

网上鲜花订购系统的设计与实现

摘要

网上鲜花购物系统是基于 web 技术实现的应用系统。通过鲜花订购系统，顾客不必在受时间、地点的约束，可以通过网络在任何地点、任意时间实现轻松购物。网上鲜花购物系统可以简化传统的购物模式，顾客可以足不出户就能轻松购物，使购物成为一种更轻松、更愉快的休闲方式。

本文主要介绍了网上鲜花订购系统的研究背景，对于设计思想和设计流程也做出了全面的叙述，在数据库创建思想以及各个数据表之间的具体关联等方面也做出了详细说明，并且具体剖析了系统各个功能的实现过程以及详细设计过程。该系统采用 B/S 架构，基于 PHP 技术实现，采用了 smarty 模板和 adodb 的较流行的技术。

关键词：PHP MySQL Smarty Adodb 网上信息管理

Online Flower Shopping System Based on PHP

Abstract

Online flower shopping system is based on the application of web technology systems. Through the online flower shopping system, Customers do not need to be constrained by time and place. They can through the network at any place, any time, for easy shopping. Online shopping system simplifies the traditional shopping patterns, to make shopping an easier and more enjoyable leisure.

This paper describes the background of online flower ordering system. It has also made a comprehensive statement about the design thinking and design process. And gives details about database creation thought and each specific association between the data tables. The paper analysed realization of various functions and detailed design process of the system. The system uses B / S Architecture, popular technologies such as smarty templates and adodb, and based on PHP technology.

Key Words: PHP MySQL Smarty Adodb Online information management

目录

| | |
|--------------------------|-----------|
| 1. 绪论 | 1 |
| 1.1 课题研究背景 | 1 |
| 1.2 课题研究目的 | 1 |
| 1.3 课题研究意义 | 2 |
| 1.4 课题使用的技术 | 2 |
| 1.5 论文结构 | 3 |
| 1.6 本章小结 | 3 |
| 2. 相关技术 | 5 |
| 2.1 PHP 技术的介绍 | 5 |
| 2.1.1 PHP 技术发展背景 | 5 |
| 2.1.2 PHP 技术的特点 | 5 |
| 2.2 Smarty 技术介绍 | 7 |
| 2.2.1 smarty 简介 | 7 |
| 2.2.2 smarty 语法 | 8 |
| 2.2.3 smarty 变量调节器 | 10 |
| 2.3 Adodb 技术 | 11 |
| 2.3.1 Adodb 技术介绍 | 11 |
| 2.3.1 Adodb 使用 | 12 |
| 2.4 本章小结 | 13 |
| 3. 系统分析 | 14 |
| 3.1 功能需求分析 | 14 |
| 3.2 可行性分析 | 16 |
| 3.2.1 技术可行性分析 | 16 |
| 3.2.2 经济上的可行性 | 17 |
| 3.2.3 实践可行性分析 | 17 |
| 3.3 软硬件需求 | 17 |
| 3.4 本章小结 | 17 |

| | |
|-------------------------|---------------------------|
| 4. 系统设计与实现 | 18 |
| 4.1 数据库设计 | 18 |
| 4.2 系统功能模块设计与实现 | 21 |
| 4.2.1 购物模块设计与实现 | 21 |
| 4.2.2 产品添加功能模块 | 26 |
| 4.2.3 订单管理功能模块 | 30 |
| 4.3 本章小结 | 32 |
| 5. 总结和展望 | 34 |
| 5.1 论文总结 | 34 |
| 5.2 今后工作展望 | 34 |
| 参考文献 | 36 |

1. 绪论

1.1 课题研究背景

伴随计算机技术在各行各业日益广泛和深入的应用，网络的概念早已深入人心。网络在各行业各业的发展战略中占据了重要的位置，成为商家不可分割的部分。商品的宣传已不止局限于电视和报纸，网络已成为商家展示自己的另一个舞台。商家建立网站，将商家各方面的宣传与服务展现于网络中，通过网络可实现如网上购物、信息查询等功能，这些在改变我们原有经营方式与经营理念的同时，也为商家带来了更高的效益。因此，对于商家来说，拥有一个属于自己的网络是至关重要的。

在美、日等信息化程度较高的国家和地区，网络商店发展速度迅猛。调查表明：美国的家庭已越来越习惯于在家中从网上购物。

在我国，网上购物从无到有也不过短短几年的时间。我国第一家网上购物发生在 1996 年，燕莎友谊商场首次通过网上商城售出一个景泰蓝，虽然贷款的支付不是在网上进行的，但是这毕竟为我国零售业奏出了网上购物的先声。

