**摘要** 现如今人们的生活水平的快速提高，越来越多的人为了陶冶情操，排解内心的孤单和寂寞，都会选择去养一只宠物，而养宠物则需要一些宠物日常用品。随着电商时代到来，网络购物已经步入了寻常百姓家，为了迎合时代的发展，满足人们足不出户就能购买商品，给用户的日常生活带来了更多的便利和更好的体验。它采用ThinkPHP框架及MVC开发模式作为主要实现方式，完成了用户登录，商品管理，权限管理等模块。系统界面友好，操作简单、易于维护扩展。

**关键词** 宠物用品；网上商城；ThinkPHP

**ABSTRACT** Nowadays, people's living standards are rapidly improving. More and more people choose to raise a pet in order to cultivate their sentiments and resolve their inner loneliness and loneliness. Pets need some pet daily necessities. With the advent of the e-commerce era, online shopping has entered the homes of ordinary people. In order to cater to the development of the times, people can buy goods without leaving home, bringing more convenience and better to the daily life of users. Experience. It adopts ThinkPHP framework and MVC development mode as the main implementation method, and completes user login, commodity management, and rights management modules. The system interface is friendly, easy to operate, easy to maintain and expand.

**KEY WORDS** Pet supplies; online store; ThinkPHP

**目 录**

[第1章 前 言 1](#_Toc417634359)

[1.1 课题的背景 1](#_Toc417634360)

[1.2 课题研究目的与意义 1](#_Toc417634361)

[1.3 本文的组织结构 2](#_Toc417634362)

[第2章 相关技术和开发运行环境 3](#_Toc417634363)

[2.1 PHP语言简介 3](#_Toc417634364)

[2.2 ThinkPHP框架简介 3](#_Toc417634365)

[2.3 MVC架构简介 4](#_Toc417634366)

[2.4 CSS样式表 4](#_Toc417634367)

[2.5 JavaScript脚本语言 5](#_Toc417634368)

[2.6系统运行环境 5](#_Toc417634369)

[第3章 系统需求分析和设计 6](#_Toc417634370)

[3.1 系统需求分析 6](#_Toc417634371)

[3.2 系统功能结构图 7](#_Toc417634372)

[3.3 购物流程图 9](#_Toc417634373)

[3.4数据库设计 10](#_Toc417634374)

[3.4.1 数据库分析 10](#_Toc417634375)

[3.4.2数据库E-R图 10](#_Toc417634376)

[3.4.3 数据库结构表 12](#_Toc417634377)

[3.5系统模块设计 14](#_Toc417634378)

[3.5.1 系统架构 14](#_Toc417634379)

[3.5.2 目录结构 15](#_Toc417634380)

[3.5.3 模块设计 15](#_Toc417634381)

[3.5.4 模型设计 15](#_Toc417634382)

[3.5.5 模板设计 15](#_Toc417634383)

[第4章 系统的实现 17](#_Toc417634384)

[4.1 系统前台功能模块的实现 17](#_Toc417634385)

[4.1.1 前台首页的布局 17](#_Toc417634386)

[4.1.2 登录管理 18](#_Toc417634387)

[4.1.3 注册管理 20](#_Toc417634388)

[4.1.4 分类展示 22](#_Toc417634389)

[4.1.5 搜索管理 23](#_Toc417634390)

[4.1.6 购物车管理 25](#_Toc417634391)

[4.2 系统后台功能模块的实现 26](#_Toc417634392)

[4.2.1管理员登录模块 27](#_Toc417634393)

[4.2.2后台首页的布局 28](#_Toc417634394)

[4.2.3权限管理模块 28](#_Toc417634395)

[4.2.4商品管理模块 30](#_Toc417634396)

[第5章 总结和展望 46](#_Toc417634397)

[5.1总结 46](#_Toc417634398)

[5.2 展望 46](#_Toc417634399)

[致 谢 47](#_Toc417634400)

[参考文献 48](#_Toc417634401)

# 前 言

## 课题的背景

随着社会不断的发展，人们的消费水平地提高，人们的物质需求已达到满足，为满足人们的精神需求，饲养宠物已逐渐成为人们的一大喜好。随着拥有宠物的消费者地增加，对于宠物日常用品的需求也随之增加。

而如今世界已处于互联网时代，互联网时代的来临催生了网络媒体、远程医疗、网络娱乐、互联网农业、电子商务、社交网络、网络教育、物联网以及互联网金融一系列新兴产业。在这一新兴事物的影响下，人们已经开始习惯足不出户在网上购买商品，传统店铺销售已不如以前。为了顺应潮流，PetMall宠物网上商城正是为此而开发。

## 1.2 课题研究目的与意义

我国的宠物行业属于朝阳产业，相关数据表明，当一个国家的人均 GDP 在 3000 至 8000 美元之间，宠物行业就会迅速发展。以目前的情况来看，我国的很多城市市区人均 GDP 已经超过了 3000美元，宠物行业已经发展到一定的规模，从事宠物行业的人超过 6万，宠物行业拥有的固定资产超过 20 亿元，一年创造的产值超过60 亿元。在北京登记注册的宠物狗数量超过 50 万只，上海则超过 70 万只。通过研究数据，可以发现国内的宠物市场以纯种狗和猫为主，经济涨幅超过两成。在宠物花费上，上海每年养狗的费用超过 6 亿元，每个家庭每个月的宠物花费大概在 300 元; 在青岛，养宠物的花费在几百块至几千块，而且这部分的花费仍然在持续增加。按照这样的速度发展下去，我国将拥有超过 1． 5 亿只的宠物，宠物行业的产值将超过 400 亿元。[1]

在宠物销售量持续增加的同时，与宠物相关的产品也得到了热销，例如宠物服装、宠物医疗等，销售宠物周边也成为做生意的门道。宠物食品包括饲料、鱼虾罐头、饼干、悠闲食品等。国产饲料的价钱比较便宜，一般的饲料每公斤 12 元，进口宠物食品的价钱是国产饲料的数倍。比起宠物养殖，销售宠物食品以及其他用品的形式更为灵活，在宠物经济中占据较大的比例。在国际市场，已经有企业将注意力放在生产和销售宠物服饰、香水、玩具等，甚至奢侈品行业的巨头也为宠物设计一系列的产品。[1]

因而，本课题的主要目标就是为方便消费者更好的购买宠物日常商品，促进传统实体店进行转型。

## 1.3 本文的组织结构

表1-1 本文的组织结构

|  |  |
| --- | --- |
| **章节** | **主 要 内 容** |
| 第一章 前言 | 本章简单介绍了时代已处于互联网时代，阐述了新时代背景下的宠物经济发展，并由此提出了项目的背景、目标。 |
| 第二章 相关技术和开发运行环境 | 本章主要阐述系统开发所用到的关键技术和开发运行环境。 |
| 第三章 PetMall网上商城系统的设计 | 本章分析了网站系统的需求，详细规划了网站的结构与功能，并进行了数据库的设计。 |
| 第四章 PetMall网上商城系统的实现 | 本章根据第三章的设计内容，给出网站系统的几个重要模块的实现过程。 |
| 第五章 总结与展望 | 本章是对所做工作的总结与展望，为后续的再开发奠定基础。 |

