毕业设计说明书

基于移动互联网的跳蚤市场

系统的设计与实现

1407094237

学生姓名： 学号：

大数据学院

吴霖铧

学 院：

物联网工程

专 业：

武瑞娟

指导教师：

2018年6月

基于移动互联网的跳蚤市场系统的设计与实现

摘要

基于移动互联网的跳蚤市场系统的设计与实现通过javaweb语言进行编写，使得系统可以在网络上使用运行，开发工具是MyEclipse，数据库采用MySQL，服务器使用Tomcat8.0，使用SSH框架，并采用MVC开发模式。

本论文主要分为四个部分，第一部分是关于系统的需求分析，具体详细的分析了使用本系统的用户想要实现的功能；第二部分是系统的概要设计，主要是使用类图与用例图展示系统各模块主要实现的功能是什么，用户使用该模块能得到什么结果；第三部分是系统的详细设计，主要是介绍系统各模块怎么使用，各模块的具体执行步骤；第四部分是系统的测试，主要是展示系统功能实际运行的效果。

通过以上步骤的完成，基于移动互联网的跳蚤市场系统的设计与实现就得到了完成。

关键词：SSH、管理、跳蚤市场

Design and Implementation of the Flea Market System

Based on Mobile Internet

Abstract

The design and implementation of a flea market system based on the mobile internet is written in the javaweb language, making the system available for use on the network. The development tool is MyEclipse, the database uses MySQL, the server uses Tomcat8.0, uses the SSH framework, and adopts the MVC development model. .

This paper is mainly divided into four parts. The first part is the analysis of the requirements of the system. The detailed analysis of the functions of the users who want to use this system is performed. The second part is the outline design of the system, mainly using class diagrams and use cases. The diagram shows what the main functions of each module of the system are, and what results the user can get using the module. The third part is the detailed design of the system. It mainly introduces how each module of the system is used and the specific steps of each module. The fourth part is The test of the system is mainly to demonstrate the effect of the actual operation of the system function.

Through the completion of the above steps, the design and implementation of a flea market system based on the mobile Internet has been completed.

Keywords: SSH, management, flea market

目 录

[1 需求分析](#_Toc22984_WPSOffice_Level1) [1](#_Toc22984_WPSOffice_Level1)

[1.1 项目概要](#_Toc19368_WPSOffice_Level2) [1](#_Toc19368_WPSOffice_Level2)

[1.2 客户需求](#_Toc12773_WPSOffice_Level2) [1](#_Toc12773_WPSOffice_Level2)

[1.3 系统功能结构](#_Toc31025_WPSOffice_Level2) [2](#_Toc31025_WPSOffice_Level2)

[1.4 用户角色和功能摘要](#_Toc9657_WPSOffice_Level2) [3](#_Toc9657_WPSOffice_Level2)

[2 概要设计](#_Toc19368_WPSOffice_Level1) [5](#_Toc19368_WPSOffice_Level1)

[2.1 开发环境](#_Toc1933_WPSOffice_Level2) [5](#_Toc1933_WPSOffice_Level2)

[2.2 系统功能描述](#_Toc8124_WPSOffice_Level2) [5](#_Toc8124_WPSOffice_Level2)

[2.3 功能模块设计](#_Toc604_WPSOffice_Level2) [7](#_Toc604_WPSOffice_Level2)

[2.3.1 用户登陆注册模块](#_Toc32540_WPSOffice_Level3) [7](#_Toc32540_WPSOffice_Level3)

[2.3.2 用户个人信息模块](#_Toc5100_WPSOffice_Level3) [9](#_Toc5100_WPSOffice_Level3)

[2.3.3 浏览商品模块](#_Toc4022_WPSOffice_Level3) [12](#_Toc4022_WPSOffice_Level3)

[2.3.4 购买商品模块](#_Toc5911_WPSOffice_Level3) [14](#_Toc5911_WPSOffice_Level3)

[2.3.5 用户管理商品模块](#_Toc21612_WPSOffice_Level3) [17](#_Toc21612_WPSOffice_Level3)

[2.3.6 订单查看模块](#_Toc5875_WPSOffice_Level3) [20](#_Toc5875_WPSOffice_Level3)

[2.3.7 管理员登陆模块](#_Toc16633_WPSOffice_Level3) [22](#_Toc16633_WPSOffice_Level3)

[2.3.8 用户管理模块](#_Toc2691_WPSOffice_Level3) [25](#_Toc2691_WPSOffice_Level3)

[2.3.9 商品管理模块](#_Toc16835_WPSOffice_Level3) [28](#_Toc16835_WPSOffice_Level3)

[2.3.10 订单管理模块](#_Toc24663_WPSOffice_Level3) [31](#_Toc24663_WPSOffice_Level3)

[2.4 数据库设计](#_Toc27746_WPSOffice_Level2) [34](#_Toc27746_WPSOffice_Level2)

[2.4.1 概念设计](#_Toc30676_WPSOffice_Level3) [34](#_Toc30676_WPSOffice_Level3)

[2.4.2物理结构设计](#_Toc22305_WPSOffice_Level3) [35](#_Toc22305_WPSOffice_Level3)

[3 详细设计](#_Toc12773_WPSOffice_Level1) [37](#_Toc12773_WPSOffice_Level1)

[3.1 系统定名](#_Toc32336_WPSOffice_Level2) [37](#_Toc32336_WPSOffice_Level2)

[3.2 模块详细设计](#_Toc1912_WPSOffice_Level2) [37](#_Toc1912_WPSOffice_Level2)

[3.2.1 用户登陆注册模块详细设计](#_Toc2462_WPSOffice_Level3) [37](#_Toc2462_WPSOffice_Level3)

[3.2.2 用户个人信息模块详细设计](#_Toc31729_WPSOffice_Level3) [38](#_Toc31729_WPSOffice_Level3)

第 Ⅰ 页

[3.2.3 浏览商品模块详细设计](#_Toc22217_WPSOffice_Level3) [39](#_Toc22217_WPSOffice_Level3)

[3.2.4 购买商品模块详细设计](#_Toc827_WPSOffice_Level3) [41](#_Toc827_WPSOffice_Level3)

[3.2.5 用户管理商品模块详细设计](#_Toc4006_WPSOffice_Level3) [41](#_Toc4006_WPSOffice_Level3)

[3.2.6 订单查看模块详细设计](#_Toc6929_WPSOffice_Level3) [42](#_Toc6929_WPSOffice_Level3)

[3.2.7 管理员登陆模块详细设计](#_Toc29945_WPSOffice_Level3) [43](#_Toc29945_WPSOffice_Level3)

[3.2.8 用户管理模块详细设计](#_Toc2109_WPSOffice_Level3) [44](#_Toc2109_WPSOffice_Level3)

[3.2.9 商品管理模块](#_Toc27435_WPSOffice_Level3) [45](#_Toc27435_WPSOffice_Level3)

[3.2.10 订单管理模块详细设计](#_Toc11552_WPSOffice_Level3) [47](#_Toc11552_WPSOffice_Level3)

[4 测试](#_Toc31025_WPSOffice_Level1) [48](#_Toc31025_WPSOffice_Level1)

[4.1 登陆模块测试](#_Toc25174_WPSOffice_Level2) [48](#_Toc25174_WPSOffice_Level2)

[4.2 注册模块测试](#_Toc5418_WPSOffice_Level2) [50](#_Toc5418_WPSOffice_Level2)

[4.3 浏览商品模块测试](#_Toc32540_WPSOffice_Level2) [52](#_Toc32540_WPSOffice_Level2)

[4.4 购买商品模块测试](#_Toc5100_WPSOffice_Level2) [55](#_Toc5100_WPSOffice_Level2)

[4.5 用户管理模块测试](#_Toc4022_WPSOffice_Level2) [56](#_Toc4022_WPSOffice_Level2)

[4.6 商品管理模块测试](#_Toc5911_WPSOffice_Level2) [57](#_Toc5911_WPSOffice_Level2)

[4.7 订单管理模块测试](#_Toc21612_WPSOffice_Level2) [58](#_Toc21612_WPSOffice_Level2)

