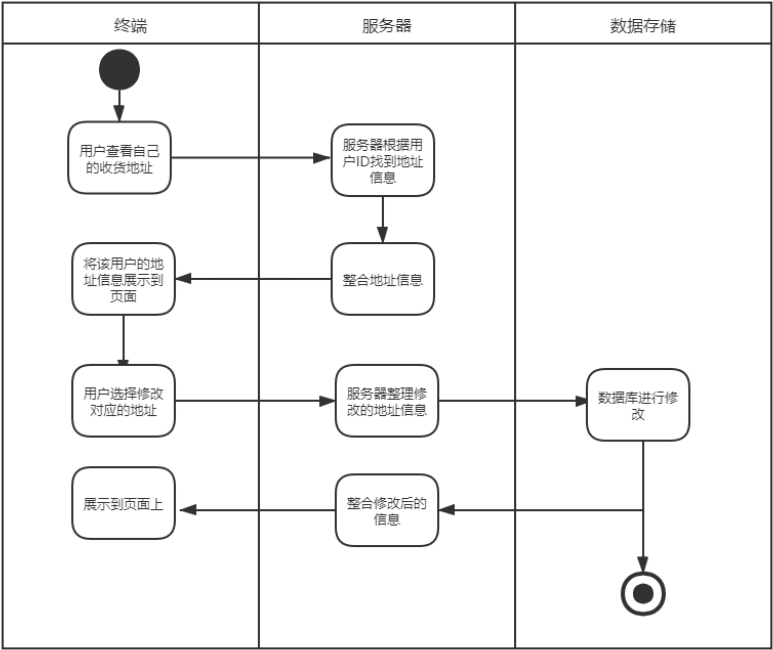
程序逻辑：



修改地址

功能描述：用户可以查看收货地址，其中含有用户创建的收货地址（详细信息）用户可以更新自己的收货地址，对之前的收货地址进行修改。

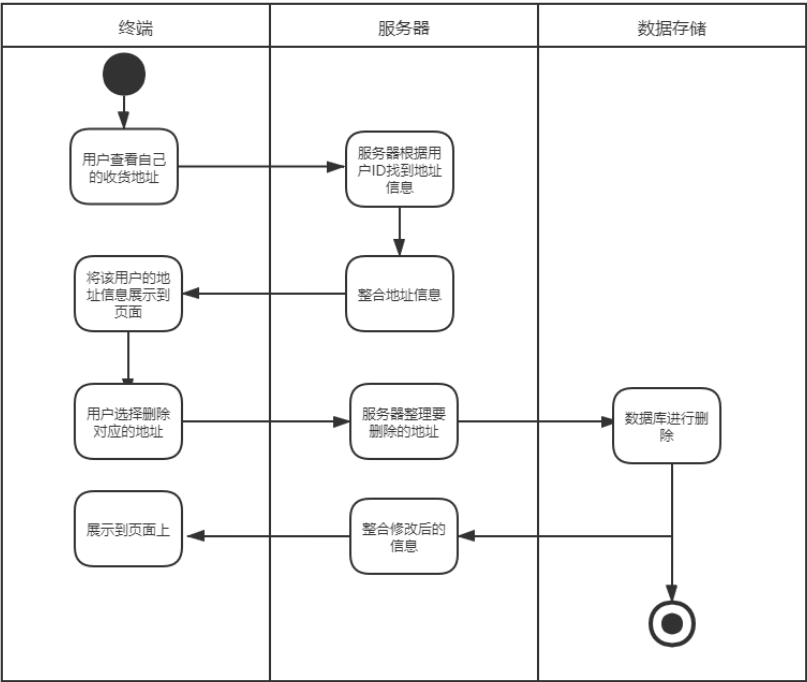
程序流程：



删除地址

功能描述：用户在查看完自己的收货地址后，可以删除想要删除的地址，

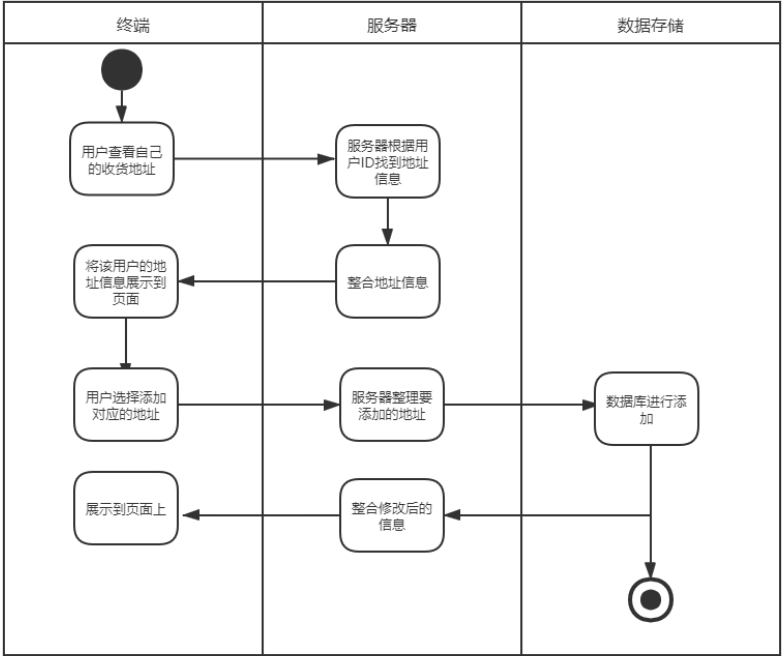
程序逻辑：



添加地址

功能描述：用户在查看完自己的收货地址后，可以添加想要添加的地址

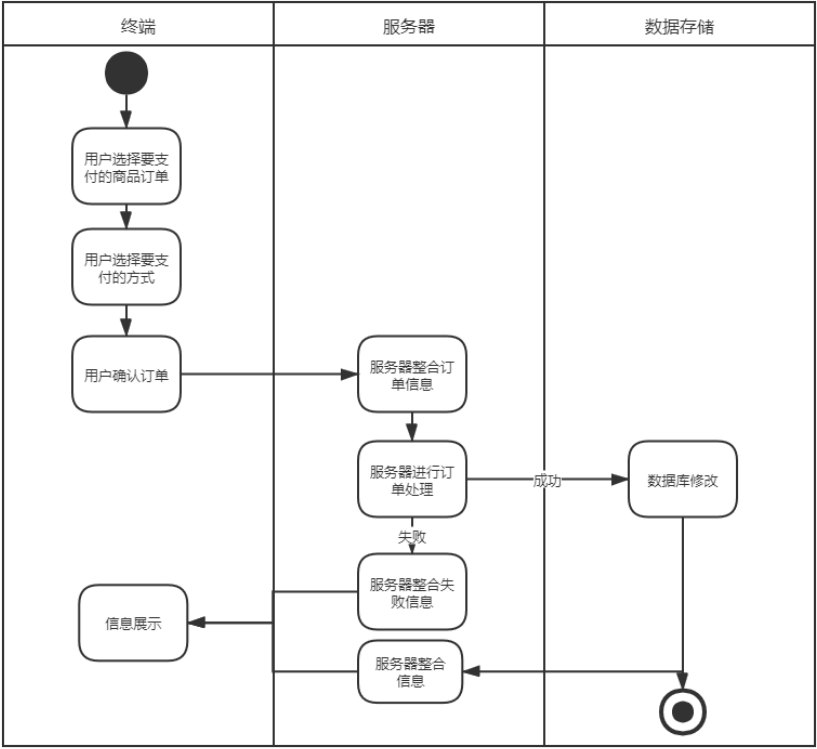
程序流程：



在线支付