# 第2章 相关技术和开发运行环境

本章主要介绍了在项目开发过程中需要使用到的相关技术和开发工具，并且简单介绍了本网站系统的开发运行环境。

## 2.1 PHP语言简介

PHP是一种广泛使用的服务器端编程语言，用于开发动态网页。它是一种开源的、跨平台的、独立于架构的、解释的、面向对象的、快速的、简单易学的、健壮的 、安全的、性能优越的用于Web服务器端动态网页开发的语言。目前已有超过上千万的网站和数万家公司，以及数百万的程序开发人员在使用它。它是目前动态网页开发中使用最为广泛的语言之一。PHP语言以其优异的性能、强大的数据库操作功能迅速成为主流的Web服务器端开发语言[2]。

## 2.2 ThinkPHP框架简介

ThinkPHP是一款由国人开发基于MVC机构的轻巧型开发开源框架，提供了基础web开发功能，其内置了很多安全处理机制，开发人员可以使用ThinkPHP开发出强壮而又高效的PHP应用，大大简化了企业级应用开发。ThinkPHP的MVC模式吸收了所有主流框架的特点提供了层次分明、设计合理的MVC分层设计。ThinkPHP很好内置了许多安全处理机制，帮助发者在应用层方面尽量减少安全漏洞[3]。

## 2.3 MVC架构简介

MVC是一种先进的开发模式，能够解决团队开发之间协同配合的问题，使得网站各部件以更高的效率运行。MVC模式将网站分为3大部分，分别为模型、视图、控制器[3]。MVC的结构如图2-1所示。

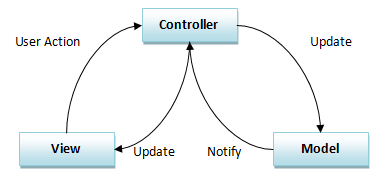


图2-1 MVC模型

模型（Model）： 用于封装与web应用与业务逻辑有关的数据以及对其的处理方法。模型有对数据库直接访问的权利，及模型不依赖视图和控制器，模型不关心自身会被如何显示。

视图（View）：对控制器转过来的数据进行有目的的显示。在这一层中一般没有业务逻辑。

控制器（Controller）：控制器起到组合模型和视图的组织作用，用于控制web应用的流程，对用户的事件或数据库反馈的数据进行响应。

## 2.4 CSS样式表

CSS(Cascading Style Sheet，简称CSS)中文译为层叠式表, 又称串样式列表、级联样式表、串接样式表、阶层式样式表，一种用来为结构化文档（HTML页面或XML应用）添加样式，并允许将样式信息与文档分离的一种计算机语言，由W3C定义和维护。CSS的引入就是为了使得HTML语言能够更好地适应页面的美工设计。它以HTML用语言为基础，提供了丰富的格式化功能，如字体、颜色、背景和整体排版等，并且页面设计者可以针对各种可实话浏览器设置不同的样式风格。

CSS样式一般是作用在DIV上的，可以为每个DIV区域定义独特的CSS，这样我们就可以将网页中指定的DIV部分变成我们所需要的风格、样式，而网页又是由多个DIV构成，从而达到梅花页面的作用[4]。

## 2.5 JavaScript脚本语言

JavaScrip是面向Web的编程语言。绝大多数现代网站都使用了JavaScript,并且所以的现代网站都使用了JavaScript，并且所以的现代Web浏览器一一基于桌面系统、游戏机、平板电脑和智能手机的浏览器一一均包含了JavaScript解释器。这使得JavaScript能够称得上史上使用最广泛的编程语言。JavaScript也是前端开发工程师必须掌握的三种技能之一：描述网页内容的HTML、描述网页样式的CSS以及描述网页行为的JavaScript.[5]。

## 2.6系统运行环境

WampServer是Windows Apache Mysql PHP集成安装环境，即Windows下的Apache+Mysql/MariaDB+Perl/PHP/Python，一组常用来搭建动态网站或者服务器的[开源软件](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90%E8%BD%AF%E4%BB%B6/8105369)，本身都是各自独立的程序，但是因为常被放在一起使用，拥有了越来越高的兼容度，可以让开发者一次安装并设置好开发环境。本课题系统的运行将采直接采用wampserver。其版本为WampServer 3.1.4，包含版本：Apache 2.4.4 ，Mysql 5.6.12， PHP 5.4.16。

# 第3章 系统需求分析和设计

## 3.1 系统需求分析

关于本系统几个方面的需求，包括系统的用户群，界面，功能，安全与保密性等。

系统用户群：消费者、管理员；

界面：友好，简单，易用；

本系统是由前台展示区和后台网站管理组成。

前台展示区模块的主要功能有以下一些：

l、任何用户都能直接在网络上浏览网站上的商品；

2、可以在线注册网站的会员用户；

3、会员用户可以在线浏览商品，订购商品，生成订单，在线支付；

4、会员用户可以查询已购买的商品，以及配送状态；

5、会员用户可以对已购买的商品进行查看和评价；

6、会员用户可以对自己的账户信息进行管理

7、销售排行

8、购物车

后台网站管理主要功能有以下一些：

1. 注册用户与权限管理
2. 用户管理
3. 员工管理
4. 添加管理组
5. 栏目分类管理
6. 分类战术
7. 一级分类添加
8. 二级分类添加
9. 商品内容管理
10. 订单展示
11. 商品信息展示
12. 商品信息添加
13. 商品评价展示
14. 用户地址展示
15. 服务管理
16. 网站服务展示
17. 新增一级分类
18. 新增二级分类
19. 其他参数管理
20. 配置展示
21. 添加配置
22. 修改配置
23. 友情链接

针对该公司的需求，本文3.2节对系统进行了分模块的设计说明。

## 3.2 系统功能结构图

为了能更好的说明网站的结构，下面给出本电子商务网站的前台功能模块结构图和后台功能模块结构图。

手工品网上商城系统前台包含一个电子商务网站所应具有的一些基本功能，其设计能满足用户正常浏览商品，加入购物车，购物商品，同时用户可以在用户中心中查看订单和修改个人信息等进行一些基本操作[10]。手工品网上商城系统前台管理模块的功能设计，前台功能模块结构图，如图3-1所示。

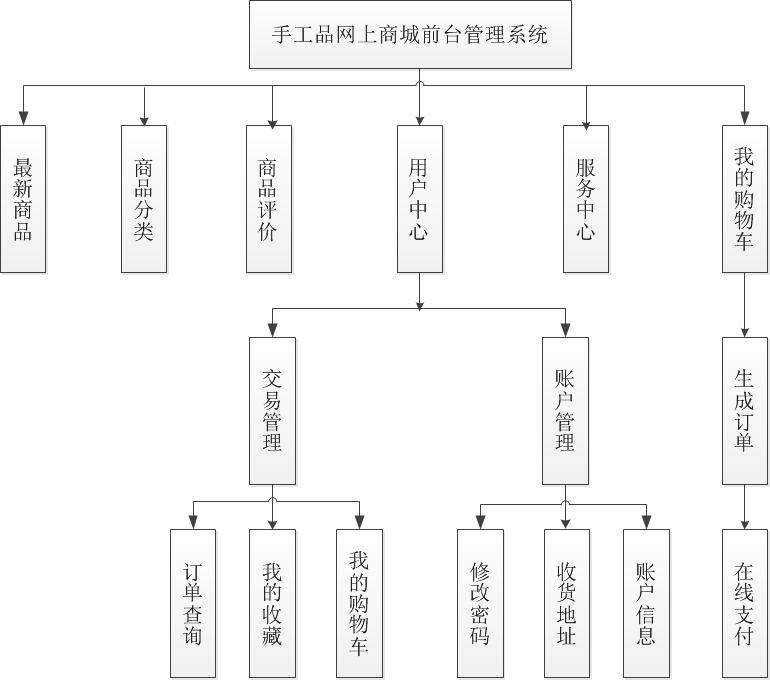


图3-1 前台功能模块结构图

与系统前台功能相对应，手工品网上商城系统的后台包含用户管理，商品管理，订单管理和信息管理等一些基本功能。管理员可以删除用户，添加新商品，回复用户留言等进行些基本的操作。手工品网上商城系统后台管理模块的功能设计，如图3-2所示。