[5 总结](#_Toc9657_WPSOffice_Level1) [61](#_Toc9657_WPSOffice_Level1)

[5.1 项目评价](#_Toc5875_WPSOffice_Level2) [61](#_Toc5875_WPSOffice_Level2)

[5.2 总结](#_Toc16633_WPSOffice_Level2) [61](#_Toc16633_WPSOffice_Level2)

[参 考 文 献](#_Toc1933_WPSOffice_Level1) [62](#_Toc1933_WPSOffice_Level1)

[致 谢](#_Toc8124_WPSOffice_Level1) [64](#_Toc8124_WPSOffice_Level1)

第 Ⅱ 页

第 Ⅱ 页

**1 需求分析**

**1.1 项目概要**

基于移动互联网的跳蚤市场系统的设计与实现能满足所有用户对商品进行浏览，支持游客进行注册，通过验证的用户进行登陆。只有登陆后的用户才能进行商品的购买与发布，买卖双方的交流沟通。后台管理员可以查看商品信息与编辑商品分类，也可以查看用户信息并对非法或违规的用户进行修改或者删除。

**1.2 客户需求**

客户需求概要表如表1.1所示：

表1.1 客户需求概要表

|  |  |
| --- | --- |
| 需求编号 | 需求内容 |
| X1 | 游客与用户都可以浏览商品内容 |
| X2 | 游客可以进行注册 |
| X3 | 注册成为用户后可以进行登陆，查看修改个人资料，查看交易情况 |
| X4 | 用户可以购买商品，发布商品，修改商品信息，删除商品 |
| X5 | 管理员可以查看所有商品、用户与交易订单，并能够对其编辑、删除 |
| X6 | 买卖双方可以进行联系沟通 |

客户需求详细表如表1.2所示：

表1.2 客户需求详细表

|  |  |
| --- | --- |
| 需求编号 | 需求内容 |
| X1 | 不论是游客还是已经通过注册的用户都可以对商品进行浏览，查看商品信息 |
| X2 | 游客通过注册之后可以进行用户登陆 |
| X3 | 3.1：登陆  用户通过填写能在数据库中找到相对应的账号密码进行登陆  3.2：查看修改个人资料  用户在登陆状态下可以对除了账号以外的所有信息进行修改  3.3：查看交易情况  用户可以查看订单了解订单的具体情况 |

续表1.2 客户需求详细表

|  |  |
| --- | --- |
| 需求编号 | 需求内容 |
| X4 | 4.1：购买商品  已登陆用户可以通过主页浏览，分类浏览确定想买的商品，通过下订单填写一些必要信息购买商品  4.2：发布商品  用户对商品信息进行编辑上传，发布的商品会出现在主页和相对应的分类中  4.3：修改商品  用户可以查看自己发布的商品，对商品的信息进行修改  4.4：删除商品  用户可以对不想出售的商品进行删除操作 |
| X5 | 5.1：查看商品、用户、订单  管理员在后台查看所有商品、用户、订单的详细内容或者信息  5.2：编辑和删除商品、用户、订单  管理员可以对违法或违规的商品、用户进行编辑或者删除，对存在问题的订单进行处理 |
| X6 | 用户在浏览商品时可以通过与发布该商品的用户取得联系，对商品进行讨论 |

**1.3 系统功能结构**

系统功能结构图如图1.1所示：

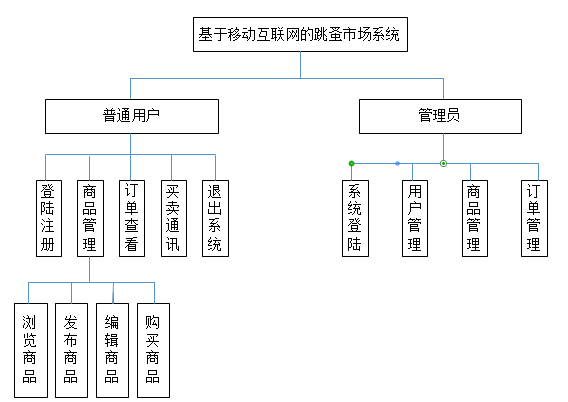


图1.1 系统功能结构图

基于移动互联网的跳蚤市场系统分为用户与管理员两个子系统。

用户通过登陆后可以进入用户子系统，在该子系统下用户可以对商品进行相关的操作，包括：浏览商品、发布商品、编辑商品、购买商品。用户可以查看自己的资料，也可以修改除了账号之外的所有资料，用户也能查看与自己有关的订单详情，但是不能对订单进行修改。用户可以通过商品页面中的相关按钮与该商品的发布人获得联系，对商品交易进行交流。

管理员通过后台登陆进入后台管理员子系统，在该子系统中，管理员可以对用户进行管理，具体包括对用户进行编辑或者删除，同样的，管理员也能对商品、订单进行相应的操作。

**1.4 用户角色和功能摘要**

用户角色说明表如表1.3所示：

表1.3 用户角色说明表

|  |  |
| --- | --- |
| 用户角色 | 用户描述 |
| 游客 | 没有注册，或者是没有登陆的用户，只能够查看商品，不能购买发布商品 |
| 用户 | 已经登陆的用户，能够进行商品相关的查看、购买、发布、编辑、删除，与卖家进行沟通交流，对个人信息进行查看修改 |
| 管理员 | 通过后台登陆，可以查看商品内容，对商品进行管理，对用户管理，对商品订单进行管理 |

功能摘要说明表如表1.4所示：

表1.4 功能摘要说明表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 主要功能点 | 功能描述 | 优先级 |
| 登陆、注册 | 登录、注册 | 游客申请账号，存到数据库，用户凭借账号密码登陆获取相关服务 | 高 |
| 首页 | 浏览商品、  跳转页面 | 不论是否登陆都可以在首页浏览商品，通过首页连接进行其他功能的跳转使用 | 高 |
| 个人信息 | 查看、修改个人信息 | 用户在登陆状态下对自己的信息进行查看修改 | 高 |
| 我的订单 | 查看订单 | 用户登陆状态下查看订单 | 高 |
| 商品详情 | 查看商品 | 用户点击商品图片跳转到商品具体信息页面，可以进行购买或者与卖家进行沟通协商 | 高 |
| 我的发布 | 发布编辑商品 | 用户在该模块可以发布新商品，也可以对先前发布的商品进行修改或者删除 | 高 |
| 订单管理 | 管理订单 | 管理员对所有订单进行管理，修改或者删除错误、无效的订单 | 高 |
| 商品管理 | 管理商品 | 管理员后台对商品进行管理，对违法或者违规商品进行删除 | 高 |
| 用户管理 | 管理用户 | 管理员对违规或违法用户进行删除 | 高 |

**2 概要设计**

**2.1 开发环境**

数据库：MySQL

Java Server：JDK 1.8.0\_131

开发模式：B/S

开发语言：Java、Android

开发工具：MyEclipse、Android Studio

开发系统环境：Windows 10

**2.2 系统功能描述**

2.2.1 系统综述

基于移动互联网的跳蚤市场是使用B/S模式，Java与Android语言进行开发的一个为高校学生提供闲置物品出售的平台。近年来移动互联网与电商的不断发展，基本上每位高校学生都有手机，本系统采用Java编写服务器来进行数据与数据库之间的交互，响应由客户端上发来的请求。客户端由Android进行开发，主要负责发送请求至服务器，将服务器中发送回来的数据进行展示，也能将数据发送至服务器中，服务器再将数据存入数据库中。