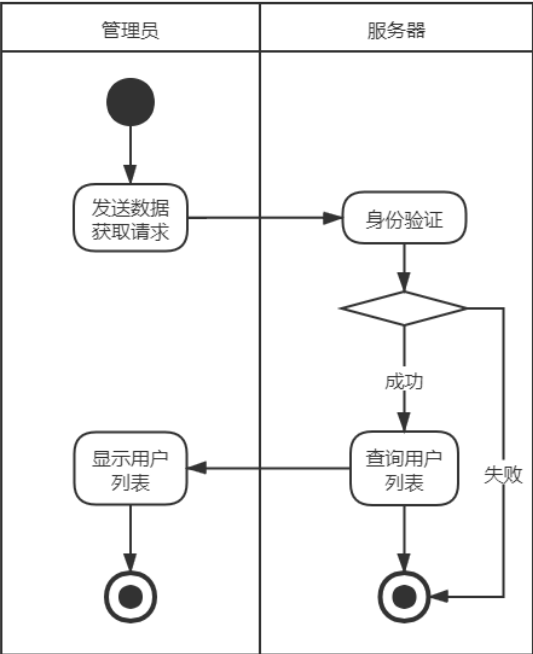
功能描述：用户在确认订单后可以支付，选择自己的支付方式

程序流程：



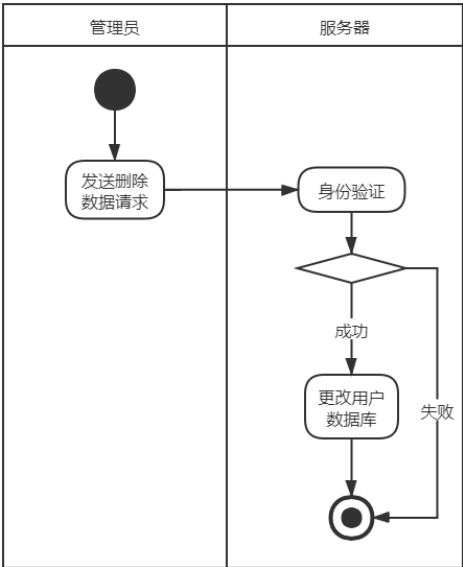
查看用户列表

功能描述：管理员可以查看用户列表，在此基础上做下一步工作。
程序流程：



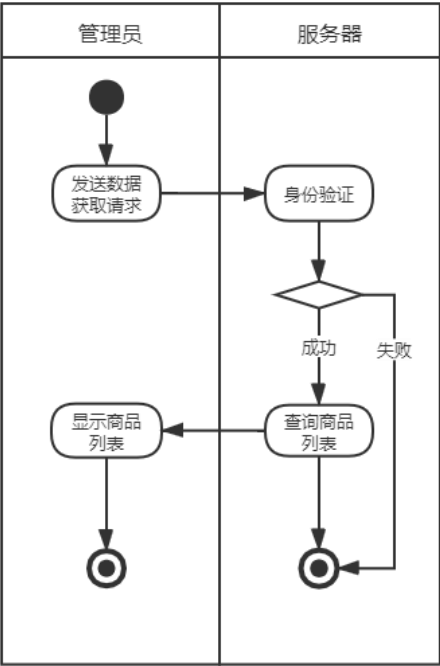
删除用户

功能描述：管理员找到想到删除的用户，能成功删除该用户。
程序流程：



查看商品列表

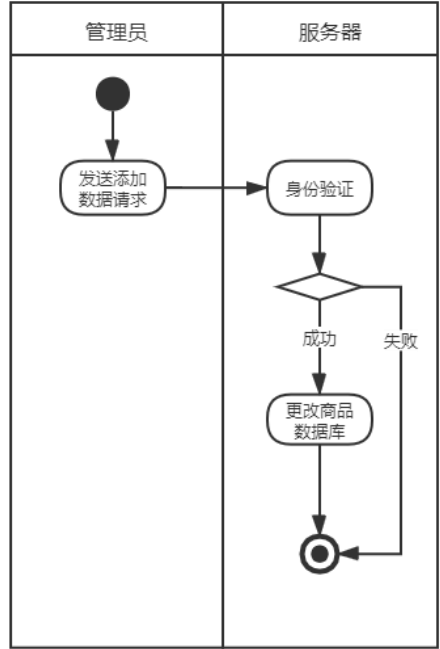
功能描述：管理员可以查看商品列表，在此基础上做下一步工作。
程序流程：



添加商品

功能描述：管理员录入新添加商品的信息。