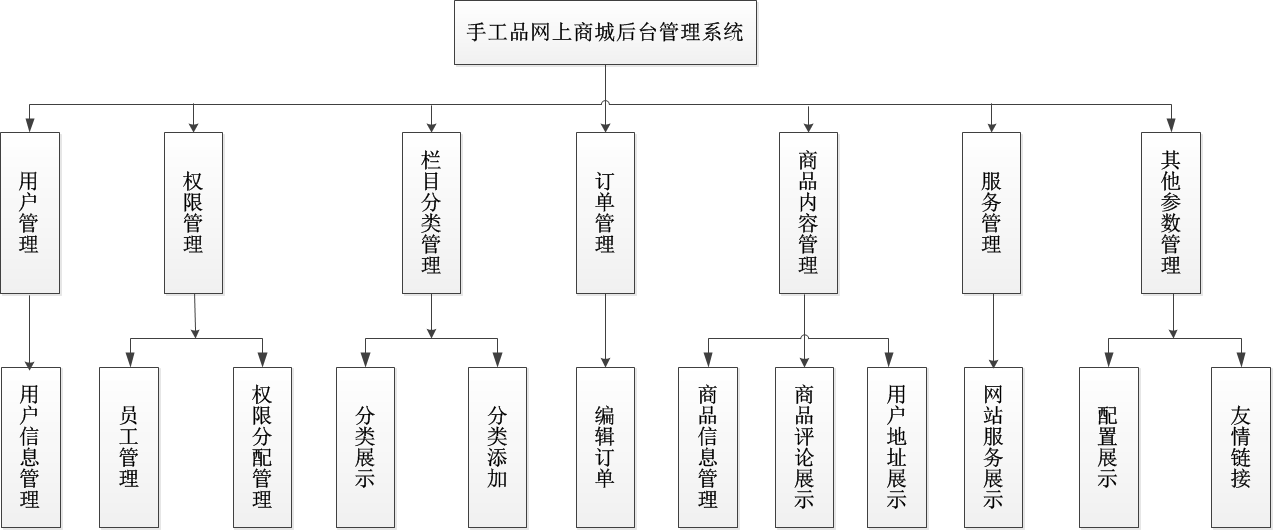


图3-2 后台功能模块结构图

## 3.3 购物流程图

所谓电子商务，实质就是建立一个虚拟的购物超市，首先客户应该在网页中选购自己需要的商品并将商品放入购物车中，选购好商品后就可以到收银台填写订单信息，提交订单和付款等操作[11]。电子商务平台的购物流程图，图3-3所示。

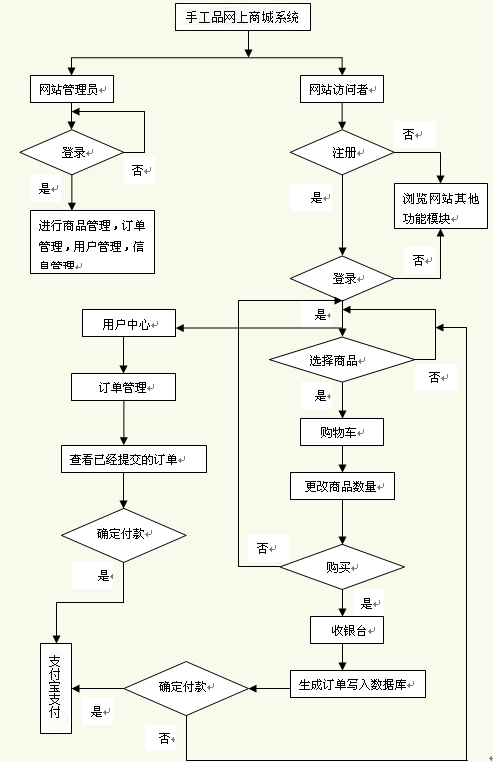


图3-3 购物流程图

## 3.4数据库设计

## 3.4.1 数据库分析

因为本项目需要充分考虑到成本及用户需求等问题，而MySQL是世界上最为流行的开放源码的数据库，是完全网络化的跨平台的关系型数据库，这正好满足了中小型企业的需求，所以本系统采用MySQL数据库。

## 3.4.2数据库E-R图

根据以上对系统的需求分析，规划出本购物系统的实体关系E-R图。表示方法为：

实体：用矩形表示，矩形框内标注实体名

属性：用椭圆表示，用横线将其与实体连接起来。

联系：用棱形表示（1:n ,1:1，m:n）。

1、用户信息实体

用户信息实体包括用户ID(id)，登录名(name)，密码(pwd)，真名(realname),性别(sex),邮箱(email),等级(level),积分(score),注册时间(regtime),登录时间(logtime)，手机号码(phone)，所在地（address）。

用户信息实体的E-R图,如图3-4所示。

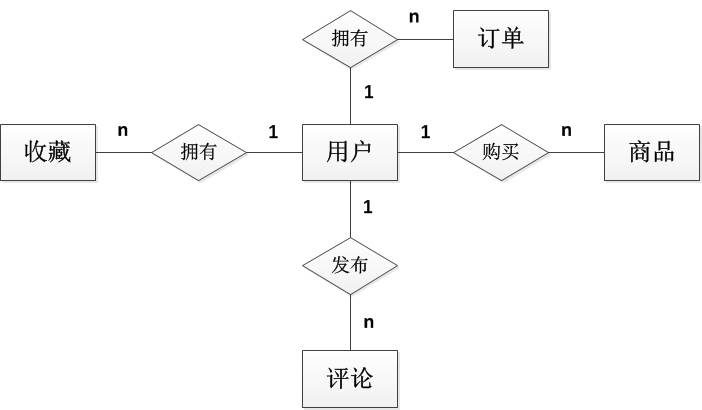


图3-4 用户信息实体E-R图

2、管理员信息实体

管理员信息实体包括管理员ID(id),角色ID(role\_id),管理员名（name）,密码(password)。管理员信息实体的E-R图,如图3-5所示。

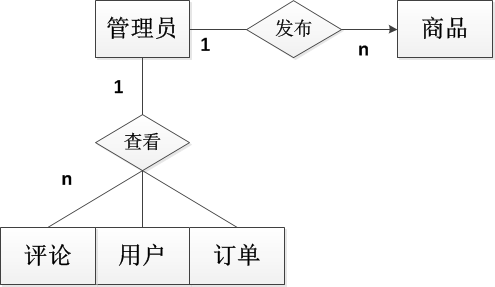


图3-5 管理员信息实体E-R图

3、商品信息实体

商品信息实体包括编号(id),商品名称(name),商品所属分类(cateid),商品存货(store),商品品牌(brandid),商品现价(price),商品原价（oldprice）,上架状态（status）,图片链接地址(imgs),商品描述（description）。商品信息实体E-R图如图3-6所示。