2.2.2 用户登陆注册模块

说明：未注册的用户通过填写信息并通过验证完成注册,已注册用户通过账号密码登陆

先决条件：未注册用户进行注册或已注册用户登陆

运行结果：用户完成登陆或者未注册用户完成注册

2.2.3 用户个人信息模块

说明：用户对自己的信息进行查看编辑

先决条件：用户已登陆

运行结果：用户查看到自己信息或者将修改后的信息传至数据库中

2.2.4 浏览商品模块

说明：所有用户都可以在首页或者分类中浏览商品

先决条件：所有用户都可以操作

运行结果：用户查看到商品信息

2.2.5 购买商品模块

说明：登陆后的用户购买商品

先决条件：登陆后的用户才能操作

运行结果：用户订单中出现新记录

2.2.6 用户管理商品模块

说明：用户发布商品、修改已发布商品信息、删除商品

先决条件：用户已登陆

运行结果：数据库中数据发生改变，首页上商品出现变化，商品内容改变

2.2.7 订单查看模块

说明：用户查看与自身相关的商品订单

先决条件：用户已登陆有过购买记录

运行结果：将数据库中关于该用户的订单详情展示

2.2.8 买卖双方通讯模块

说明：卖家与买家沟通交流

先决条件：买卖双方在线

运行结果：买卖双方进行即时通讯

2.2.9 管理员登陆模块

说明：管理员通过账号密码后台登陆

先决条件：数据库中存有管理员信息

运行结果：管理员进入后台管理系统

2.2.10 用户管理模块

说明：管理员对所有注册用户进行管理

先决条件：管理员已登陆

运行结果：数据库中用户信息改变

2.2.11 商品管理模块

说明：管理员对所有商品进行管理

先决条件：管理员已登陆

运行结果：数据库中商品信息改变

2.2.12 订单管理模块

说明：管理员对所有订单进行管理

先决条件：管理员已登陆

运行结果：数据库中订单信息改变

**2.3 功能模块设计**

2.3.1 用户登陆注册模块

（1）用户登陆注册模块类图如图2.1所示



图2.1 用户登陆注册类图

1. 用户登陆注册类图说明如表2.1所示：

表2.1用户登陆注册类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| UserAction | 处理关于用户操作Web请求，由Spring进行管理 |
| UserService | 用户操作具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |

续表2.1 用户登陆注册类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| UserDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| User | POJO类，对应相应的数据库中的表 |

（3）用户登陆注册模块方法说明如表2.2所示：

表2.2 用户登陆注册模块方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserAction：login() | 获取用户在页面上输入的数据，并调用Service层相应的方法 |
| UserService：login() | 隔离事务，并调用DAO层相应方法 |
| UserDao:login() | 对数据库进行查询操作，并向上级返回数据 |
| UserAction:regist() | 获取用户在页面上输入的数据，并调用Service层相应的方法 |
| UserService:regist() | 隔离事务，并调用DAO层相应方法 |
| UserDao:regist() | 对数据库进行增加记录操作 |
| User:getUname() | 获取用户账号的方法 |
| User:setUname() | 设置用户账号的方法 |
| User：getUpwd() | 获取用户密码的方法 |
| User:setUpwd() | 设置用户密码的方法 |

（4）用户登陆注册模块序列图如图2.2所示：

图2.2 用户登陆注册模块序列图

（5）用户登陆注册模块序列图说明表如表2.3所示：

表2.3 用户登陆注册模块序列图说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserAction:regist() | 获取用户在页面上输入的数据，并调用Service层相应的方法 |
| UserService:regist() | 隔离事务操作 |
| UserDao:regist() | 对数据库进行增加记录操作，并向上级返回数据 |
| UserAction:login() | 获取用户在页面上输入的数据，并调用Service层相应的方法 |
| UserService:login() | 隔离事务操作 |
| UserDao:login() | 对数据库进行查询操作，并向上级返回数据 |

2.3.2 用户个人信息模块

（1）用户个人信息模块类图如图2.3所示：



图2.3 用户个人信息模块类图

（2）用户个人信息模块类图说明表如表2.4所示：

表2.4 用户个人信息模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| UserAction | 处理关于用户操作Web请求，由Spring进行管理 |
| UserService | 用户操作具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| UserDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| User | POJO类，对应相应的数据库中的表 |

1. 用户个人信息模块方法说明如表2.5所示：

表2.5 用户个人信息模块方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserAction：info（） | 调用Service层相应方法将获得的数据展示在页面 |

续表2.5 用户个人信息模块方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserService：info() | 隔离事务，并调用DAO层相应方法 |
| UserDao:info() | 对数据库进行查询操作，并向上级返回数据 |
| UserAction: update() | 获取用户在页面修改的信息，调用Service层方法 |
| UserService:update() | 隔离事务，并调用DAO层相应的方法 |
| UserDao:update() | 对数据库进行修改操作 |
| User:getUname() | 获取用户账号的方法 |
| User:getUpwd() | 获取用户密码的方法 |
| User：getName() | 获取用户姓名的方法 |
| User:getPhone() | 获取用户电话的方法 |
| User:getAddr() | 获取用户地址的方法 |

1. 用户个人信息模块序列图如图2.4所示



图2.4 用户个人信息模块序列图

1. 用户个人信息模块序列图说明表如表2.6所示：

表2.6 用户个人信息模块序列图说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserAction:info() | 获取正登陆状态下的用户，并调用Service层相应的方法 |
| UserService:info() | 隔离事务操作 |
| UserDao:info() | 对数据库进行查询操作，并向上级返回数据 |
| UserAction:update() | 获取用户在页面上输入的数据，并调用Service层相应的方法 |
| UserService:update() | 隔离事务操作 |
| UserDao:update() | 对数据库进行增加记录操作 |

2.3.3 浏览商品模块

（1）浏览商品模块类图如图2.5所示：



图2.5 浏览商品模块类图

1. 浏览商品模块类图说明表如表2.7所示：

表2.7 浏览商品模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| GoodsAction | 处理关于商品操作Web请求，由Spring进行管理 |

续表2.7 浏览商品模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| GoodsService | 操作商品具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| GoodsDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Goods | POJO类，对应相应的数据库中的表 |

1. 浏览商品模块方法说明表如表2.8所示：

表2.8 浏览商品模块方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| GoodsAction: findByGid() | 获取到被浏览的商品信息，并调用Service层相应的方法 |
| GoodsService:findByGid() | 隔离事务，并调用DAO层相应的方法 |
| GoodsDao:findByGid() | 对数据库进行查询操作，并向上层返回信息 |
| Goods:getGname() | 获取商品名称的方法 |
| Goods:getGprice() | 获取商品价格的方法 |
| Goods:getGimage() | 获取商品图片的方法 |
| Goods:getGdesc() | 获取商品描述的方法 |

1. 浏览商品模块序列图如图2.6所示：

图2.6 浏览商品序列图

1. 浏览商品序列图说明表如表2.9所示：

|  |  |
| --- | --- |
| 方法名 | 说明 |
| GoodsAction:findByGid() | 获取被浏览的商品，并调用Service层相应的方法 |
| GoodsService:findByGid() | 隔离事务操作 |
| GoodsDao:findByGid() | 对数据库进行查询操作，并向上级返回数据 |

2.3.4 购买商品模块

（1）购买商品模块类图如图2.7所示：

图2.7 购买商品模块类图

1. 购买商品模块类图说明表如表2.10所示：

表2.10 购买商品模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| OrderAction | 处理关于用户订单操作Web请求，由Spring进行管理 |
| CartAction | 处理用户购物车操作Web请求，由Spring进行管理 |
| GoodsService | 操作商品具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| GoodsDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Goods | POJO类，对应相应的数据库中的表 |

1. 购买商品模块类图方法说明表如表2.11所示：

表2.11 购买商品模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| OrderAction: saveOrder() | 获取到想要购买的商品信息与购买人信息，调用CartAction中的方法 |
| CartAction:findByGid() | 获取到想要购买的商品信息，调用Service层中的方法 |
| GoodsSercive:findByGid() | 隔离事务，调用DAO层中的方法 |
| GoodsDao:findByGid() | 对数据库进行查询操作，并向上层返回信息 |
| Goods:getGname() | 获取商品名称的方法 |
| Goods:getGprice() | 获取商品价格的方法 |
| Goods:getGimage() | 获取商品图片的方法 |
| Goods:getGdesc() | 获取商品描述的方法 |

1. 购买商品模块序列图如图2.8所示：



图2.8 购买商品模块序列图

1. 购买商品模块序列图说明表如表2.12所示：

表2.12 购买商品模块序列图说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| OrderAction:saveOrder() | 新建一个订单信息，调用CartAction中的方法 |
| CartAction:findByGid() | 调用GoodsService层方法找到商品信息 |
| GoodsService:findByGid() | 隔离事务操作 |
| GoodsDao:findByGid() | 对数据库进行查询操作，并向上级返回数据 |