按照 CNNIC 发布的数字，2005 年上半年，中国网上购物的规模就达到 100 亿。2007 年是中国网络购物市场快速发展的一年，无论是 C2C 电子商务还是 B2C 电子商务市场交易规模都分别实现了 125.2% 和 92.3% 快速增长。网络购物年均增长 50%，中国网络购物的市场规模在 2009 年实际接近 1000 亿。预计在 2011 年底，中国网络购物市场规模将达到 4060 亿。

1.2 课题研究目的

此网上鲜花订购系统是为方便本市市民与店家的交流与沟通，便于商家与消费者之间进行联系和交流而提供的一个平台。利用网络资源优势和技术优势，通过提供完善的网上服务和规范会员的管理，以达到方便客户与商家的联系，更好的服务大众。

网上鲜花订购系统作为一种方便商家和客户之间联系的实用系统便在这样的情况下产生与发展起来的。目前 Internet 上的网上鲜花订购系统众多，他们提供的功能与服务都大同小异，在技术上也日趋成熟。网上鲜花订购系统是商家和客户之间进行交流和联系提供的一个平台。

该课题的目的便是在实现一个有自己特色的网上鲜花订购系统的同时，对当前的 web 开发技术有进一步的了解，熟练使用 smarty 模板和 adodb，使自己对 web 开发有进一步的了解。

1.3 课题研究意义

在互联网高速发展的今天，Internet 已成为人们获取、发布和传递信息的重要渠道，它在人们的日常工作、学习和生活中正发挥着越来越重要的作用。而网站作为互联网上信息交流的主要手段之一，其在商业领域应用的重要性更是不可言喻，它已成为商业组织、政府和企事业单位信息化建设中的重要组成部分，从而倍受人们的重视。另一方面，当前人们的生活节奏越来越快，越来越多的人希望可以在互联网通过点击鼠标完成复杂的事务，免除一些繁琐的过程。而本课题的设计正是网站在便捷式商业交流领域应用的一个较好体现。

网上鲜花订购系统，是一个将传统的商务交流场景转移到 Web 浏览器上的系统。该系统解决了传统的商务交易中买卖双方必须面对面对话的约束，交易双方只要可以连接网络，便可以在任何时间、任何地点下订单或接受订单，通过网站使交易方式更方便、更快捷。

1.4 课题使用的技术

此网上鲜花订购系统是基于 PHP 技术的，它是一门开源的、常用于 Web 编程的脚本语言。PHP 是英文超级文本预处理语言（PHP:Hypertext Preprocessor）的缩写，它是一种允许于服务器端、跨平台、HTML 内嵌式的脚本语言。Php 作为一门专业的 web 开发语言，已经被越来越多的开发者使用。

此网上鲜花订购系统主要采用 B/S 结构，B/S 结构是 WEB 兴起后的一种网络结构模式，WEB 浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。客户机上只要安装一个浏览器（Browser），如 Netscape Navigator 或 Internet Explorer，服务器安装 MySQL、Sybase、Informix 或 SQL Server 等数据库。浏览器通过 Web Server 同数据库进行数据交互。

此外，此系统在实现过程中使用到了 php 界流行的 smarty 模板。Smarty 是一个使用 PHP 写出来的 PHP 模板引擎，目的是要使 PHP 程序同美工分离，当程序员改变程序的逻辑内容时不会影响到美工的页面设计，美工重新修改页面

时不会影响到程序的程序逻辑，这在多人合作的项目中显的尤为重要。

最后，在数据库方面我们使用了 adodb 数据库引擎类。ADODB 是 Active Data Objects Data Base 的简称，它是一种 PHP 存取数据库的中间函式组件。虽然 PHP 是建构 Web 系统强有力的工具，但是 PHP 存取数据库的功能，一直未能标准化，每一种数据库，都使用另一种不同且不兼容的应用程序接口 (API)。为了填补这个缺憾，因此才有 ADODB 的出现。一旦存取数据库的接口予以标准化，就能隐藏各种数据库的差异，若欲转换至其它不同的数据库，将变得十分容易。

1.5 论文结构

本文介绍了[开发网上花店订购](#)系统所使用的 PHP 技术以及[网上花店订购](#)系统的开发过程，主要章节安排如下：

第 1 章介绍了网上鲜花订购系统的研究背景、目的和意义以及论文结构。

第 2 章介绍了本系统所采用的主要技术。首先我们介绍了 php 技术，然后将该技术同时下相同的 web 技术进行了比较，接着我们介绍了[了子 smarty](#) 模板及其语法，最后我们介绍 adodb 类库。