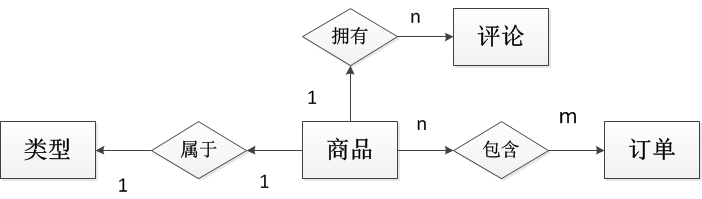


图3-6 商品信息实体E-R图

4、商品订单实体

商品订单实体包括订单ID(id),订单号(number),下单时间(time),下单人(ordername),订单状态(state),订单价格(sumprice),订单用户ID(uid),商品名称(gname),订单地址（addr）。商品订单实体E-R图,如图3-7所示。



图3-7 商品订单实体E-R图

## 3.4.3 数据库结构表

根据在数据库设计中给出的数据库实体E-R图，设计数据表结构。数据库名称为myproject，包含17张数据表，表前缀为aimer\_ 。具体表格清单如表3-1所示。

表3-1 数据库中的数据表

|  |  |  |
| --- | --- | --- |
| **表名** | **备注** | **用途** |
| aimer\_address | 地址信息表 | 存储用户地址的详细信息 |
| aimer\_collect | 收藏商品表 | 存储用户收藏的商品信息 |
| aimer\_comment | 用户评论表 | 存储用户的评论信息 |
| aimer\_order | 订单信息表 | 存储用户订单信息 |
| aimer\_user | 用户信息表 | 存储用户发送站内信信息 |
| aimer\_access | 用户权限表 | 存储用户的访问权限 |
| aimer\_brand | 商品品牌表 | 存储商品品牌信息 |
| aimer\_link | 友情链接表 | 存储友情链接的信息 |
| aimer\_role | 用户角色分组表 | 存储用户角色分组信息 |
| aimer\_role\_user | 用户角色映射表 | 存储用户角色映射 |
| aimer\_goods | 商品信息表 | 存储商品的详细信息 |
| aimer\_cate | 商品分类信息表 | 存储商品分类的信息 |
| aimer\_goodlist | 商品类型表 | 存储商品类型的信息 |
| aimer\_detaill | 商品分类映射表 | 存储商品分类映射的信息 |
| aimer\_node | 节点表 | 存储节点信息 |
| aimer\_service | 商城服务管理表 | 存储服务管理信息 |
| aimer\_website | 网站配置信息表 | 存储网站配置信息 |

以下是列出几个重要的表格的详细设置：

订单信息表（aimer\_order）字段详细设置，如表3-2所示。

表3-2 aimer\_order数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **属性** | **约束条件** | **说明** |
| id | int | 无符号、非空、自增 | 主键 | 订单id |
| number | char(30) | 非空 |  | 订单编号 |
| uid | int | 非空 |  | 订单用户id |
| gid | int | 非空 |  | 订单商品id |
| time | int | 非空 |  | 下单时间 |
| total | int | 非空 |  | 订单商品总数 |
| sumprice | int | 非空 |  | 订单商品总价 |
| state | int | 非空 |  | 订单目前状态 |
| color | varchar | 非空 |  | 商品颜色 |
| size | varchar | 非空 |  | 商品尺寸 |
| addr | varchar | 非空 |  | 商品地址 |

用户权限表（aimer\_access）字段详细设置，如表3-3所示。

表3-3 aimer\_access数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **属性** | **约束条件** | **说明** |
| role\_id | int | 无符号、非空、自增 |  | 角色id |
| node\_id | smalint | 非空 |  | 节点id |
| level | tinyint | 非空 |  | 等级 |
| module | varchar | 非空 |  | 备注 |

商品分类信息表（aimer\_cate）字段详细设置，如表3-4所示。

表3-4 aimer\_cate数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **属性** | **约束条件** | **说明** |
| id | int | 无符号、非空、自增 | 主键 | 分类id |
| name | char | 非空 |  | 商品类别 |
| pid | int | 非空 |  | 人群id |
| image | varchar | 非空 |  | 商品图片 |
| lid | int | 非空 |  | 商品类别ID |

用户角色分组表（aimer\_rol）字段详细设置，如表3-5所示。

表3-5 aimer\_rol数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段** | **类型** | **属性** | **约束条件** | **说明** |
| id | int | 无符号、非空、自增 | 主键 | 用户角色分组id |
| name | varchar | 非空 |  | 分组名称 |
| status | tinyint | 非空 |  | 分组状态 |

## 3.5系统模块设计

## 3.5.1 系统架构

手工品网上商城采用ThinkPHP3.2框架作为系统架构。ThinkPHP框架的设计采用MVC设计模式，Controller对应MVC控制器，Model对应MVC模型，View对应MVC视图。相对应地，规划设计控制器、模型和模板成为代码开发阶段的主要工作[5]。

## 3.5.2 目录结构

系统采用单一入口模式进行项目部署和访问，一个统一的入口完成需求的所以功能。于入口文件同级的是一些特定文件的目录，如下表所示：

表3-6 目录结构说明

|  |  |
| --- | --- |
| **文件/目录** | **说明** |
| Application | 应用目录 |
| ThinkPHP | 框架目录 |
| public | 资源文件目录，存放图片、CSS、JS和其它静态文件 |
| README.md | README文件 |
| index.php | 入口文件 |

## 3.5.3 模块设计

针对ThinkPHP的模块而言，就是控制器的设计。系统采用模块和操作的方式来执行，首先，用户的请求会通过入口文件生成一个应用实例，应用控制器（也称为核心控制器）会管理整个用户执行的过程，并负责模块的调度和操作的执行，并且在最后销毁该应用实例。用户所需要的那些功能和操作，具体换成的一个URL访问，都是某个模块的某个操作。所以，实现系统功能，就是设计模块和编写方法。

## 3.5.4 模型设计

每个模型类对应一张数据表，主要用来实现以下两点功能。

* 自动验证：自动验证是ThinkPHP模型层提供的一种数据验证方法，可以在使用create创建数据对象的时候自动进行数据验证。
* 自动完成：自动完成是ThinkPHP提供用来完成数据自动处理和过滤的方法，使用create方法创建数据对象的时候会自动完成数据处理。

## 3.5.5 模板设计

模板作为渲染前的初级样式，由静态页面转换而成。模板设计的内容包括以下几项。

* 为对应的模块方法创建相应的模板文件。
* 将html代码替换成可行的模板标签或php代码。
* 对公共部分进行模板布局，即抽去所以页面的头部和尾部形成独立文件。