2.3.5 用户管理商品模块

（1）用户管理商品模块类图如图2.9所示：



图2.9 用户管理商品模块类图

1. 用户管理商品模块类图说明表如表2.13所示：

表2.13 用户管理商品模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| UserGoodsAction | 处理关于用户对商品操作的Web请求，由Spring进行管理 |

续表2.13 用户管理商品模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| GoodsService | 操作商品具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| GoodsDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Goods | POJO类，对应相应的数据库中的表 |

1. 用户管理商品模块类图方法说明表如表2.14所示：

表2.14 用户管理商品模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserGoodsAction: findByUid() | 获取到用户信息，调用Service层中的方法 |
| UserGoodsAction: save() | 获取到想要发布的商品信息，调用Service层中的方法 |
| UserGoodsAction: update() | 获取到需要修改的商品信息，调用Service层中的方法 |
| UserGoodsAction: delete() | 获取到需要删除的商品信息，调用Service层中的方法 |
| GoodsSercive:findByUid() | 隔离事务，调用DAO层中的方法 |
| GoodsSercive:save() | 隔离事务，调用DAO层中的方法 |
| GoodsSercive:update() | 隔离事务，调用DAO层中的方法 |
| GoodsSercive:delete() | 隔离事务，调用DAO层中的方法 |
| GoodsDao:findByUid() | 对数据库进行查询操作，并向上层返回信息 |
| GoodsDao:save() | 从上层获取到信息，将信息存入数据库 |
| GoodsDao:update() | 从上层获取到信息，将信息存入数据库 |
| GoodsDao:delete() | 从上层获取到信息，根据信息删除记录 |
| Goods:getGname() | 获取商品名称的方法 |
| Goods:getGprice() | 获取商品市场价格的方法 |

续表 2.14 用户管理商品模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| Goods:getGimage() | 获取商品图片的方法 |
| Goods:getGdesc() | 获取商品描述的方法 |
| Goods:getUid() | 获取发布者Id的方法 |

（4）用户管理商品序列图如图2.9所示：



图2.9 用户管理商品序列图

（5）用户管理商品序列图说明表如表2.15所示：

表2.15 用户管理商品序列图说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserGoodsAction:findByUid() | 获取用户信息，调用Service层相关的方法 |

续表2.15 用户管理商品序列图说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| GoodsService:findByUid() | 隔离事务操作 |
| GoodsDao:findByUid() | 对数据库进行查询操作，并向上级返回数据 |
| UserGoodsAction:save() | 获取商品与用户信息，调用Service层相关的方法 |
| GoodsService:save() | 隔离事务操作 |
| GoodsDao:save() | 从上层获取信息存入数据库中 |
| UserGoodsAction:update() | 获取商品与用户信息，调用Service层相关的方法 |
| GoodsService:update() | 隔离事务操作 |
| GoodsDao:update() | 从上层获取信息,对数据库中相应的信息进行修改 |
| UserGoodsAction:delete() | 获取商品与用户信息，调用Service层相关的方法 |
| GoodsService:delete() | 隔离事务操作 |
| GoodsDao:delete() | 从上层获取信息,对数据库中相应的信息进行删除 |

2.3.6 订单查看模块

（1）订单查看模块类图如图2.10所示：



图2.10 订单查看模块类图

1. 订单查看模块类图说明表如表2.16所示：

表2.16 订单查看模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| OrderAction | 处理关于用户订单操作Web请求，由Spring进行管理 |
| OrderService | 操作订单具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| OrderDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Order | POJO类，对应相应的数据库中的表 |

1. 订单查看模块类图方法说明表如表2.17所示：

表2.17 订单查看模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| OrderAction: findByUid() | 获取到用户信息，调用Service层中的方法 |
| OrderSercive:findByUid() | 隔离事务，调用DAO层中的方法 |
| OrderDao:findByUid() | 对数据库进行查询操作，并向上层返回信息 |
| Order:getOid() | 获取订单id的方法 |
| Order:getOdate() | 获取订单时间的方法 |
| Order:getOstate() | 获取订单状态的方法 |
| Order:getOuid() | 获取购买者Id的方法 |
| Order:getGid() | 获取商品Id的方法 |

1. 订单查看模块序列图如图2.11所示：



图2.11 订单查看模块序列图

1. 订单查看模块序列图说明表如表2.18所示：

表2.18 订单查看模块序列图说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| OrderAction:findByUid() | 获取用户信息，调用Service层相关的方法 |
| OrderService:findByUid() | 隔离事务操作 |
| OrderDao:findByUid() | 对数据库进行查询操作，并向上级返回数据 |

2.3.7 管理员登陆模块

（1）管理员登陆模块类图如图2.12所示：



图2.12 管理员登陆模块类图

1. 管理员登陆模块类图说明表如表2.19所示：

表2.19 管理员登陆模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| AdminAction | 处理关于管理员操作Web请求，由Spring进行管理 |
| AdminService | 管理员操作具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| AdminDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Admin | POJO类，对应相应的数据库中的表 |

1. 管理员登陆模块类图方法说明表如表2.20所示：

表2.20 管理员登陆模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminAction: login() | 获取到管理员信息，调用Service层中的方法 |
| AdminSercive:login() | 隔离事务，调用DAO层中的方法 |
| AdminDao:login() | 对数据库进行查询操作，并向上层返回信息 |
| Admin:getAname() | 获取管理员账号的方法 |
| Admin:getApwd() | 获取管理员密码的方法 |

1. 管理员登录模块序列图如图2.13所示：



图2.13 管理员登录模块序列图

1. 管理员登录模块序列图方法说明表如表2.21所示：

表2.21 管理员登录模块序列图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminAction:login() | 获取用户信息，调用Service层相关的方法 |
| AdminService:login() | 隔离事务操作 |
| AdminDao:login() | 对数据库进行查询操作，并向上级返回数据 |

2.3.8 用户管理模块

（1）用户管理模块类图如图2.14所示：



图2.14 用户管理模块类图

1. 用户管理模块类图说明表如表2.22所示：

表2.22 用户管理模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| UserAdminAction | 处理关于管理员对用户的操作Web请求，由Spring进行管理 |
| UserService | 管理员对用户操作具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |

续表 2.22 用户管理模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| UserDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| User | POJO类，对应相应的数据库中的表 |

1. 用户管理模块类图方法说明表如表2.23所示：

表2.23 用户管理模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserAdminAction: findAll() | 获取到用户信息，调用Service层中的方法 |
| UserSercive:findAll() | 隔离事务，调用DAO层中的方法 |
| UserDao:findAll() | 对数据库进行查询操作，并向上层返回信息 |
| UserAdminAction: delete() | 获取到用户信息，调用Service层中的方法 |
| UserSercive:delete() | 隔离事务，调用DAO层中的方法 |
| UserDao:delete() | 对数据库进行查询操作，并向上层返回信息 |
| UserAdminAction: edit() | 获取到用户信息，调用Service层中的方法 |
| UserSercive:edit() | 隔离事务，调用DAO层中的方法 |
| UserDao:edit() | 对数据库进行查询操作，并向上层返回信息 |
| UserAdminAction: update() | 获取到用户信息，调用Service层中的方法 |
| UserSercive:update() | 隔离事务，调用DAO层中的方法 |
| UserDao:update() | 对数据库进行查询操作，并向上层返回信息 |
| User:getName() | 获取用户姓名的方法 |
| User:getPhone() | 获取用户电话的方法 |
| User:getAddr() | 获取用户地址的方法 |

1. 用户管理模块序列图如图2.15所示：



图2.15 用户管理模块序列图

（5）用户管理模块序列图方法说明表如表2.24所示：

表2.24 用户管理模块序列图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserAdminAction:findAll() | 获取用户信息，调用Service层相关的方法 |
| UserService:findAll() | 隔离事务操作 |
| UserDao:findAll() | 对数据库进行查询操作，并向上级返回数据 |
| UserAdminAction:delete() | 获取用户信息，调用Service层相关的方法 |
| UserService:delete() | 隔离事务操作 |
| UserDao:delete() | 从上层获取信息并对数据库进行删除操作 |
| UserAdminAction:edit() | 获取用户信息，调用Service层相关的方法 |
| UserService:edit() | 隔离事务操作 |