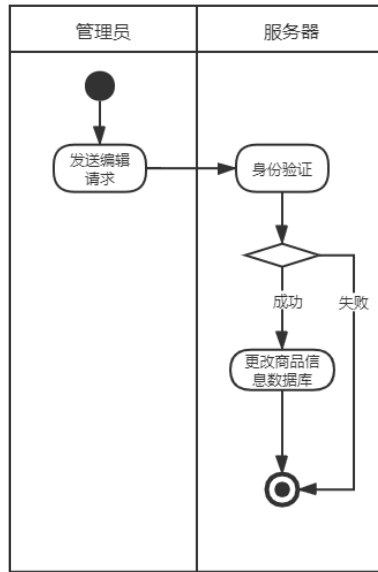
程序流程：



编辑商品信息

功能描述：管理员可以对已经上架的商品信息进行更改。

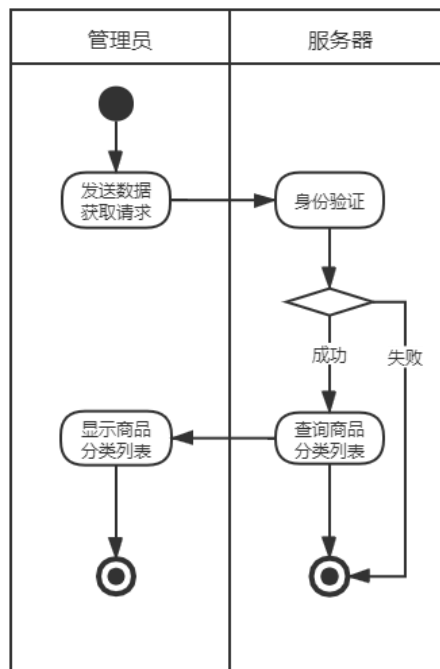
程序流程：



查看商品分类

功能描述：实现管理员查看系统中的商品分类的相关信息。

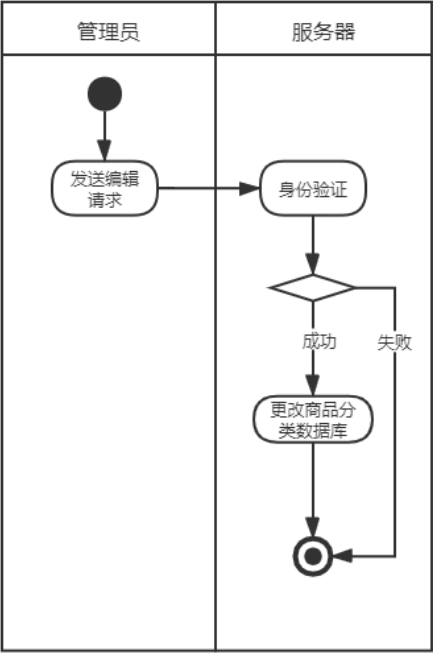
程序流程：



编辑商品分类

功能描述：管理员可以随时编辑商品分类名。

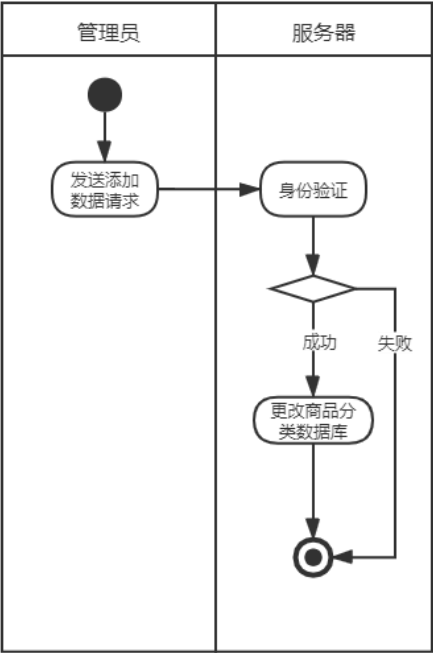
程序流程：



添加商品分类

功能描述：管理员可以随时添加商品分类名。

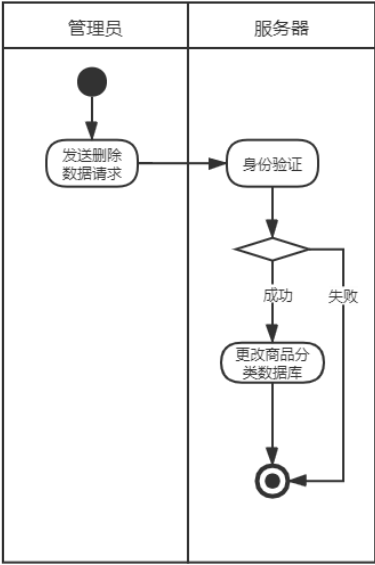
程序流程：



删除商品分类