# 第4章 系统的实现

## 4.1 系统前台功能模块的实现

## 4.1.1 前台首页的布局

商城首页采用了css+div的布局。商城首页，如图4-1所示。

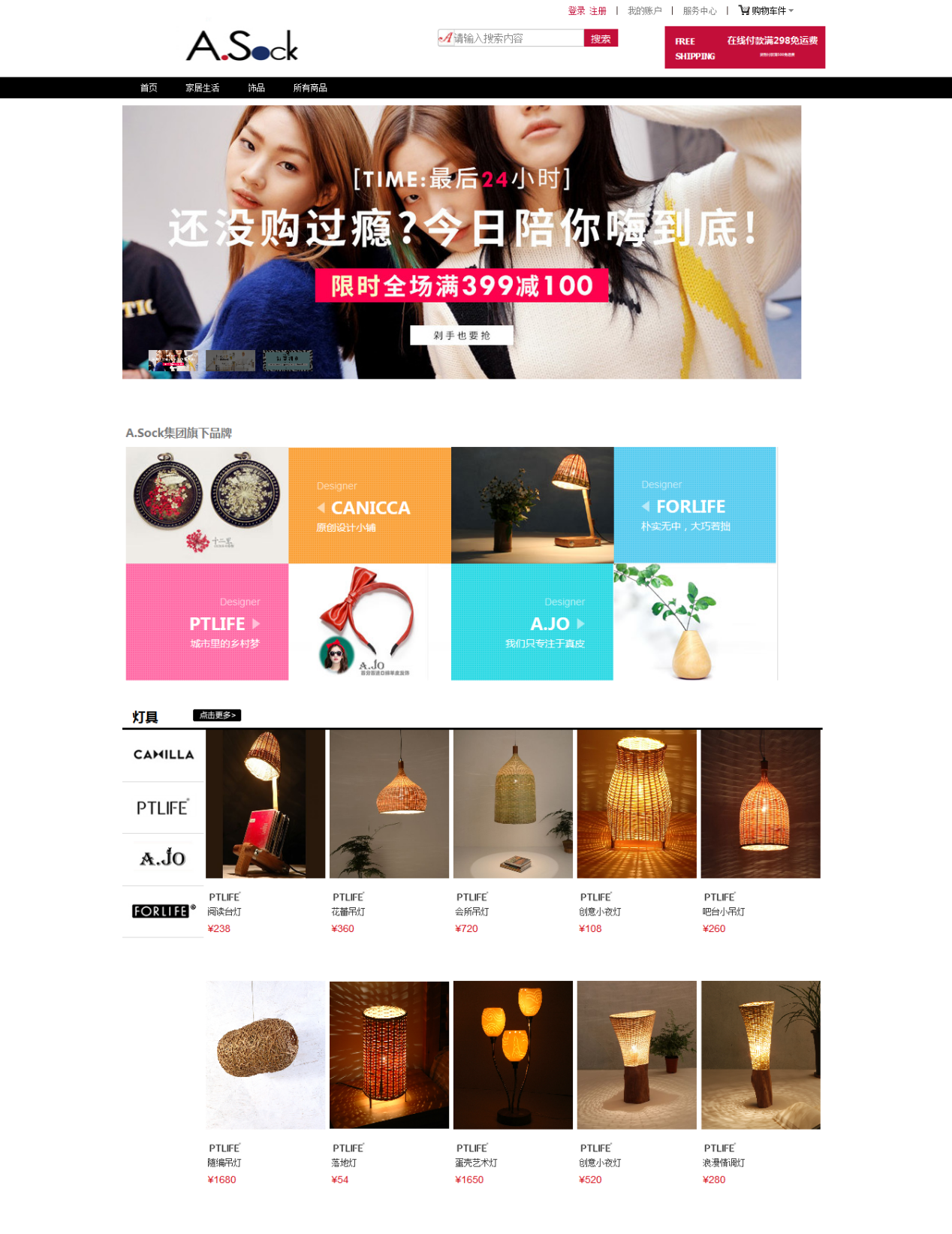


图4-1 前台首页示意图

## 4.1.2 登录管理

用户登录管理模块的实现主要使用两个文件，一个是填写登录信息的视图文件：Application/Home/View/login/index.html,另一个是接收登录信息写入数据库的实现文件：Application/Home/Controller/LoginController.class.php。用户登录页如图4-2所示。



图4-2 登录页面

登录功能的文字描述：

一、验证输入的用户名：

当登录名的输入框失去焦点时发生blue事件

当登录名输入框获得焦点时发生focus事件

二、验证输入的密码：

当密码的输入框失去焦点时发生blue事件

当密码输入框获得焦点时发生focus事件

三、回车触发提交

1、按回车键，触发登录按钮的click事件

2、用ajax方法提交用户名和密码到对应控制器的checklogin方法

四、ajax验证是否存在该用户

用ajax方法提交用户名到对应控制器的ajaxfind方法

五、checklogin方法：

查询用户：通过I来获取前台提交过来的数据，实例化User模型，查询用户名，如果用户名和密码正确，就创建session，保持最后登陆的IP和时间

六、ajaxfind方法：

通过$\_POST['name']获取用户名信息，实例化空模型，创建SQL语句查询用名，输出结果

其对应的LoginController控制器实现脚本的核心代码如下：

//查询用户,并创建session

public function checklogin() {

sleep(1);

$username = I('post.name');

$password = I('post.pwd');

$check = I('post.check'); //通过I来获取前台提交过来的数据

$user = M('user'); //实例化User模型

$data = array(

'name' => $username,);

$r = $user->where($data)->find();

if($r){

if(md5($password) != $r['pwd']){

echo 'pwd';

return;

}elseif($r['status'] ==1 ){

echo 'status';

return;

}else{

$\_SESSION= $r;

$ip=ip2long($\_SERVER['REMOTE\_ADDR']);

$time=time();

$data = array(

'lastip'=>$ip,

'logtime' => $time

);

$db = M('User');

$db -> where("id = {$r['id']}")->save($data);

echo 'ok';

}

}else{

echo 'nouser';

return;

}

## 4.1.3 注册管理

用户登录管理模块的实现主要使用两个文件，一个是填写注册信息的视图文件：Application/Home/View/Register/index.html,另一个是接收注册信息的实现文件：Application/Home/Controller/RegisterController.class.php。注册页面，如图4-3所示。



图4-3 注册页面

注册功能的文字描述：

一、验证输入登录名的格式是否等于邮箱

二、验证输入的登录名：

当登录名的输入框失去焦点时发生blue事件

当登录名输入框获得焦点时发生focus事件

三、显示密码强度， 获取焦点事件

四、验证输入的密码

用thinkphp框架的test方法

五、验证重复的密码

六、判断用户名是否被占用

通过对应的控制器的ajaxfind方法查询数据

七、回车触发提交

1、按回车键，触发登录按钮的click事件

2、用ajax方法提交用户名和密码到对应控制器的checkreg方法

八、checkreg方法：

通过I来获取前台提交过来的数据，create方法创建数据对象。add方法写入数据库

九、ajaxfind方法：

通过$\_POST['name']获取用户名信息，实例化空模型，创建SQL语句查询用名，输出结果

其对应的RegeditController控制器实现脚本的核心代码如下：

public function checkreg() {

sleep(1);