续表2.24 用户管理模块序列图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| UserDao:edit() | 从上层获取信息并对数据库进行修改操作 |
| UserAdminAction:update() | 获取用户信息，调用Service层相关的方法 |
| UserService:update() | 隔离事务操作 |
| UserDao:update() | 从上层获取信息并对数据库进行更新操作 |

2.3.9 商品管理模块

（1）商品管理模块类图如图2.16所示：



图2.16 商品管理模块类图

1. 商品管理模块类图说明表如表2.25所示：

表2.25 商品管理模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| AdminGoodsAction | 处理关于管理员对商品操作Web请求，由Spring进行管理 |
| GoodsService | 商品操作具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| GoodsDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Goods | POJO类，对应相应的数据库中的表 |

1. 商品管理模块类图方法说明表如表2.26所示：

表2.26 商品管理模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminGoodsAction: findAll() | 获取到商品信息，调用Service层中的方法 |
| GoodsSercive:findAll() | 隔离事务，调用DAO层中的方法 |
| GoodsDao:findAll() | 对数据库进行查询操作，并向上层返回信息 |
| AdminGoodsAction:save() | 获取到商品信息，调用Service层中的方法 |
| GoodsSercive:save() | 隔离事务，调用DAO层中的方法 |
| GoodsDao:save() | 从上层获取信息并对数据库进行增加记录操作 |
| AdminGoodsAction: delete() | 获取到商品信息，调用Service层中的方法 |
| GoodsSercive:delete() | 隔离事务，调用DAO层中的方法 |
| GoodsDao:delete() | 从上层获取信息并对数据库进行删除操作 |
| AdminGoodsAction: edit() | 获取到商品信息，调用Service层中的方法 |
| GoodsSercive:edit() | 隔离事务，调用DAO层中的方法 |
| GoodsrDao:edit() | 从上层获取信息并对数据库进行修改操作 |

续表 2.26 商品管理模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminGoodsAction: update() | 获取到商品信息，调用Service层中的方法 |
| GoodsSercive:update() | 隔离事务，调用DAO层中的方法 |
| GoodsDao:update() | 从上层获取信息并对数据库进行更新操作 |
| Goods:getGname() | 获取商品名称的方法 |
| Goods:getGprice() | 获取商品市场价格的方法 |
| Goods:getGimage() | 获取商品图片的方法 |
| Goods:getGdesc() | 获取商品描述的方法 |

（4）商品管理模块序列图如图2.17所示：



图2.17 商品管理模块序列图

（5）商品管理模块序列图方法说明表如表2.27所示：

表2.27 商品管理模块序列图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminGoodsAction:findAll() | 获取商品信息，调用Service层相关的方法 |
| GoodsService:findAll() | 隔离事务操作 |
| GoodsDao:findAll() | 对数据库进行查询操作，并向上级返回数据 |
| AdminGoodsAction:delete() | 获取商品信息，调用Service层相关的方法 |
| GoodsService:delete() | 隔离事务操作 |
| GoodsDao:delete() | 从上层获取信息并对数据库进行删除操作 |
| AdminGoodsAction:edit() | 获取商品信息，调用Service层相关的方法 |
| GoodsService:edit() | 隔离事务操作 |
| GoodsDao:edit() | 从上层获取信息并对数据库进行修改操作 |
| AdminGoodsAction:update() | 获取商品信息，调用Service层相关的方法 |
| GoodsService:update() | 隔离事务操作 |
| GoodsDao:update() | 从上层获取信息并对数据库进行更新操作 |
| AdminGoodsAction:save() | 获取商品信息，调用Service层相关的方法 |
| GoodsService:save() | 隔离事务操作 |
| GoodsDao:save() | 从上层获取信息并对数据库进行增加记录操作 |

2.3.10 订单管理模块

（1）订单管理模块类图如图2.18所示：



图2.18 订单管理模块类图

（2）订单管理模块类图说明表如表2.28所示：

表2.28 订单管理模块类图说明表

|  |  |
| --- | --- |
| 类名 | 说明 |
| AdminOrderAction | 处理关于管理员对订单操作Web请求，由Spring进行管理 |
| OrderService | 订单操作具体业务组件开发，主要用于处理各类业务逻辑，并且能隔离事务和对其他资源调度。 |
| OrderDao | 专门独立出来对数据进行持久化，即处理书记的CRUD操作，也是利用Spring对Hibernate提供的DAO支持类HibernateDaoSupport来实现的。每个DAO组件对应一个数据库表 |
| Order | POJO类，对应相应的数据库中的表 |

1. 订单管理模块类图方法说明表如表2.29所示：

表2.29 订单管理模块类图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminOrderAction: findAll() | 获取到订单信息，调用Service层中的方法 |
| OrderSercive:findAll() | 隔离事务，调用DAO层中的方法 |
| OrderDao:findAll() | 对数据库进行查询操作，并向上层返回信息 |
| AdminOrderAction:updateState() | 获取到订单信息，调用Service层中的方法 |
| OrderSercive:updateState() | 隔离事务，调用DAO层中的方法 |
| OrderDao:updateState() | 从上层获取信息并对数据库进行修改操作 |
| AdminOrderAction: findOrderItem() | 获取到订单信息，调用Service层中的方法 |
| OrderSercive:findOrderItem() | 隔离事务，调用DAO层中的方法 |
| OrderDao:findOrderItem() | 从上层获取信息并对数据库进行查询操作 |

1. 订单管理模块序列图如图2.19所示：



图2.19 订单管理模块序列图

1. 订单管理模块序列图方法说明表如表2.30所示：

表2.30 订单管理模块序列图方法说明表

|  |  |
| --- | --- |
| 方法名 | 说明 |
| AdminOrderAction:findAll() | 获取订单信息，调用Service层相关的方法 |
| OrderService:findAll() | 隔离事务操作 |
| OrderDao:findAll() | 对数据库进行查询操作，并向上级返回数据 |
| AdminOrderAction:updateState() | 获取订单信息，调用Service层相关的方法 |
| OrderService:updateState() | 隔离事务操作 |
| OrderDao:updateState() | 从上层获取信息并对数据库进行修改操作 |
| AdminOrderAction:findOrderItem() | 获取订单信息，调用Service层相关的方法 |
| OrderService:findOrderItem() | 隔离事务操作 |
| OrderDao:findOrderItem() | 从上层获取信息并对数据库进行查询操作 |

**2.4 数据库设计**

2.4.1 概念设计

数据库设计E-R图如图2.20所示：

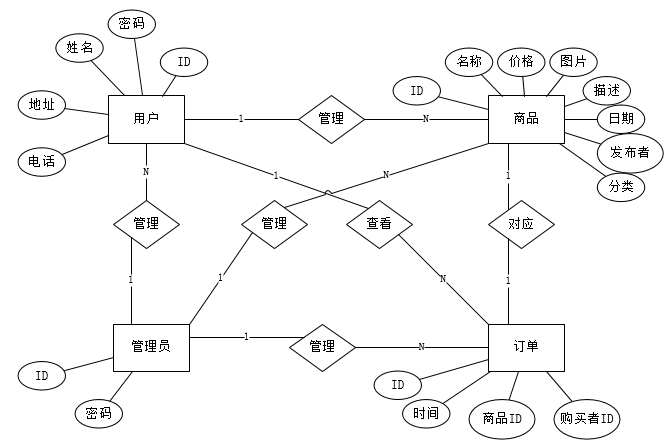


图2.20 E-R图

图2.20是数据库E-R图，其中存在四个对象，分别是用户，商品，管理员，订单；用户带有的属性是：ID，密码，姓名，地址，电话；商品带有的属性是：ID，名称，价格，图片，描述，日期，发布者，分类；管理员带有的属性是：ID，密码；订单带有的属性是：ID，时间，商品，购买者