功能描述：管理员可以随时删除商品分类。

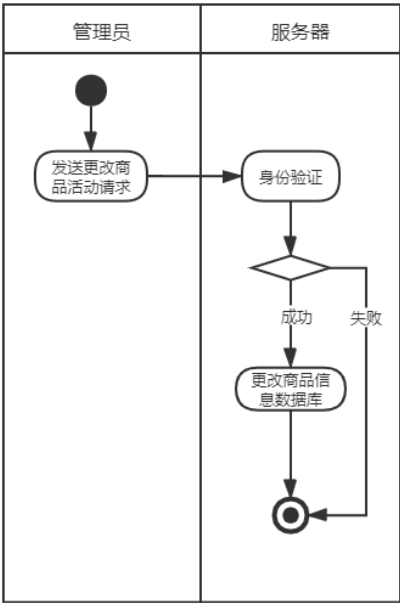
程序流程：



管理商品活动

功能描述：管理员可以管理商品所参加的活动。

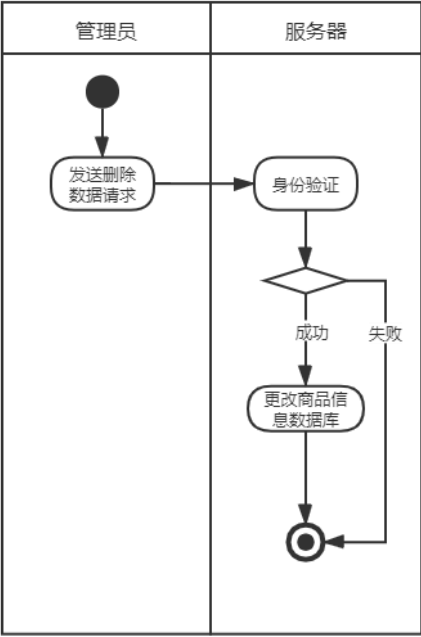
程序流程：



删除商品

功能描述：管理员可以删除已经存在的商品。

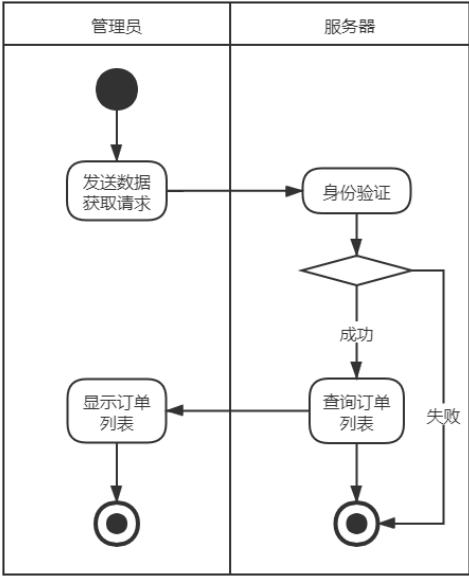
程序流程：



查看订单

功能描述：实现管理员查看系统中的订单的相关信息。

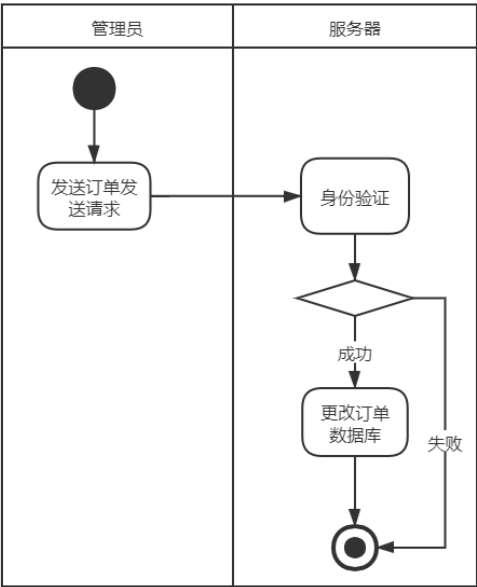
程序流程：



订单发送

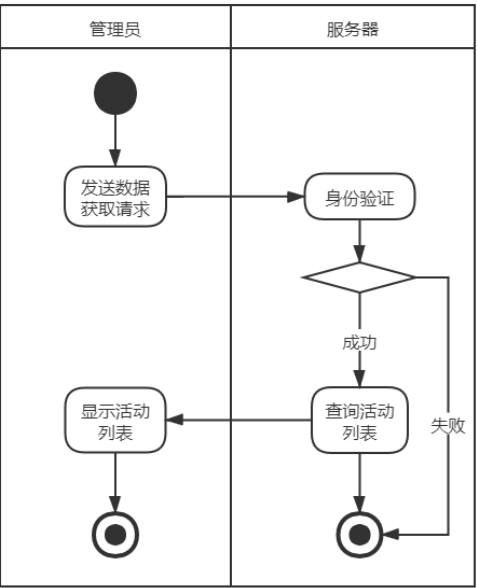