$name = I('post.name');

$pwd = I('post.pwd');

$user = D('user');

if($user->create()){ // 创建数据对象

$user->regtime = time();

$user->logtime = time();

$user->lastip = ip2long($\_SERVER['REMOTE\_ADDR']);

$user->add()；

$data = array(

'name' => $name,

'pwd' => md5($pwd)

);

$r = $user->where($data)->find();

$\_SESSION = $r;

echo 'yes';

}else{

echo 'no';

return;

}

}

## 4.1.4 分类展示

分类展示模块的实现主要使用两个文件，一个是显示分类展示的视图文件：Application/Home/View/List/index.html,另一个是实现分类展示的文件：Application/Home/Controller/ListController.class.php。分类展示页面，如图4-4所示。

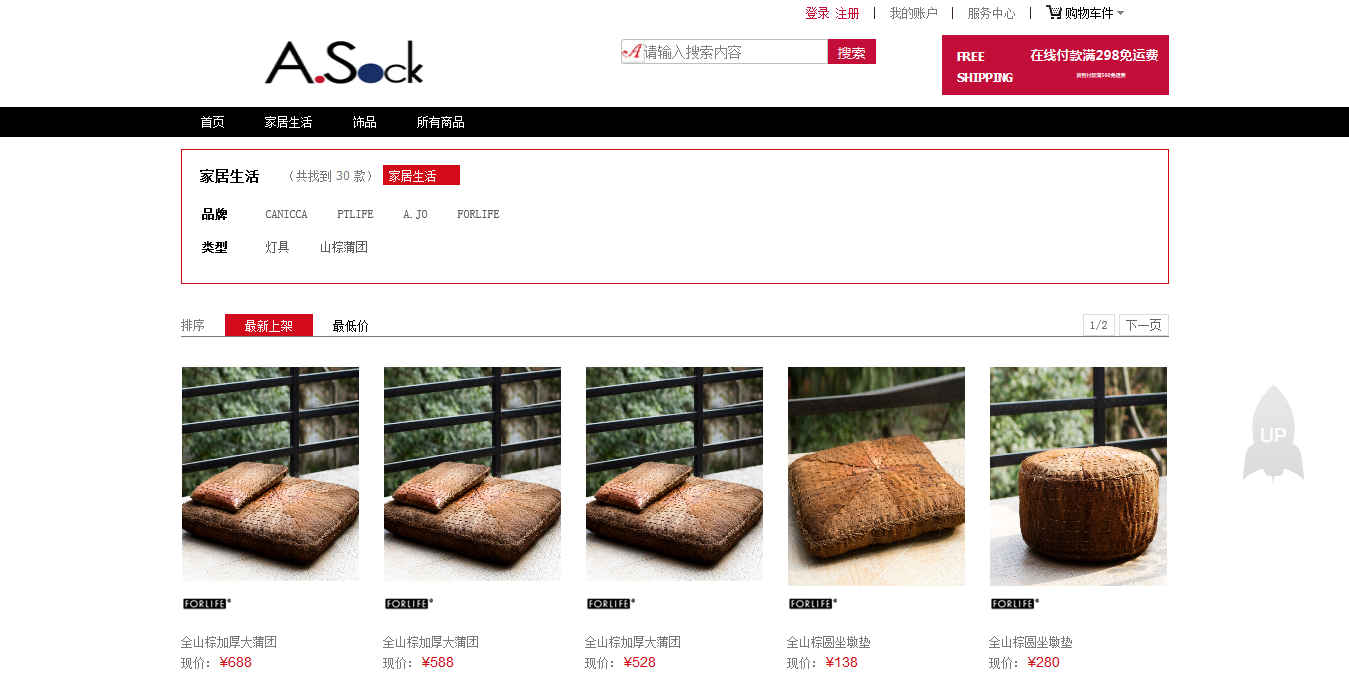


图4-4 分类展示页面

分类展示功能的文字描述：

实例化Cate模型，通过I来获取前台提交的数据，select方法来查询数据，assign方法来赋值变量，display方法来输出变量

其对应的ListController控制器实现脚本的核心代码如下：

public function go(){

$db = M('cate');

$p=I('get.p');

$ls=I('get.ls');

$resu=I('get.b');

$brand=I('get.gb');

$b=$db->select();

$list = $db->select();

$accon['pid'] = array('gt',0);

$lost = $db->where($accon)->select();

$d=$db->where("pid=0")->select();

$arr = $db->where('pid=0')->select();

$acc = $db->table('aimer\_goodlist')->select();

$detail=$db->table('aimer\_listdetail')->select();

$usedetail=$db->table('aimer\_detailall')->select();

$getbrand=$db->table('aimer\_brand')->select();

$f=$db->table('aimer\_goods')->where("listid={$ls}")->select();

$a=$db->where("id={$resu}")->find(); //查出一维数据,页面跳转时使用的

$c=$db->where("id={$p}")->find();

$brandget=$db->table("aimer\_brand")->where("id={$brand}")->find();

$e=$db->table('aimer\_goodlist')->where("id={$ls}")->find();

......

}

## 4.1.5 搜索管理

搜索模块的实现主要使用两个文件，一个是显示搜索管理的视图文件：Application/Home/View/Search/index.html,另一个是实现分类展示的文件：Application/Home/Controller/SearchController.class.php。搜索页面，如图4-5所示。



图4-5 搜索页面

搜索功能的文字描述：

1、查询cate分类表，分配导航栏数据

2、接收前台首页搜索框传过来的关键词

3、查询商品数据表

4、将传过来的搜索关键词赋值给数组$map

5、以传过来的关键词为条件查出相关数据总条数

6、实例化分页类，为搜索到的数据创建分页

7、以限制条件查询每页应显示的查询结果

8、把post传过来的关键词分配到前台搜索页面模板

9、向搜索页面的分页按钮分配数据

10、向搜索页面分配搜索结果的显示数据

其对应的SearchController控制器实现脚本的核心代码如下：

public function index(){

//查询cate分类表，分配导航栏数据

$db = M('cate');

$arr = $db->where('pid=0')->select();

$list1 = $db->select();

$gid = $\_SESSION['id'];

$ac = $db->table('aimer\_user')->where("id={$gid}")->find();

$this -> assign('namego',$ac);

$this -> assign('toplist',$list1);

$this -> assign('listone',$arr);

//接收前台首页搜索框传过来的关键词

$search= I('get.keywords');

//从cate分类表中匹配搜索关键词

$map['name'] = array('like',"%$search%");

$mlist = $db -> where($map) -> field('id > 3') -> select();

foreach($mlist as $value){

$arr\_cate[] = $value['id'];

}

//dump(array\_values($arr\_cate));

//exit;

//查询商品数据表

$list = M('goods');

//通过匹配到的分类表id查询商品表的分类cateid，并赋值给数组$map

//$map['cateid'] = array('in',array\_values($arr\_cate));

// 将传过来的搜索关键词赋值给数组$map

$map['name'] = array('like',"%$search%");

//$map['color'] = array('like',"%$search%");

//$map['size'] = array('like',"%$search%");

$map['cateid'] = array('in',array\_values($arr\_cate));