第 3 章介绍了本系统的系统分析。作为网上商铺系统开发的第一个主要阶段，结合实际情况，确定了系统的功能需求和软硬件需求等，对实现本系统的可行性进行了分析，为后续的系统设计奠定基础。

第 4 章介绍了系统设计与实现。按照软件工程的思想，根据概要设计和详细设计，进行了系统功能设计和实现，还介绍了数据库的设计与实现，给出了系统设计的流程图和实现的界面。

第 5 章对毕业设计所作的工作进行总结和展望，并对系统存在的问题进行说明。

1.6 本章小结

本章从实际出发，重点介绍了网上鲜花订购系统的研究背景，在此基础上探讨了开发基于 PHP 技术的网上鲜花订购系统的重要作用，总结得出了本课题的研究目的和意义，然后介绍了本系统使用的一些相关技术，最后介绍了论文

的结构。本章是整个研究与设计工作的出发点。

2. 相关技术

2.1 PHP 技术的介绍

2.1.1 PHP 技术发展背景

PHP 是英文 Hypertext Preprocessor（超级文本预处理语言）的缩写。PHP 是一种在服务器端执行的嵌入 HTML 文档的脚本语言，语言的风格类似于 C 语言，现在被很多的网站编程人员广泛的运用。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创新的语法。用 PHP 做出的动态页面与其他的编程语言相比，PHP 是将程序嵌入到 HTML 文档中去执行，执行效率比完全生成 HTML 标记的 CGI 要高许多，它可以比 Perl 更快速的执行动态网页。

PHP 于 1994 年创建，当时 PHP 还只是一个简单的用 Perl 语言编写的程序。后来用 C 重写后加入了数据库访问的支持。1995 年发表第一个版本。随后，越来越多的网站开始使用 PHP，并且要求加入一些新的特性，如循环和数组等，在 1995 年 PHP2.0 正式发布，该版本加入了对 mysql 的支持。1997 年第三个版本开始开发，并最终命名为 PHP3。2000 年，PHP4.0 问世，PHP4.0 整个脚本程序的核心大幅变更，让程序的执行速度更快，满足更快的要求，PHP4.0 更换了引擎，采用 Zend 引擎，使用了更有效的编译—执行方式，而且 PHP4.0 强化了面向对象的功能、引入了 PDO。时至今日，PHP 已经到了 PHP5 的时代，PHP5 更成熟、更稳定，对面向对象的支持也更好。今天有很多大型网站在使用 PHP，比如 Yahoo, Facebook,而且超过 2 千万的站点更在运行着 PHP。而且令所有 PHPer 兴奋的是 PHP6 正在开发中，在不久之后就会与大家见面。

2.1.2 PHP 技术的特点

- 开放的源代码：所有的 PHP 源代码都可以得到
- PHP 是免费的
- 基于服务器端：由于 PHP 是运行在服务器端的脚本
- PHP 是跨平台的：可以运行在 UNIX、LINUX、WINDOWS 下
- 可嵌入 HTML
- 简单的语言：PHP 的语法类似 C 语言，简单、易懂
- 效率高：PHP 消耗相当少的系统资源

- 图像处理：用 PHP 可以动态创建图像
- 面向对象：在 PHP4,PHP5 中，面向对象方面都有了很大的改进

2.1.3 PHP和ASP、JSP比较

当前，在 Web 开发界还有 ASP 和 JSP 这两个开发语言，下面将对这 3 中语言的特点进行比较。

2.1.3.1 ASP 技术

ASP 是 Active Server Page 的缩写，意为“动态服务器页面”。ASP 是微软公司开发的代替 CGI 脚本程序的一种应用，它继承了微软产品的一贯传统，只能运行于微软的 IIS 上，对平台的支持不够。另外，ASP 的语法非常类似 VB，语言非常简单，因此其代码也简单易懂，结合 HTML 代码，可快速地完成网站的应用程序。它可以与数据库和其它程序进行交互，是一种简单、方便的编程工具。ASP 的网页文件的格式是.asp，现在常用于各种动态网站中。