功能描述：对于未发货状态的订单，管理员可以进行发货操作。

程序流程：



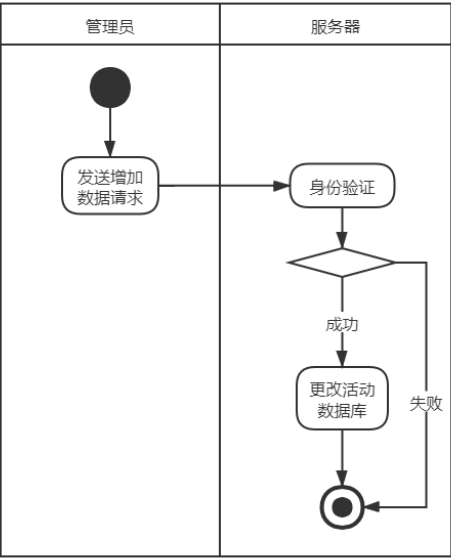
查看活动

功能描述：实现管理员查看系统中的活动的相关信息，以便进行下面的工作。
程序流程：



增加活动

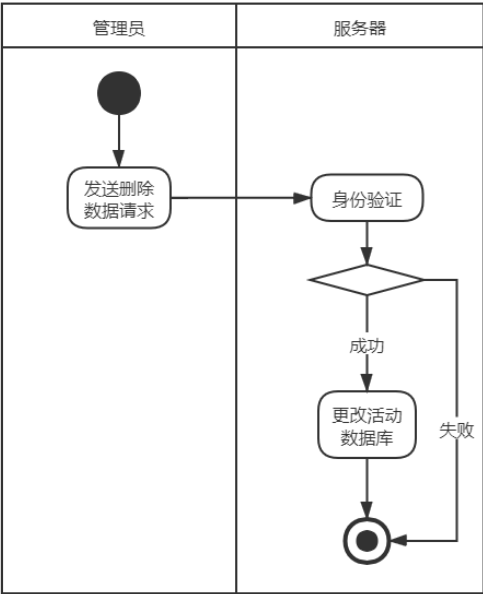
功能描述：实现管理员添加活动。
程序流程：



删除活动

功能描述：管理员可以删除已经存在的活动。

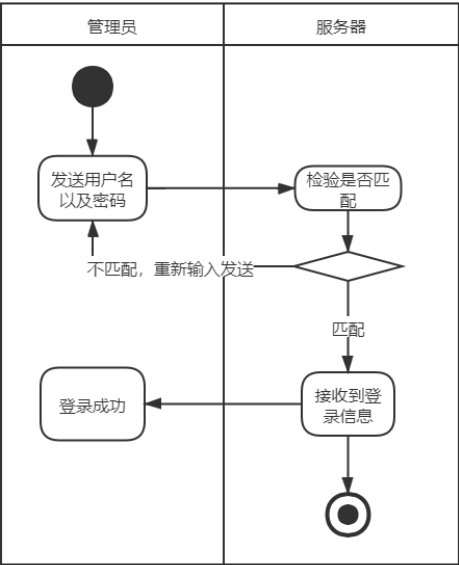
程序流程：



登录

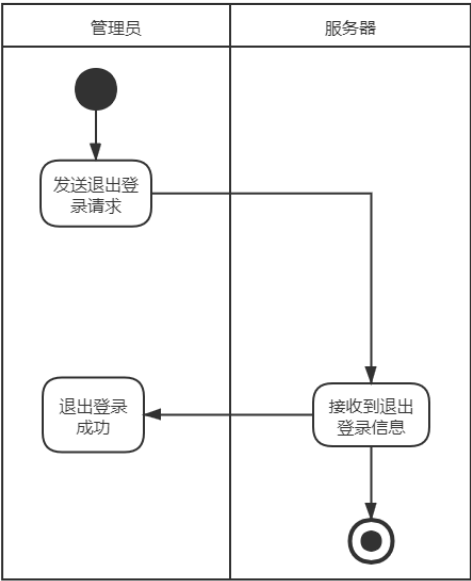
功能描述：管理员可以登录到管理员系统对商城中的用户、订单、商品以及活动信息进行查看以及其他操作。