//$map['status'] = 1;

$map['\_logic'] = 'OR';

//dump($map);

//exit;

……

}

## 4.1.6 购物车管理

购物车管理模块的实现主要使用两个文件，一个是购物车管理的视图文件：Application/Home/View/Shopcar/index.html,另一个是实现购物车管理的文件：Application/Home/Controller/ShopcarController.class.php。购物车页面，如图4-6所示。



图4-6 购物车页面

购物车功能的文字描述：

addCard方法（添加购物车）：

1.用I方法获取购物车所有商品的id和数量。

2.使用session保存购物信息，把id和数量保存到二维数组中

clear方法（清空购物车）：

用unset方法，销毁session

其对应的ShopcarController控制器实现脚本的核心代码如下：

function addCart() {

//1.获取请求组装数组

$id = I('post.id');

$num = I('post.num'); // 商品数量

$color = I('post.color');

$size = I('post.size');

$cart = array('id'=>$id,'color'=>$color,'size'=>$size,'num'=>$num);

$arr = $\_SESSION['cart'][$id];

if(isset($arr)&&$arr['color']==$color&&$arr['size']==$size){

$arr['num'] = $arr['num']+$num;

$\_SESSION['cart'][$id] = $arr;

}else{

$\_SESSION['cart'][$id] = $cart;

}

///dump($\_SESSION['cart']);exit;

$this->success('添加购物车成功！',U('Shopcar/index'));

}

## 4.2 系统后台功能模块的实现

系统后台的功能模块有，注册用户与权限管理模块、栏目分类管理模块、商品内容管理模块、服务管理模块和其他参数管理模块。其界面和功能与用户管理模块和商品管理模块大同小异，相关的核心代码也类似，只是对数据库表操作的sql语句略有不同。下面给出，管理员登陆模块，用户管理模块，商品管理模块的主要功能模块的实现过程。

## 4.2.1管理员登录模块

管理员登陆界面的实现主要对应三个php文件，一个是登录界面的视图文件：Application/Admin/view/Login/index.html,一个是对输入信息进行验证和生成验证码文件：Application/Admin/Controller/LoginController.class.php

管理员登录模块的文字描述:

checklogin方法：

查询用户：通过I来获取前台提交过来的数据，实例化User模型，查询用户名，如果存在用户名，检查密码是否正确；是否是工作人员；

如果正确，就创建session

其对应的LoginController控制器实现生成验证码的核心代码如下：

function code($i=1) {

$verify = new \Think\Verify();

$verify->fontSize = 30;

$verify->length = 3;

$verify->useNoise = false;

$verify->useCurve = false;

$verify -> entry($i);

}

## 4.2.2后台首页的布局

后台首页采用了css+div的布局。

(后台首页页面:Application/Admin/view/index/index.html;

对应的css文件：Public/Admin/css/index.css)。后台首页，如图4-7所示。

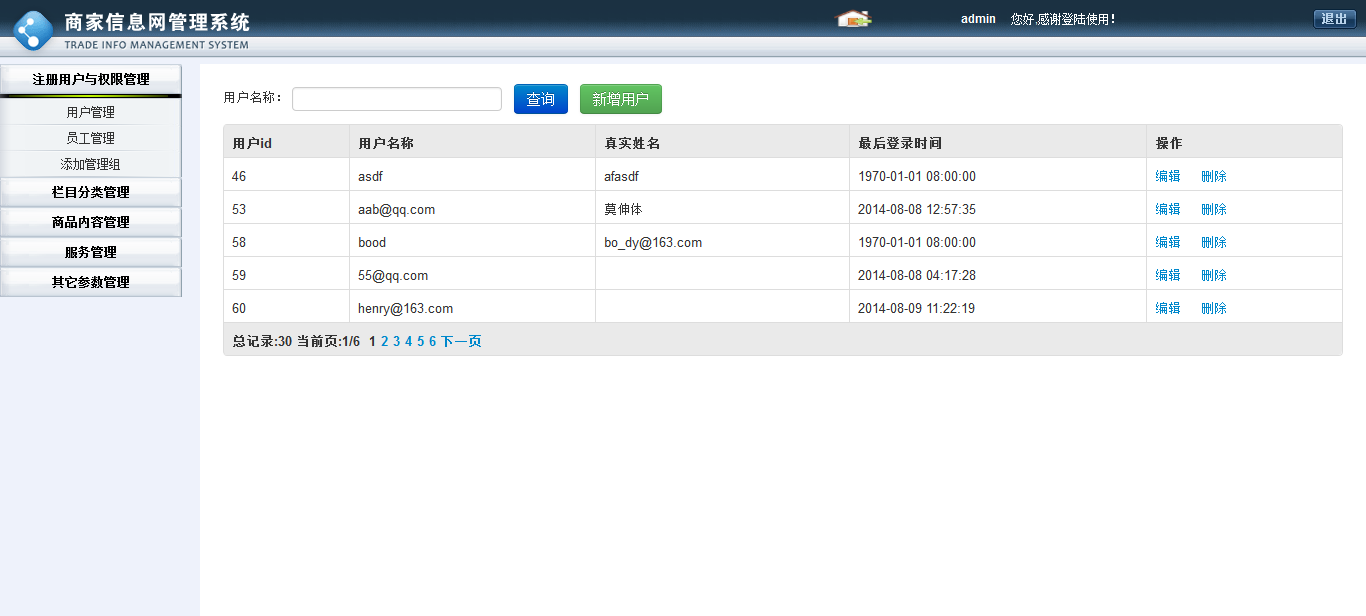


图4-7 后台首页

## 4.2.3权限管理模块

权限管理模块的实现主要使用两个文件，一个是权限管理的视图文件：Application/Admin/View/Role/index.html,另一个是实现权限管理模块的文件：Application/Admin/Controller/RoleController.class.php。权限管理页面，如图4-8所示。