2.4.2物理结构设计

1. 数据库用户信息表如表2.31所示：

表2.31 数据库用户信息表user

|  |  |  |  |
| --- | --- | --- | --- |
| 描述 | 字段 | 数据类型 | 注释 |
| ID | uid | int | 主键 |
| 账号 | uname | varchar(255) | 无 |
| 密码 | upwd | varchar(255) | 无 |
| 姓名 | name | varchar(255) | 无 |
| 地址 | addr | varchar(255) | 无 |
| 电话 | phone | varchar(255) | 无 |

1. 数据库商品信息表如表2.32所示：

表2.32 数据库商品信息表goods

|  |  |  |  |
| --- | --- | --- | --- |
| 描述 | 字段 | 数据类型 | 注释 |
| ID | gid | int | 主键 |
| 名称 | upwd | varchar(255) | 无 |
| 价格 | gprice | double | 无 |
| 图片 | gimage | varchar(255) | 无 |
| 描述 | gdesc | varchar(255) | 无 |
| 日期 | gdate | datatime | 无 |
| 发布者 | uid | int | 外键 |
| 分类 | csid | int | 外键 |

1. 数据库管理员信息表如表2.33所示：

表2.33 数据库管理员信息表admin

|  |  |  |  |
| --- | --- | --- | --- |
| 描述 | 字段 | 数据类型 | 注释 |
| ID | aid | int | 主键 |
| 账号 | aname | varchar(255) | 无 |
| 密码 | apwd | varchar(255) | 无 |

1. 数据库订单信息表如表2.34所示：

表2.34 数据库订单信息表order

|  |  |  |  |
| --- | --- | --- | --- |
| 描述 | 字段 | 数据类型 | 注释 |
| ID | oid | int | 主键 |
| 时间 | odate | datatime | 无 |
| 状态 | ostate | int | 无 |
| 商品 | gid | varchar(255) | 外键 |
| 购买者 | uoid | varchar(255) | 外键 |

**3 详细设计**

**3.1 系统定名**

中文名：基于移动互联网的跳蚤市场系统的设计与实现

英文名：Design and Implementation of the Flea Market System Based on Mobile Internet

功能定位：二手商品交易平台

**3.2 模块详细设计**

3.2.1 用户登陆注册模块详细设计

（1）用户登陆注册模块流程图如图3.1所示：



图3.1 用户登陆注册模块流程图

1. 用户登陆注册模块流程调用方法详述

用户进入主页面之后，可以点击登陆或者注册按钮进行相应的页面跳转与系统响应。

点击登陆按钮过后，页面跳转至登陆界面，用户在相应的位置输入与数据库中相同的用户名与密码，点击登陆按钮之后，页面中的表单会将用户所填写的信息发送到UserAction的login()方法中，UserAction的login()方法调用UserService中的login()方法，UserService中的login()调用UserDao中的login()方法。将用户的信息通过与数据库中的信息进行对比，如果存在相同的记录则登陆成功，否则登录失败。

点击注册按钮之后，页面跳转到注册页面，用户在填写信息之后点击注册按钮，页面中的表单会将信息发送到UserAction中的regist()方法，UserAction中的regist()方法调用UserService中的regist()方法，UserService中的regist()方法调用UserDao中的regist()方法。将用户填写的信息存入数据库中，注册成功则返回成功信息，失败则返回失败信息。

3.2.2 用户个人信息模块详细设计

（1）用户个人信息模块流程图如图3.2所示：



图3.2 用户个人信息模块流程图

（2）用户个人信息模块流程调用方法详述

用户在登陆状态下点击用户姓名按钮，页面跳转到个人信息展示界面，个人信息展示界面中包括个人账号，密码，姓名，地址，电话这些个人信息，用户可以点击编辑按钮进行除了个人账号以外信息的更改，编辑完成后点击确定按钮，页面中的表单会将用户填写的信息传到UserAction的update()方法中，UserAction的update()方法调用UserService的update()方法,UserService的update()方法调用UserDao的update()方法，UserDao的update()方法对数据库中相应的记录进行更新。

3.2.3 浏览商品模块详细设计

（1）浏览商品模块流程图如图3.3所示：



图3.3 浏览商品模块流程图

（2）浏览商品模块流程调用方法详述

用户可以在任意状态（即登陆状态或者未登陆状态）下进行商品的浏览，进入首页之后系统自动调用IndexAction中的findAll()方法，IndexAction中的findAll()方法调用GoodsService中的findAll()方法,GoodsService中的findAll()方法调用GoodsDao中的findAll()方法，将数据库中的商品信息查询出来之后保存到系统的值栈中，再根据商品发布时间进行倒叙排序，将最新的商品排到第一个。用户也可以点击分类进行该类型商品的查看，当用户点击某个分类时，系统进行页面跳转，调用GoodsAction中的findByCid()方法，GoodsAction中的findByCid()方法调用GoodsService中的findByCid()方法，GoodsService中的findByCid()方法调用GoodsDao中的findByCid()方法，从数据库中读取出相应的商品集合，将集合发送到页面中以供用户查看。用户可以点击想要查看的商品图片，系统依次调用GoodsAction、GoodsService、GoodsDao中的findByGid()方法，从数据库中读出相应的商品信息，发送到商品详情界面。

3.2.4 购买商品模块详细设计

（1）购买商品模块流程图如图3.4所示：



图3.4 购买商品模块流程图

（2）购买商品模块流程调用方法详述

用户只有在登陆状态下才能进行商品的购买操作，用户在想要购买的商品信息页面点击添加进入购物车按钮，系统依次调用CartAction的findByGid()方法，GoodsService的findByGid()方法，GoodsDao的findByGid()方法，将获取到的信息传到购物车中，在购物车页面点击提交订单，系统调用OrderAction的saveOrder()方法中，生成新的订单信息。

3.2.5 用户管理商品模块详细设计

（1）用户管理商品模块流程图如图3.5所示：



图3.5 用户管理商品模块流程图

（2）用户管理商品模块流程调用方法详述

用户在登陆状态下点击我的商品按钮，系统依次调用UserAction中的myGoods()方法，GoodsService中的findByUid()方法，GoodsDao中的findByUid()方法，将查询到的数据传输到跳转后的页面。用户可以在该页面点击添加按钮跳转到添加商品界面，点击确定按钮，系统依次调用UserGoodsAction、GoodsService、GoodsDao中的save()方法，将商品信息存入数据库中。关于用户删除、修改商品的流程与添加商品步骤类似，这里就不多叙述。

3.2.6 订单查看模块详细设计

（1）订单查看模块流程图如图3.6所示：



图3.6 订单查看模块流程图

1. 订单查看模块调用方法详述

用户在登陆的状态下点击我的订单按钮，系统依次调用OrderAction、OrderService、OrderDao的findByUid()方法，根据页面上session传过来的用户进行订单的查询，将查询到的结果传至跳转后的查看订单页面。

3.2.7 管理员登陆模块详细设计

（1）管理员登陆模块流程图如图3.7所示：



图3.7 管理员登陆模块流程图

（2）管理员登陆模块调用方法详述

管理员通过向页面中的表单中填写用户名和密码进行管理员后台登陆，系统依次调用AdminAction、AdminService、AdminDao中的login()方法，将页面表单中传过来的信息与数据库中的Admin表进行匹配，若没有找到记录，则返回错误并从定向到管理员后台登陆界面中，反之则登陆成功，跳转进入后台管理员管理界面。

3.2.8 用户管理模块详细设计

（1）用户管理模块流程图如图3.8所示：



图3.8 用户管理模块流程图

（2）用户管理模块调用方法详述

管理员在登陆进入到后台管理界面之后可以点击所有用户按钮，系统依次调用UserAdminAction、UserService、UserDao中的findAll()方法查询数据库中所有的用户，并将查询后的结果传输到跳转后的页面上，每条记录后面都存在着编辑和删除按钮，管理员可以点击编辑按钮，跳转到用户信息编辑页面，编辑完成后点击确定按钮，页面中的表单将信息发送到UserAdminAction中，系统在依次调用UserAdminAction、UserService、UserDao中的update()方法,将更新后的用户信息保存到数据库中。若是要进行删除用户操作，只需管理员点击用户记录后的删除按钮，页面将所要删除的用户Uid传输到UserAdminAction中，系统在依次调用UserAdminAction、UserService、UserDao中的delete()方法,将数据库中相对应的数据删除。