程序流程：



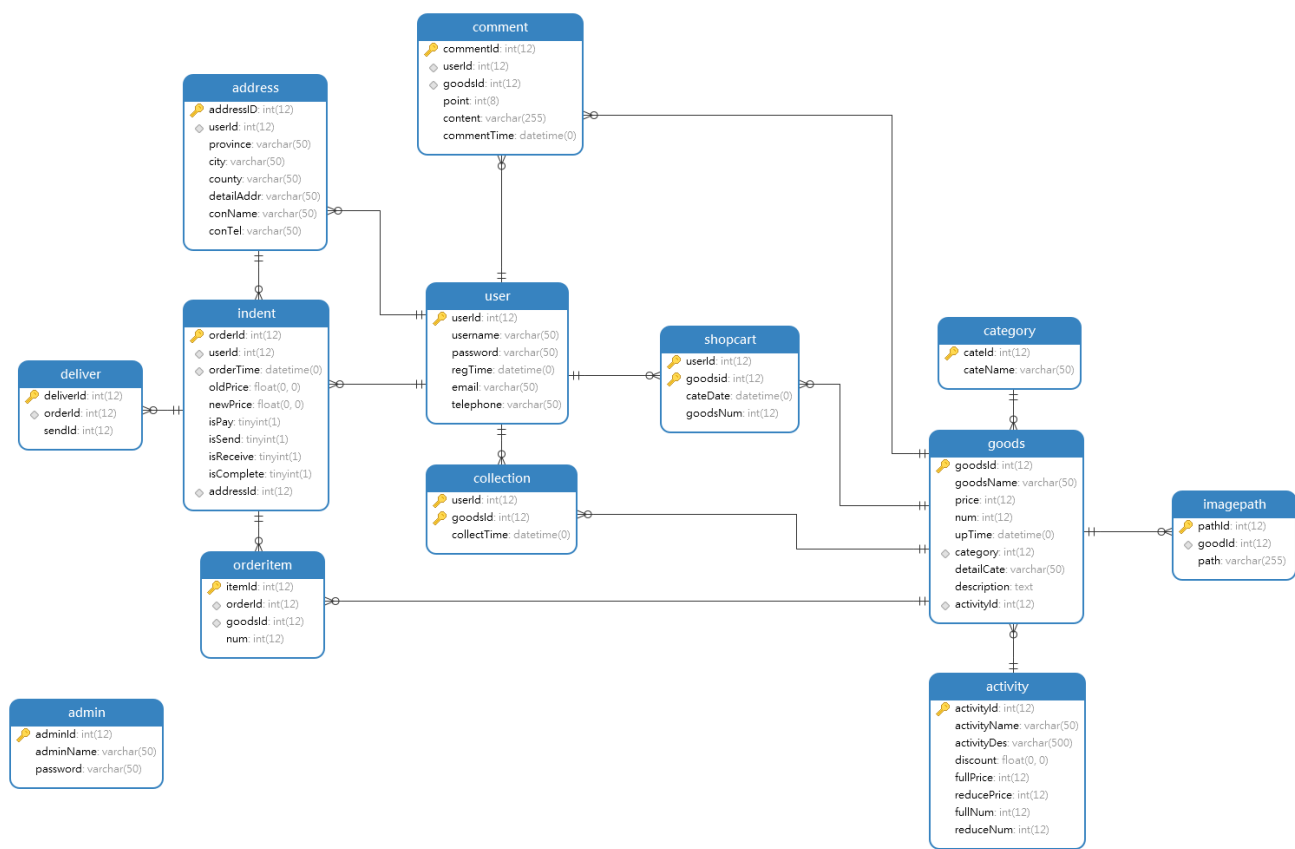
退出登录

功能描述：实现管理员退出系统。



5.数据库设计

5.1 E-R 图



5.2 数据库建表

5.2.1 活动表 activity

5.2.1.1 数据项说明

属性	说明	类型	字段约束	字段类型
activityId	活动编号	Int(自增)	Not null	PK
activityName	活动名称	varchar(50)	Not null	
activityDes	活动宣传语	varchar(500)	Not null	
discount	折扣	float	Default 1	
fullPrice	最高价	int(12)	Default null	
reducePrice	最低价	int(12)	Default null	

fullNum	最多数量	int(12)	Default null	
reduceNum	最低数量	int(12)	Default null	

5.2.1.2 建表 SQL 语句

```
CREATE TABLE `activity` (
  `activityId` int(12) NOT NULL AUTO_INCREMENT,
  `activityName` varchar(50) NOT NULL,
  `activityDes` varchar(500) NOT NULL,
  `discount` float DEFAULT '1',
  `fullPrice` int(12) DEFAULT NULL,
  `reducePrice` int(12) DEFAULT NULL,
  `fullNum` int(12) DEFAULT NULL,
  `reduceNum` int(12) DEFAULT NULL,
  PRIMARY KEY (`activityId`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

5.2.2 地址表 address

5.2.2.1 数据项说明

属性	说明	类型	字段约束	字段类型
addressID	地址编号	Int(自增)	Not null	PK
userId	用户编号	int(12)	Not null	FK
province	省份	varchar(50)	Not null	
city	城市	varchar(50)	Not null	
county	区县	varchar(50)	Not null	
detailAddr	详细地址	varchar(50)	Not null	
conName	收件人名字	varchar(50)	Not null	
conTel	联系电话	varchar(50)	Not null	

5.2.2.2 建表 SQL 语句

```
CREATE TABLE `address` (
  `addressID` int(12) NOT NULL AUTO_INCREMENT,
  `userId` int(12) NOT NULL,
  `province` varchar(50) NOT NULL,
  `city` varchar(50) NOT NULL,
  `county` varchar(50) NOT NULL,
```