ASP 是一种服务器端脚本编写环境，可以用来创建和运行动态网页或 Web 应用程序。ASP 网页可以包含 HTML 标记、普通文本、脚本命令以及 COM 组件等。利用 ASP 可以向网页中添加交互式内容（如在线表单），也可以创建使用 HTML 网页作为用户界面的 Web 应用程序。与 HTML 相比，ASP 网页具有以下特点：

- 利用 ASP 可以实现突破静态网页的一些功能限制，实现动态网页技术；
- ASP 文件是包含在 HTML 代码所组成的文件中的，易于修改和测试；
- 服务器上的 ASP 解释程序会在服务器端执行 ASP 程序，并将结果以 HTML 格式传送到客户端浏览器上，因此使用各种浏览器都可以正常浏览 ASP 所产生的网页；
- ASP 提供了一些内置对象，使用这些对象可以使服务器端脚本功能更强。例如可以从 Web 浏览器中获取用户通过 HTML 表单提交的信息，并在脚本中对这些信息进行处理，然后向 Web 浏览器发送信息；
- ASP 可以使用服务器端 ActiveX 组件来执行各种各样的任务，例如存取数据库、发送 Email 或访问文件系统等；
- 由于服务器是将 ASP 程序执行的结果以 HTML 格式传回客户端浏览器，

因此使用者不会看到 ASP 所编写的原始程序代码，可防止 ASP 程序代码被窃取；

- 方便连接 ACCESS 与 SQL 数据库；

2.1.3.2 JSP 技术

JSP (JavaServer Pages) 是 Sun 公司推出的一种动态网页技术。JSP 技术是以 Java 语言作为脚本语言的，熟悉 JAVA 语言的人可以很快上手。

JSP 本身虽然也是脚本语言，但是却和 PHP、ASP 有着本质的区别。PHP 和 ASP 都是由语言引擎解释执行程序代码，而 JSP 代码却被编译成 Servlet 并由 Java 虚拟机执行，这种编译操作仅在对 JSP 页面的第一次请求时发生。因此普遍认为 JSP 的执行效率比 PHP 和 ASP 都高。

JSP 是一种服务器端的脚本语言，最大的好处就是开发效率较高，JSP 可以使用 JavaBeans 或者 EJB (Enterprise JavaBeans) 来执行应用程序所要求的更为复杂的处理，但是这种网站架构因为其业务规则代码与页面代码混为一团，不利于维护，因此并不适应大型应用的要求，取而代之的是基于 MVC 的 Web 架构。MVC 的核心思想是将应用分为模型、视图和控制器三部分。模型是指应用程序的数据，以及对这些数据的操作；视图是指用户界面；控制器负责用户界面和程序数据之间的同步。通过 MVC 的 Web 架构，可以弱化各个部分的耦合关系，并将业务逻辑处理与页面以及数据分离开来，这样当其中一个模块的代码发生改变时，并不影响其他模块的正常运行，所以基于 MVC 的 Web 架构更适应于大型应用开发的潮流。

因此，不少国外的大型企业系统和商务系统都使用以上的 MVC 架构，能够支持高度复杂的基于 Web 的大型应用。

以上两节简单的介绍了另外两种 Web 技术 ASP 和 JSP，通过比较可以看出 PHP 对平台的支持上明显好于另外两种，但是执行效率低于 JSP。

2.2 Smarty 技术介绍

2.2.1 smarty 简介

Smarty 是一个使用 PHP 写出来的模板引擎，是目前业界最著名的 PHP 模板引擎之一。它分离了逻辑代码和外在的内容，提供了一种易于管理和使用的方法，用来将原本与 HTML 代码混杂在一起 PHP 代码逻辑分离。简单的讲，

目的就是要使 PHP 程序员同前端人员分离，使程序员改变程序的逻辑内容不会影响到前端人员的页面设计，前端人员重新修改页面不会影响到程序的程序逻辑，这在多人合作的项目中显的尤为重要。早在 1999 年后期，就已经有人开始为模板引擎写说明文档。在完成这个文档之后，人们用 C 写了一个模板引擎，并希望被包含到 PHP 里去。在撞上了许多的技术难题的同时，“什么是模板应该做的，什么不该做”这个问题，也被热烈的讨论着。讨论最终决定应该用 PHP 将模板引擎写成一个类，让任何觉得合适的人使用它，从此就有了 Smarty。

Smarty 具有以下特性：

速度：采用 Smarty 编写的小程序可以获得最大速度的提高，这一点是相对于其它的模板引擎技术而言的

编译型：采用 Smarty 编写的程序在运行时要编译成一个非模板技术的 PHP 文件，这个文件采用了 PHP 与 HTML 混合的方式，在下一次访问模板的时候将 WEB 请求直接转换到这个文件中，而不再进行模板重新编译