3.2.9 商品管理模块

（1）商品管理模块流程图如图3.9所示：



图3.9 商品管理流程图

(2)商品管理模块调用方法详述

管理员在后台管理页面中点击查看所有商品按钮，跳转到相应的查看商品页面，系统依次调用AdminGoodsAction、GoodsService、GoodsDao中的findAll()方法，从数据库中读取出所有商品的信息发送到查看商品页面中，每条商品记录的旁边都有编辑和删除按钮。管理员可以点击编辑按钮，页面跳转到编辑商品信息页面，并将该商品的Gid传输到AdminGoodsActi中，系统依次调用AdminGoodsAction、GoodsService、GoodsDao中的findByGid()方法，将数据库中查询到的相应的商品信息传输到编辑商品信息页面中，管理员编辑完成后点击确定按钮，提交表单中的内容到AdminGoodsActio中，系统依次调用AdminGoodsAction、GoodsService、GoodsDao中的update()方法,将数据库中相应的数据进行更新。管理员点击商品记录后的删除按钮之后，页面表单将商品的Gid传输到AdminGoodsAction中，系统依次调用AdminGoodsAction、GoodsService、GoodsDao中的delete()方法,将数据库中对应的记录删除。

3.2.10 订单管理模块详细设计

（1）订单管理模块流程图如图3.10所示：



图3.10 订单管理模块流程图

（2）订单管理模块流程调用方法详述

管理员点击查看所有订单按钮，页面跳转到查看所有订单，系统依次调用AdminOrderAction、OrderService、OrderDao中的findAll()方法，将数据库中的所有订单记录查询出来并发送到查看所有订单页面中，管理员可以进行查看订单信息，修改订单状态。管理员点击查看订单信息按钮后，页面跳转到查看订单详情页面，页面表单将该订单的oid传输到AdminOrderAction中，系统依次调用AdminOrderAction、OrderService、OrderDao中的findByOid()方法，将查询到的订单信息发送到订单详情页面。管理员点击更改订单状态按钮后，系统依次调用AdminOrderAction、OrderService、OrderDao中的updateState()方法，将数据库中相对应的订单记录中的ostate属性进行更新操作。

**4 测试**

**4.1 登陆模块测试**

已注册的用户信息为：用户名：wlh,密码：wlh;

已经保存在数据库中的管理员信息为：管理员名：admin,密码：admin;

（1）登陆模块测试用例表如表4.1所示：

表4.1 登陆注册模式测试用例表

|  |  |  |
| --- | --- | --- |
| 输入信息  （用户名，密码，确认登陆） | 预期结果 | 实际结果 |
| 用户名：aaa；密码：aaa | 账户名或密码错误，重新输入 | 账户名或密码错误重新输入! |
| 管理员名：admin；密码：ad | 账户名或密码错误，重新输入 | 账户名或密码错误，重新输入 |
| 用户名：wlh；密码：wlh | 登录成功 | 登录成功 |
| 管理员名：admin；密码：admin | 登录成功 | 登录成功 |

（2）登陆测试用例实际显示显示图如下图4.1，4.2，4.3，4.4所示：

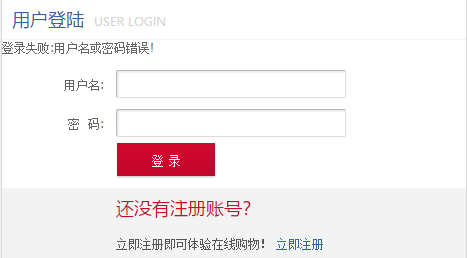


图4.1 登陆测试用例实际图1

用户名为aaa，密码为aaa登陆失败后情况

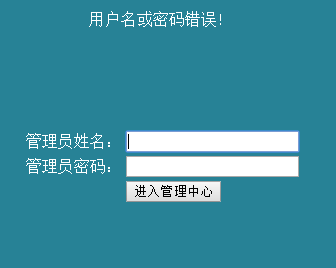


图4.2 登陆测试用例实际图1

管理员名为admin，密码为ad登陆失败的情况



图4.3 登陆测试用例实际图3

用户名为wlh，密码为wlh登陆成功后的情况

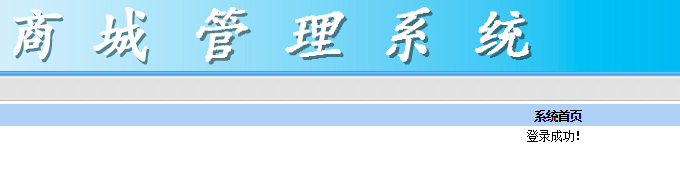


图4.4 登陆测试用例实际图4

管理员名为admin，密码为admin登陆成功的情况

**4.2 注册模块测试**

已存在的用户：用户名：wlh，密码：wlh；

用户名：wu，密码：wu

1. 注册模块测试用例表如表3.2所示：

表3.2 注册测试用例表

|  |  |  |
| --- | --- | --- |
| 输入信息  （用户名，密码，确认登陆） | 预期结果 | 实际结果 |
| 用户名：wlh；密码：wlh | 用户名已存在 | 用户名已存在 |
| 管理员名：w；密码： | 密码不能为空 | 密码不能为空 |
| 用户名：wu；密码：wu | 用户名已存在 | 用户名已存在 |
| 管理员名：w；密码：w | 注册成功 | 注册成功 |

1. 注册测试用例实际显示图如下列图4.5，4.6，4.7，4.8所示：



图4.5 注册测试用例实际显示图1

用户名为wlh，密码为wlh注册失败的情况

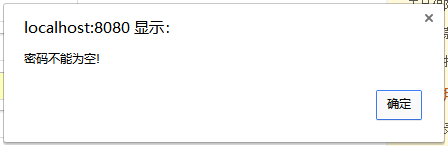


图4.6 注册测试用例实际显示图2

用户名为w，密码为空注册失败的情况



图4.7 注册测试用例实际显示图3

用户名为wu，密码为wu注册失败的情况

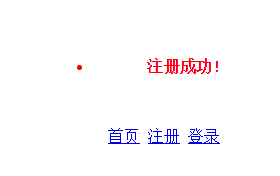


图4.8 注册测试用例实际显示图4

用户名为w,密码为w注册成功的情况

**4.3 浏览商品模块测试**

（1）浏览商品模块测试用例表如表4.3所示：

表4.3 浏览商品模块测试用例表

|  |  |  |
| --- | --- | --- |
| 点击按钮 | 预期结果 | 实际结果 |
| 首页 | 进入首页看到所有商品 | 进入首页看到所有商品 |

续表4.3 浏览商品模块测试用例表

|  |  |  |
| --- | --- | --- |
| 点击按钮 | 预期结果 | 实际结果 |
| 一级分类 | 进入一级分类界面看到商品 | 进入一级分类界面看到商品 |
| 二级分类 | 进入二级分类界面看到商品 | 进入二级分类界面看到商品 |
| 商品图片 | 进入商品详细内容界面 | 进入商品详细内容界面 |