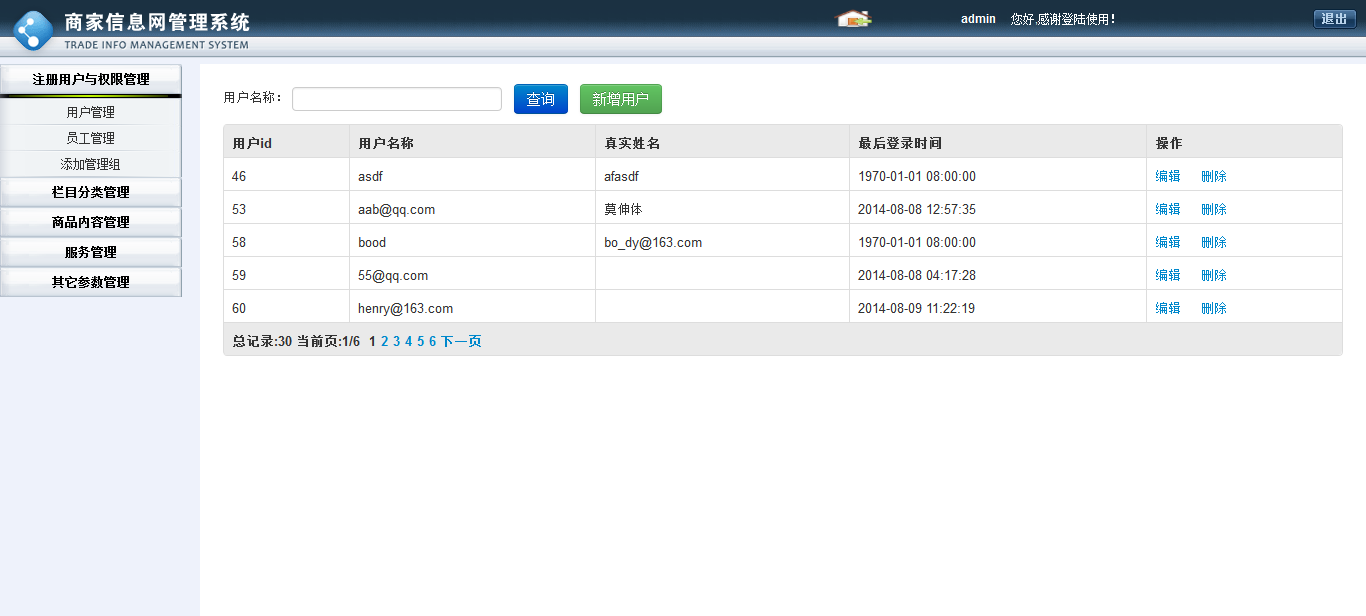


图4-8 权限管理页面

权限管理功能的文字描述：

通过I来获取前台提交过了的数据，实例化Role模型，add方法插入数据

其对应的RoleController控制器实现脚本的核心代码如下：

public function add\_do(){//添加分组

// $group =I('post.group');

$node = I('post.node');

$b = I('post.status');

$a = I('post.title');

$db = M('role');

$data['status']=$b;

$data['name']=$a;

$r = $db->add($data);

if($r){

//转换二维数组添加

$arr =array();

foreach ($node as $v) {

$arr[]=array('role\_id'=>$r,'node\_id'=>$v);

}

$ar= M('access')->addAll($arr);

if($ar){

$this->success('添加管理组成功',U('index'),1);

}

}else{

$this->error('添加管理组失败',U('add'),1);

}

}

## 4.2.4商品管理模块

商品管理模块的实现主要使用两个文件，一个是商品管理的视图文件：Application/Admin/View/Goods/index.html,另一个是实现商品管理模块的文件：Application/Admin/Controller/GoodsController.class.php。商品管理页面，如图4-9所示。



图4-9 商品管理页面

商品管理功能的文字描述：

添加商品的操作方法

1、实例化文件上传类

2、设置文件最大上传大小

3、设置文件上传的后缀名

4、设置文件上传的根路径

5、设置文件上传后的名字

6、创建子目录

7、给子目录命名

8、调用upload类中的upload方法上传文件

9、文件上传成功后，自定义文件名，例，20140726/53878fdf878.jpg

10、把上传成功的文件赋值给$\_POST,以便向数据库写入

11、用add方法向数据库写入数据

其对应的GoodsController控制器实现脚本的核心代码如下：

//添加商品的操作方法

public function insert(){

$goods = M('goods');

//实例化文件上传类

$upload = new \Think\Upload();

//设置文件最大上传大小

$upload -> maxSize = 3145728;

//设置文件上传的后缀名

$upload -> exts = array('jpg','gif','png','jpeg');

//设置文件上传的根路径

$upload -> rootPath = './Public/Uploads/';

//设置文件上传后的名字

$upload -> savename = array('uniqid','');

// 创建子目录

$upload -> autoSub = true;

//给子目录命名

$upload -> subName = array('date','Ymd');

//调用upload类中的upload方法上传文件

$info = $upload->upload();

if(!$info) {

//判断，如果上传失败后的错误

$this->error($upload->getError());

}else{

//文件上传成功后，自定义文件名，例，20140726/53878fdf878.jpg

$picpath = $info['0']['savepath'].$info['0']['savename'];

$picpath1 = $info['1']['savepath'].$info['1']['savename'];

//把上传成功的文件赋值给$\_POST,以便向数据库写入

$\_POST['pics'] = $picpath;

$\_POST['imgs'] = $picpath1;

$\_POST['addtime'] = time();

$\_POST['code'] = uniqid();

$\_POST['color'] = join(',',$\_POST['color']);

$\_POST['size'] = join(',',$\_POST['size']);

//向数据库写入数据

if($goods -> create()){

if($goods -> add()){

$this -> success('添加成功!','index');

}else{

$this -> error('添加失败！');

}

}else{

$this -> error('上传失败，请重新上传！');

}

}

# 第5章 总结和展望

## 5.1总结

论文分析了计算机网络的快速发展以及电子商务网站的现状，介绍了本项目的背景和研究价值，根据广州某公司实际的业务需求，提出了手工艺品网站的建设方案。 本文概述了手工艺品网上商城网站的主要任务和技术背景，分析了系统功能模块，并阐述了数据库设计的过程。然后，详细介绍了系统的具体设计与实现。最后，通过测试调试，系统基本能够满足需求分析和详细设计要求。

系统的基本功能已经实现，但是还是存在着许多的不足之处，需要对它进行进一步的完善和改进：

1. 平面设计水平不高，以至于界面比较呆板，不能让消费者有眼前一亮的感觉。
2. 数据库设计有很多冗余的地方，需进一步优化。
3. 对于网站的安全性与稳定性，需进一步优化。

## 5.2 展望

本系统还可以继续添加许多新鲜和使用的功能，但是本人的技术原因，一些功能暂时还不能实现，希望自己在将来可以不断地学习，解决掉着一些技术难题，争取把这个系统的不足完善上去。把它建设成一个强大的电子商务网站。

在本系统的设计和开发中涉及很多新的方法和技术，由于时间的关系，所以该系统还有许多不尽如人意的地方，这些都有待在以后的研究中进一步完善。

致 谢

经过两个月的努力，本系统已经开发成功。在本系统开发过程中，我要特别感谢我的指导老师覃忠台老师和我的同事孜孜不倦的指导和帮助。无论是在论文的选题、构思，还是在论文的研究方法以及成文定稿方面，我都得到了覃忠台老师悉心的教诲和无私的帮助。还有在我网站的设计和开发过程中遇到的很多技术问题都是在同事热情的指导和帮助下得以解决的。在此表示真诚的感谢和深深的谢意。

同时我还感谢母校的老师,谢谢他们为本系统提供的帮助.让我能够顺利地完成本系统的开发.

最后，向在百忙中抽出时间对本文进行评审并提出宝贵意见的各位专家表示衷心的感谢！

参考文献

1. 陈思含.浅谈新时代背景下的宠物经济发展[J].知识经济,2019(01):79+81.
2. 卫喆，陈争航.PHP5与MySQL5从入门到精通[M].北京:电子工业出版社，2008-8-1
3. 李开涌.[PHP MVC 开发实战](http://www.shangxueba.com/book/1790593.html)[M].北京：机械工业出版社,2013-07-01
4. 车开森林.浅谈CSS＋DIV在网页制作中的应用[J].计算机光盘软件与应用, 2012（2）.23-24
5. David Flanagan著；淘宝前端团队译.JavaScript权威指南（第6版）[M]. 北京：机械工业出版社，2011-04-01