缓存技术：Smarty 选用的一种缓存技术，它可以将用户最终看到的 HTML 文件缓存成一个静态的 HTML 文件中来，这相当于调用一个静态的 HTML 文件

插件技术：Smarty 可以自定义插件，插件实际就是一些自定义的函数

模板中可以使用 `if/else if/else/endif`：在模板文件使用判断语句可以非常方便的对模板进行格式重排

2.2.2 smarty 语法

作为一种模板语言，smarty 有着自己的语法。Smarty 有自己的内建函数和变量调节器，使用 smarty 甚至可以编写自己的插件，下面将对此分别介绍。

Smarty 自带一些内建函数，内建函数是模板语言的一部分，开发者不能创建名称和内建函数一样的自定义函数,也不能修改内建函数。通过在模板中使用这些内建函数，可以实现复杂的逻辑，如引入其他模板、遍历数组、捕获模板输出的内容等。

下面将简单介绍 smarty 的内建函数，主要包括 `include` 函数、`if` 条件语句、`section` 函数和 `foreach` 函数。

(1) include 函数

Include 函数用于在当前模板中包含其他模板，使用方法如下所示：

```
<{include file=" header.tpl" assign=$title }>
```

以上代码中，file 参数指定在当前模板中引入的模板文件，assign 参数指定从当前模板传递变量给被引入模板。

通常，我们的网站具有相同的布局，例如拥有相同的顶部菜单栏和底部版权声明，可以将顶部菜单栏和底部版权声明写在单独的模板中，然后引用该模板，以此可以实现牵一发而动全身的效果。

(2) if 条件语句

在模板中通过使用 if 条件语句可以实现不同的模板显示效果。Smarty 中的 if 语句和 php 中的 if 语句一样灵活易用，并增加了几个特性以适宜模板引擎。

。if 必须与 与手/if 成对出现； 可以使用 else 和 elseif 子句；可以使用以下条件修饰词：eq、ne、neq、gt、lt、lte、le、gte、ge、is even、is odd、is not even、is not odd、not、mod、div by、even by、odd by、==、!=、>、<、<=、>=。if 条件语句的使用实例如下所示：

```
{if $name eq "Fred"}
    Welcome Sir.
{elseif $name eq "Wilma"}
    Welcome Ma'am.
{else}
    Welcome, whatever you are.
{/if}
{if $var is even}
    ...
{/if}
{if $var is odd}
    ...
{/if}
{if $var is not odd}
```

```
...
{/if}
{if $var is even by 3}
...
{/if}
```

(3) section 函数

Smarty 模板中的 section 函数用于遍历数组中的数据，用法如下所示：

```
//遍历一维数组
<{section loop=$arr1 name=s start}>
Id:<{$arr1[s]}><br>
</section>>

//遍历二维数组
<{section name=customer loop=$contacts }>
Id: <{$ contacts [customer].id}>
Name: <{$ contacts [customer].name}>
</section>>
```

在上述代码中，loop 参数指定要遍历的数组，name 参数指定该循环的名称。在 smarty 中通过 ' . 键值' 的方式访问数组的值。

(4) foreach 函数

foreach 函数是 smarty 中处理循环的另一种方案，foreach 函数用于处理简单数组，它的缺点是只能处理简单的数组。foreach 函数的用法如下所示：

```
{foreach name=outer item=contact from=$contacts}
  {foreach key=key item=item from=$contact}
    {$key}: {$item}<br>
  {/foreach}
{/foreach}
```

在以上代码中，from 参数表示要遍历的数组，item 参数表示当前处理元素的变量名称，name 参数表示该循环的名称，用于访问该循环，key 参数表示当前处理元素的键名。

2.2.3 smarty 变量调节器

Smarty 模板内置的一些操作函数，我们称之为变量调节器，变量调节器用于操作变量、自定义函数和字符。在 smarty 语法中使用

“|” 应用变量操作符，多个参数用“:” 分隔开来。

- 下面将介绍几个常用的变量调节器：
- (1) truncate 变量调节器
- 该变量调节器用于安装指定长度截取字符串，使用方法如下所示：

● `{$articleTitle|truncate:30:"...":true}}`

- 在上面的代码中\$articleTitle 是需要截取的字符串，第一个参数 30 是截

取后的长度，第二个参数作为追加在截取字符串后面的文本字符串，该追加字符串被计算在截取长度中，第三个参数决定是截取到词的边界（false）还是精确到字符（true）。