```

`detailAddr` varchar(50) NOT NULL,
`conName` varchar(50) NOT NULL,
`conTel` varchar(50) NOT NULL,
PRIMARY KEY (`addressID`),
KEY `addressID` (`addressID`),
KEY `userId` (`userId`),
CONSTRAINT `address_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `user` (`userId`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8;

```

5.2.3 管理员表 admin

5.2.3.1 数据项说明

属性	说明	类型	字段约束	字段类型
adminId	管理员编号	Int(自增)	Not null	PK
adminName	管理员名字	varchar(50)	Not null	
password	密码	varchar(50)	Not null	

5.2.3.2 建表 SQL 语句

```

CREATE TABLE `admin` (
  `adminId` int(12) NOT NULL AUTO_INCREMENT,
  `adminName` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  PRIMARY KEY (`adminId`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

```

5.2.4 商品分类表 category

5.2.4.1 数据项说明

属性	说明	类型	字段约束	字段类型
cateId	分类编号	Int(自增)	Not null	PK
cateName	分类名称	varchar(50)	Not null	

5.2.4.2 建表 SQL 语句

```
CREATE TABLE `category` (  
  `catelId` int(12) NOT NULL AUTO_INCREMENT,  
  `cateName` varchar(50) NOT NULL,  
  PRIMARY KEY (`catelId`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```

5.2.5 订单表 collection

5.2.5.1 数据项说明

属性	说明	类型	字段约束	字段类型
userId	用户编号	Int(12)	Not null	PK、FK
goodsId	商品编号	int(12)	Not null	PK、FK
collectTime	下单时间	datetime	Not null	

5.2.5.2 建表 SQL 语句

```
CREATE TABLE `collection` (  
  `userId` int(12) NOT NULL,  
  `goodsId` int(12) NOT NULL,  
  `collectTime` datetime NOT NULL,  
  PRIMARY KEY (`userId`,`goodsId`),  
  KEY `collection_ibfk_2` (`goodsId`),  
  CONSTRAINT `collection_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `user` (`userId`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `collection_ibfk_2` FOREIGN KEY (`goodsId`) REFERENCES `goods` (`goodsId`)  
ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

5.2.6 评论表 comment

5.2.6.1 数据项说明

属性	说明	类型	字段约束	字段类型
commentId	评论编号	Int(自增)	Not null	PK

userId	用户编号	int(12)	Not null	FK
goodsId	商品编号	int(12)	Not null	FK
point	好评指数	int(8)	Not null	
content	评论内容	varchar(255)	Not null	
commentTime	评论时间	datetime	Not null	

5.2.6.2 建表 SQL 语句

```
CREATE TABLE `comment` (
  `commentId` int(12) NOT NULL AUTO_INCREMENT,
  `userId` int(12) NOT NULL,
  `goodsId` int(12) NOT NULL,
  `point` int(8) NOT NULL,
  `content` varchar(255) NOT NULL,
  `commentTime` datetime NOT NULL,
  PRIMARY KEY (`commentId`),
  KEY `userId` (`userId`),
  KEY `goodsId` (`goodsId`),
  CONSTRAINT `comment_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `user` (`userId`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `comment_ibfk_2` FOREIGN KEY (`goodsId`) REFERENCES `goods` (`goodsId`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

5.2.7 订单传送表 deliver

5.2.7.1 数据项说明

属性	说明	类型	字段约束	字段类型
deliverId	发送编号	Int(自增)	Not null	PK
orderId	订单编号	Int	Not null	
sendId	传送编号	Int	Not null	

5.2.7.2 建表 SQL 语句

```
CREATE TABLE `deliver` (
  `deliverId` int(12) NOT NULL AUTO_INCREMENT,
  `orderId` int(12) NOT NULL,
  `sendId` int(12) NOT NULL,
```



```

PRIMARY KEY (`deliverId`),
KEY `orderId` (`orderId`),
CONSTRAINT `deliver_ibfk_1` FOREIGN KEY (`orderId`) REFERENCES `indent` (`orderId`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

5.2.8 商品表 goods

5.2.8.1 数据项说明

属性	说明	类型	字段约束	字段类型
goodsId	商品编号	Int(自增)	Not null	PK
goodsName	商品名称	varchar	Not null	
price	商品价格	int(20)	Not null	
num	商品数量	int)	Not null	
upTime	上货时间	datetime	Not null	
category	类别	int(12)	Not null	FK
detailCate	细节	varchar(19)	Not null	
description	描述	text	Not null	
activityId	活动号	int	Not null	FK

5.2.8.2 建表 SQL 语句

```

CREATE TABLE `goods` (
  `goodsId` int(12) NOT NULL AUTO_INCREMENT,
  `goodsName` varchar(50) NOT NULL,
  `price` int(12) NOT NULL,
  `num` int(12) NOT NULL,
  `upTime` datetime NOT NULL,
  `category` int(12) NOT NULL,
  `detailCate` varchar(50) NOT NULL,
  `description` text NOT NULL,
  `activityId` int(12) NOT NULL DEFAULT '1',
  PRIMARY KEY (`goodsId`),
  KEY `activityId` (`activityId`),
  KEY `category` (`category`),
  CONSTRAINT `goods_ibfk_1` FOREIGN KEY (`activityId`) REFERENCES `activity` (`activityId`)
ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `goods_ibfk_2` FOREIGN KEY (`category`) REFERENCES `category` (`catId`)
ON DELETE CASCADE ON UPDATE CASCADE

```

) ENGINE=InnoDB AUTO_INCREMENT=98 DEFAULT CHARSET=utf8;

5.2.9 图片路径表 imagepath

5.2.9.1 数据项说明

属性	说明	类型	字段约束	字段类型
pathId	图片编号	Int(自增)	Not null	PK
goodId	商品编号	Int	Not null	
path	图片路径	varchar(255)	Not null	

5.2.9.2 建表 SQL 语句