1. 浏览商品模块测试用例实际图如下列图4.9，4.10，4.11，4.12所示：

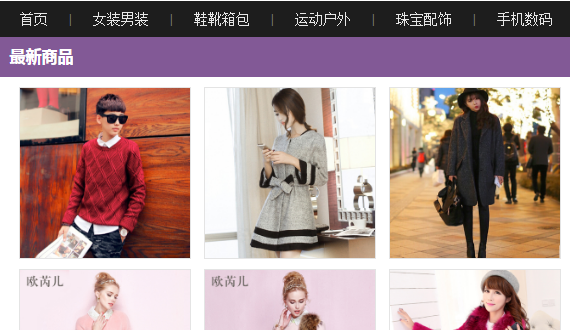


图4.9 浏览商品模块测试用例实际图1

点击首页按钮所出现的商品



图4.10 浏览商品模块测试用例实际图2

点击一级分类女装男装的结果



图4.11 浏览商品模块测试用例实际图3

点击二级分类冬季外套的结果



图4.12 浏览商品模块测试用例实际图4

点击商品图片查看商品详情的结果

**4.4 购买商品模块测试**

（1）购买商品模块测试用例表如表4.4所示：

表4.4 购买商品测试用例表

|  |  |  |
| --- | --- | --- |
| 点击按钮 | 预期结果 | 实际结果 |
| 提交订单 | 出现新订单 | 出现新订单 |

（2）购买商品模块测试用例实际图如图4.13所示：



图4.13 购买商品模块测试用例实际图

**4.5 用户管理模块测试**

（1）用户管理模块测试用例表如表4.5所示：

表4.5 用户管理模块测试用例表

|  |  |  |
| --- | --- | --- |
| 点击按钮 | 预期结果 | 实际结果 |
| 编辑用户 | 跳转到编辑界面 | 跳转到编辑界面 |
| 删除用户 | 用户记录消失 | 用户记录消失 |
| 提交编辑后的用户信息 | 用户信息改变 | 用户信息改变 |

（2）用户管理模块测试用例实际图如下列图4.14，4.15，4.16，4.17所示：



图4.14 用户管理模块测试用例实际图1

初始状态下的所有用户记录

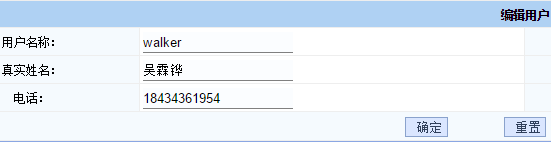


图4.15 用户管理模块测试用例实际图2

点击编辑用户按钮后的界面

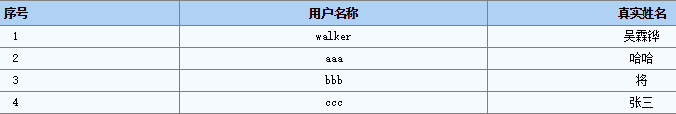


图4.16 用户管理模块测试用例实际图3

删除了第五条用户记录后的结果



图4.17 用户管理模块测试用例实际图4

修改第一条记录真实姓名后的结果

**4.6 商品管理模块测试**

（1）商品管理模块测试用例表如表4.6所示：

表4.6 商品管理模块测试用例表

|  |  |  |
| --- | --- | --- |
| 点击按钮 | 预期结果 | 实际结果 |
| 编辑商品 | 跳转到编辑界面 | 跳转到编辑界面 |
| 删除商品 | 商品记录消失 | 商品记录消失 |

（2）商品管理模块测试用例实际图如下列图4.18，4.19，4.20所示：



图4.18 商品管理模块测试用例实际图1

初始查看所有商品记录

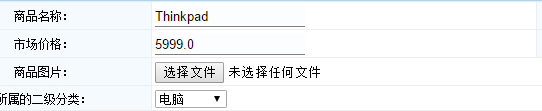


图4.19 商品管理模块测试用例实际图2

编辑商品界面



4.20 商品管理模块测试用例实际图3

删除一条商品记录的结果

**4.7 订单管理模块测试**

（1）订单管理模块测试用例表如表4.7所示：

表4.7 订单管理模块测试用例表

|  |  |  |
| --- | --- | --- |
| 点击按钮 | 预期结果 | 实际结果 |
| 查看订单 | 查看订单详情 | 查看订单详情 |
| 修改订单状态 | 订单状态改变 | 订单状态改变 |

（2）订单管理测试用例实际图如下列图4.21，4.22，4.23所示：



图4.21 订单管理测试用例实际图1

初始状态的订单列表



图4.22 订单管理测试用例实际图2

点击订单详情的结果



图4.23 订单管理测试用例实际图3

修改订单状态的结果

**5 总结**

**5.1 项目评价**

此次的毕业设计，我使用了简单易懂的SSH框架进行开发，开发过程比较迅速，学习到的新知识也比较快的能够融汇贯通。由于是通过一些16年前的资料来学习的SSH框架，导致有时候会因为环境的不匹配而出现错误，但也是这些错误让我更加深入了解了SSH框架的使用。

**5.2 总结**

时光匆匆，大学四年的生活即将结束，毕业设计也到了尾声。在这短短的几个月中，我通过为了完成毕业设计而学习新的知识，查阅一些资料慢慢的感觉到了毕业设计这段时间里自己的改变。大学不像高中那般什么知识都会有老师完完全全的教给你，大部分知识需要自己去学习和理解。毕业设计是体现一名大学生在大学四年中学习能力，动手能力的一个衡量手段，它督促着我去学习必需的知识，学着动手将学到的知识运用到实际生活中，而不是单单存在于脑海里作为理论。这段时间，辛苦夹杂着喜悦，汗水夹杂着成果，每一个阶段的实现都能让我对知识的使用更加熟练，每一个模块的完成都能让我感到汗水的值得。毕业设计教会了我，努力付出总能得到你相应的回报，这是大学生活最后也是最好的礼物了。

# 参 考 文 献

[1]王雅, 杨柳, 叶凤. 移动互联网时代高校跳蚤市场的营销模式选择[J]. 生产力研究, 2015(6):119~123.

[2]信海辉, 张姗姗, 黄晋超等. 基于互联网+的大学生跳蚤平台企划构想[J]. 课程教育研究:新教师教学, 2016(28):283~284.

[3]梁剑云, 孟秋廷, 李存等. 高校跳蚤市场的现状研究及平台构建[J]. 科技经济导刊, 2017(17):241~241.

[4]张亮亮, 孟庆国, 李瑞英等. 基于校园网上二手交易平台的研究[J]. 电脑知识与技术, 2015(16):252~253.

[5]万宏恩, 周洁, 方柏岚等. 基于微信的高校跳蚤市场平台搭建[J]. 现代经济信息, 2016(5):319~319.

[6]李典, 王一策. 高校微信平台跳蚤市场的发展现状研究[J]. 科技传播, 2016(10):94~106

[7]彭伸源, 王春璐, 张悦等. 基于微信平台信息管理的校园闲置物品交易平台应用分析[J]. 商业经济, 2017(2):184~186.

[8]方柏岚, 周洁, 万宏恩等. AISAS模式下微信平台的校园跳蚤市场的营销策略[J]. 中国市场, 2016(22):34~35.

[9]任培花, 宣昱如. 基于Android的“校园跳蚤市场”移动端App的设计[J]. 计算机与数字工程, 2016, 44(11):2276~2279.

[10]邹蕾. 基于Android的校园二手交易系统设计与实现[D]. 硕士学位论文.湖北武汉:华中科技大学, 2014.

[11]康珣. 面向高校的Android二手交易系统[J]. 信息与电脑:理论版, 2016(2):104~105.

[12]陈森. 绿色经济:“互联网+”时代的转让与回收[J]. 上海信息化, 2017(4):15~19.

[13]刘常亮. 基于Android手机的条码识别系统研究[D].硕士学位论文.陕西西安: 西安电子科技大学, 2012.

[14]Liu S T, Wang J, Yang X M. Entrepreneurial Orientation Construction Practice of Campus Flea Market Trading Platform based on O2O[J]. Computer Knowledge & Technology, 2016:266~267.

[15]Liu S T, Zhu B Z, Wang J et al. Design and Implementation of Campus Flea Market Trading System Based on O2O[J]. Value Engineering, 2016:273~277.

致 谢

本设计是在我的指导教师武瑞娟老师的悉心关怀与指导下完成。在整个设计完成的期间，不论是资料的查找与收集，还是系统逻辑的设计与编写，武老师都怀有热情、耐心的指导我。即时武老师处于课务繁忙期间，她也会在百忙之中抽出时间耐心的帮我解答疑惑，不时的给我提出宝贵的建议，让我少走了许多弯路，在武老师的指导下，我在对事务的处理，学习的方法和态度，知识的获取与理解上都有了很大的进步与提高。在此对武老师表示衷心的感谢。

感谢在我设计完成过程中给予意见和帮助的各位老师和同学。

最后，感谢在百忙之中抽出宝贵时间来参与评审的专家。