- (2) date_format 变量调节器
- 该变量调节器用于按照一定的规则格式化时间格式，使用方法如下所示：

● `{$smarty.now|date_format:"%H:%M:%S"}`

- 在以上代码中，\$smarty.now 是表现时间的戳，参数表示要格式化的时间格式，%H 代表小时，%M 代表分钟数，%S 代表秒数。

- (3) strip_tags 变量调节器

- strip_tags 变量调节器用于去除 HTML 标签，用法如下所示：

● `{$articleTitle|strip_tags}`

- 在以上代码中，\$articleTitle 代表要过滤的字符串。

- (4) indent 变量调节器

- indent 变量调节器用于在每行缩进字符串，用法如下所示：

● `{$articleTitle|indent:2:" "}`

- 在上述代码中，第一个参数代表要缩进的字符数，第二个参数表示用那个符号代替缩进的字符。

2.3 Adodb 技术

2.3.1 Adodb 技术介绍

Adodb是Active Data Objects Data Base 的简称，它是一种PHP存取数据库的中间函数组件。

虽然PHP是建构Web系统强有力的工具，但是PHP存取数据库的功能，一直未能标准化，每一种数据库，都使用另一种不同且不兼容的应用程序接口(API)。

为了填补这个缺憾，因此才有ADODB的出现。一旦存取数据库的接口予以标准化，就能隐藏各种数据库的差异，若欲转换至其它不同的数据库，将变得十分容易。

目前 ADODB最新版本是V5.11，支持的数据库种类非常地多，例如：MySQL, PostgreSQL, Interbase, Informix, Oracle, MS SQL 7, Foxpro, Access, ADO, Sybase, DB2以及一般的ODBC(其中 PostgreSQL、Informix、Sybase 的driver是由自由软件社群发展之后贡献出来的)。

使用ADODB最大的优点之一是，不管后端数据库如何，存取数据库的方式都是一致的，开发设计人员不必为了某一套数据库，而必须再学习另一套不同的存取方法，这大大减轻开发人员的知识负担，过去的知识往后仍可继续使用，转移数据库平台时，程序代码也不必做太大的更动。

2.3.1 Adodb 使用

Adodb主要用到两个对象，-ADOConnection 对象和RecordSet对象，ADOConnection对象负责数据库的连接，发送 sql 命令等；RecordSet对象主要是得到查询出来的记录集。

— (1) 实例化Adodb对象

Adodb使用NewADOConnection类实例化一个Adodb连接对象，使用方法如下所示：

```
$db = NewADOConnection($database_type);
```

在上述代码中，\$database_type表示数据库类型，可能的值有access、db2、mssql、mysql等，几乎包含了当前所有主流数据库。

(2) 使用Adodb对象连接数据库

在Adodb中连接数据库使用Connect()方法，使用方法如下所示：

```
$connect = $db -> Connect($host, $user, $password, $database);
```

在上述代码中，参数\$host表示连接的主机名，通常为IP地址或域名，\$user参数为用户名，\$password参数为密码，\$database参数表示数据库名。该方法成功时返回一个ADOConnection对象，错误时返回false。

(3) 使用Adodb执行SQL语句

Adodb使用Execute()方法执行SQL语句，具体过程如下所示：

```
$result = $db -> Execute( 'select * from user' );
```

在上述代码中，Execute()方法执行sql语句，执行成功时返回一个Recordset对象，否则返回false；

(4) Adodb其他方法介绍

Adodb提供了很多的方法，这些方法几乎可以帮助我们实现所有的需求，限于篇幅的要求，在这里我们不能向读者介绍Adodb所有的方法，因此下面我们介绍几个最基本的方法：

1 Affected_Rows()方法

该方法返回最后一个sql命令所影响的记录的行数。

2 SelectLimit(\$sql, \$nrows, \$offset)方法

该方法根据参数返回指定的记录，第二个参数表示需要返回的行数，第三个参数设置从哪一行开始查询。该方法成功时返回一个Recordset对象，失败是返回false,该方法在分页时相当有用。