```
CREATE TABLE `imagepath` (  
  `pathId` int(12) NOT NULL AUTO_INCREMENT,  
  `goodId` int(12) NOT NULL,  
  `path` varchar(255) NOT NULL,  
  PRIMARY KEY (`pathId`),  
  KEY `goodId` (`goodId`),  
  CONSTRAINT `imagepath_ibfk_1` FOREIGN KEY (`goodId`) REFERENCES `goods` (`goodId`)  
  ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=129 DEFAULT CHARSET=utf8;
```

5.2.10 订货单表 indent

5.2.10.1 数据项说明

属性	说明	类型	字段约束	字段类型
orderId	订单编号	Int(自增)	Not null	PK
userId	用户编号	Int	Not null	FK
orderTime	订单时间	datetime	Not null	
oldPrice	原价格	float	Not null	
newPrice	新价格	float	Not null	
isPay	被买	tinyint	Not null	
isSend	被发送	tinyint	Not null	
isReceive	收到	tinyint	Not null	
isComplete	完成	tinyint	Not null	
addressId	地址编号	int	Not null	FK

5.2.10.2 建表 SQL 语句

```
CREATE TABLE `indent` (  
  `orderId` int(12) NOT NULL AUTO_INCREMENT,  
  `userId` int(12) NOT NULL,  
  `orderTime` datetime NOT NULL,  
  `oldPrice` float NOT NULL,  
  `newPrice` float NOT NULL,  
  `isPay` tinyint(1) NOT NULL,  
  `isSend` tinyint(1) NOT NULL,  
  `isReceive` tinyint(1) NOT NULL,  
  `isComplete` tinyint(1) NOT NULL,  
  `addressId` int(12) NOT NULL,  
  PRIMARY KEY (`orderId`),  
  KEY `userId` (`userId`),  
  KEY `orderGoods` (`orderTime`),  
  KEY `addressId` (`addressId`),  
  CONSTRAINT `indent_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `user` (`userId`) ON  
DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `indent_ibfk_2` FOREIGN KEY (`addressId`) REFERENCES `address`  
(`addressID`) ON DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=utf8;
```

5.2.11 订单项目表 orderitem

5.2.11.1 数据项说明

属性	说明	类型	字段约束	字段类型
itemId	项目编号	Int(自增)	Not null	PK
orderId	订单编号	int(12)	Not null	FK
goodsId	商品编号	int(12)	Not null	FK
Num	商品数量	int(12)	Not null	

5.2.11.2 建表 SQL 语句

```
CREATE TABLE `orderitem` (  
  `itemId` int(12) NOT NULL AUTO_INCREMENT,  
  `orderId` int(12) NOT NULL,
```

```

`goodsId` int(12) NOT NULL,
`num` int(12) NOT NULL,
PRIMARY KEY (`itemId`),
KEY `orderId` (`orderId`),
KEY `goodsId` (`goodsId`),
CONSTRAINT `orderitem_ibfk_1` FOREIGN KEY (`orderId`) REFERENCES `indent` (`orderId`)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `orderitem_ibfk_2` FOREIGN KEY (`goodsId`) REFERENCES `goods` (`goodsId`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8;

```

5.2.12 购物车表 shopcart

5.2.12.1 数据项说明

属性	说明	类型	字段约束	字段类型
userId	用户编号	Int(自增)	Not null	PK、FK
goodsid	商品编号	Int	Not null	PK、FK
cateDate	日期	datetime	Not null	
goodsNum	商品数	Int	Not null	

5.2.12.2 建表 SQL 语句

```

CREATE TABLE `shopcart` (
  `userId` int(12) NOT NULL,
  `goodsid` int(12) NOT NULL,
  `cateDate` datetime NOT NULL,
  `goodsNum` int(12) NOT NULL,
  PRIMARY KEY (`userId`,`goodsid`),
  KEY `userId` (`userId`),
  KEY `goodsid` (`goodsid`),
  CONSTRAINT `shopcart_ibfk_1` FOREIGN KEY (`userId`) REFERENCES `user` (`userId`) ON
DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `shopcart_ibfk_2` FOREIGN KEY (`goodsid`) REFERENCES `goods` (`goodsId`)
ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

5.2.13 用户表 user

5.2.13.1 数据项说明

属性	说明	类型	字段约束	字段类型
userId	用户编号	Int(自增)	Not null	PK
username	用户名字	varchar(50)	Not null	
password	密码	varchar(50)	Not null	
regTime	注册时间	datetime	Not null	
email	邮箱	varchar(50)	Not null	
telephone	电话	varchar(50)	Not null	

5.2.13.2 建表 SQL 语句

```
CREATE TABLE `user` (  
  `userId` int(12) NOT NULL AUTO_INCREMENT,  
  `username` varchar(50) NOT NULL,  
  `password` varchar(50) NOT NULL,  
  `regTime` datetime NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `telephone` varchar(50) NOT NULL,  
  PRIMARY KEY (`userId`),  
  KEY `userId` (`userId`)  
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;
```