3 MoveNext()方法

该方法使指针在Recordset对象中下移一个，即指向下一条记录。

4 MoveFirst()方法

该方法将Recordset对象中的指针返回到对象中的第一条记录。

5 Insert_ID()方法

该方法返回最后一次执行插入语句的主键值。

6 RecordCount()方法和RowCount()方法

这两个方法返回sql指令存取到的记录的条数。

7 EOF属性

EOF是Recordset对象的属性，该属性标识了指针是否指向Recordset对象的最后一条记录，如果是返回true，否则返回false。

2.4 本章小结

本章主要介绍了php，并将其与ASP、JSP进行了对比，之后介绍了在实现系统时用到的两个主要技术，分别是smarty模板和Adodb数据库类库。通过该章的介绍使读者对该系统中使用到的技术有所了解。

3. 系统分析

3.1 功能需求分析

基于 PHP 的网上鲜花订购系统，是为了在互联网上为客户提供服务而构建的系统。该系统的总体设计是要提供给上网购物者一个效率高、能容丰富的网络空间、包括用户可以在线搜索鲜花、网上购物等种种功能。本课题研究的网上鲜花订购系统，基本可以实现当前其他网上购物网店的所有功能，如注册、登录、购物、后台管理等。

在该系统中用户主要分为三种：游客、注册会员和管理员，三种身份的用户分别有自己的功能。图 3-1 向我们展示了在该系统中基于用户身份的功能列表：

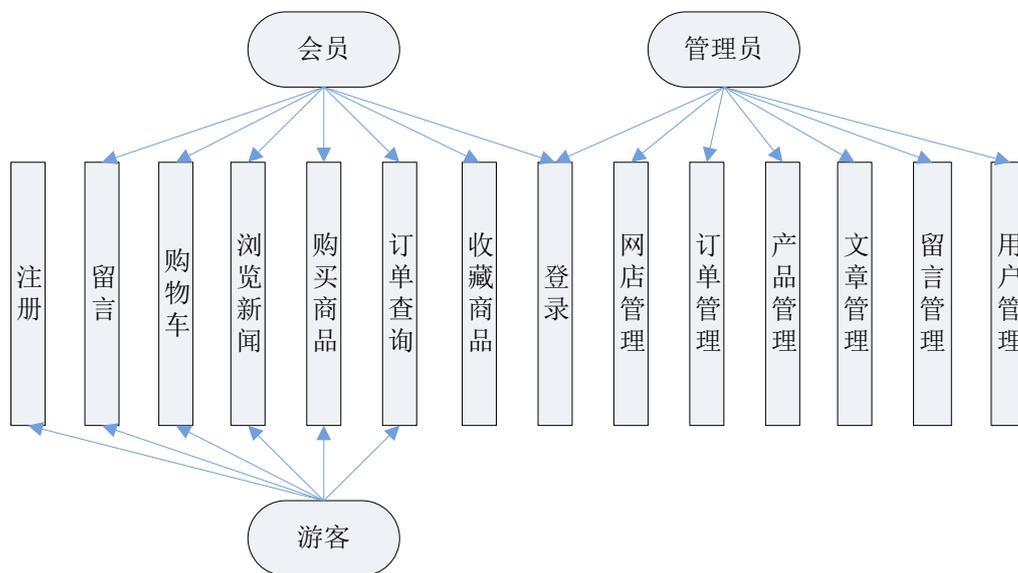


图3-1 基于用户身份的功能列表

从图 3-1 可以看出对于游客，主要有以下 4 个功能：

(1) 注册功能

该功能主要实现注册功能，注册完成后用户成为本站会员，享受进一步的待遇，如商品打折、收藏商品等。在该功能中，要求用户填写用户的基本资料，

如邮箱、地址、电话、手机等信息，该功能要求能对用户输入的信息进行有效性验证，如判断邮箱格式、电话格式和手机格式等。

(2) 留言功能

该功能允许用户在浏览网店商品的同时对商品进行留言或评价。在留言过程中，用户需要填写邮箱、评价和评价等级。

(3) 购物车功能

该功能是该主要为方便客户在浏览商品时，一旦发现心仪商品可以将该商品加入购物车中，放在购物车中的商品可以同时付款，以此可以达到方便用户购物的效果。

(4) 浏览新闻功能

该功能允许用户浏览网店发布的新闻、通知等信息。

(5) 购买商品功能

该功能主要实现最终购物的功能，在支付柜台中，用户可以选择支付单件商品或支付全部商品。在选择完购买方式后，系统会要求客户填写订单。订单的内容主要包括收件人姓名、地址、电话、手机和送货时间。

(6) 订单查询功能

该功能是为方便客户及时了解到订单动向，客户可以在系统首页使用该功能，通过该功能，客户可以查看自己购买的商品是否已发货等信息。

对于注册会员，除了可以使用游客的基本功能之外，还有自己特有的两个功能，这些功能如下：

(1) 登录功能

该功能提供给注册用户登录的功能，用户通过登录页面进行登录，登录功能要求能对用户输入的账号、密码做出正确判断。

(2) 收藏功能

该功能允许登录的用户添加自己感兴趣的商品，该功能要能对身份进行验证，即当游客使用该功能时提醒用户无法收藏。此外，该功能还要求能对会员收藏的产品唯一性进行验证，即当会员第二次收藏该产品时提醒会员该商品已收藏过。

对于管理员，在系统中拥有最高的权限，拥有所有后台管理功能的权限。

这些管理功能主要包括以下几点：

(1) 登录功能

该功能同样实现用户的登录，但是在该登录功能中要求对用户的权限进行验证，只有拥有管理权限的用户才能在此登录。

(2) 网店管理功能

该功能主要实现对网店的全局性信息进行管理，如设置网店名称，设置网店 URL，更换模板，设置网站地址、电话、QQ 等信息。

(3) 订单管理功能

该功能主要包括三个子功能，查询最新订单、查询历史订单和订单追踪功能。查询最新订单功能主要向管理员展示所有新的订单，在该功能中，管理员可以更改订单状态，如将订单状态更改为已发货、或已接收状态。查询历史订单功能主要向管理员展示所有非最新的订单，在该功能中，管理员可以将已发货的订单的状态更改为已接收状态。订单查询更能允许管理员通过输入订单号对订单进行查询，查询结果包括订单的状态和订单的基本信息。

(4) 产品管理功能

该功能包含产品类别管理和产品管理功能两个子功能。在产品类别管理功能中，允许管理员对产品的类别进行增、删、改、查的操作。在产品管理功能中，允许对产品进行增、删、改、查的操作。

(5) 文章管理功能

该功能主要实现新闻、通知等文章的发布与管理，同时，该功能还涉及文章类别管理的功能。

(6) 留言管理功能

该功能允许管理员与客户进行沟通。在该功能中。管理员可以浏览所有用户的留言，同时，该功能可以对用户的留言做出回复或者屏蔽用户的留言。

(7) 用户管理功能

该功能允许管理员查询所有用户信息、更改用户信息和删除用户账号，同时该功能还包括添加管理员功能。

3.2 可行性分析

3.2.1 技术可行性分析

该网上鲜花订购系统采用 php 技术开发，采用 smarty 模板引擎和 adodb 数据库类库。Php 是一门门槛比较低的 web 开发语言，其语法类似于 C 语言，对于有计算机开发经验的人来说都可以轻松上手。此外，smarty 模板是 php 界最成熟、最流行的模板引擎，有很多的网站都是采用 smarty 模板开发，互联网上有很多 smarty 模板的文档和社区，对我们的开发提供了支持。最后，Adodb 同样是一个成熟的数据库连接类库，它使用简单，操作方便，对很多实用方法进行了封装。基于以上三点考虑我们认为我们的网上鲜花订购系统在技术上是可行的。

3.2.2 经济上的可行性

该课题所研究的网上鲜花订购系统无需开发经费，使用的服务器均为开源免费软件，可任意使用而无需花费任何费用。在服务器端，只需要一台能承受一定访问压力的服务器。在客户端方面，客户需要一台可连接 internet 的 PC 机即可。因此，该系统从经济角度而言也是可行的。

3.2.3 实践可行性分析

本系统的开发适应了时代的发展要求，当前人们越来越倾向于网上购物，因为网上购物的方便、快捷是传统购物方式无法比拟的。此外，当前互联网上的网店犹如百花争鸣，网店必将成为未来消费的主要方式之一。因此从实践可行性角度来说，该系统同样是可行的。

3.3 软硬件需求

根据我们在3.1节的功能分析和3.2节的可行性分析，本节对本系统的软硬件需求进行了分析说明，主要包括以下几个方面：

- (1) 在Windows/Linux平台运行；
- (2) 使用apache作为Web服务器；
- (3) 后台数据库使用mysql；
- (4) 前台应用程序用php开发；
- (5) 客户端使用个人计算机即可。

3.4 本章小结

本章主要介绍了网上鲜花订购系统的系统分析，包括了对系统的需求分析，列出系统需要实现的各个功能模块，接着对本系统的可行性进行了分析，确定

系统设计的可行行，最后介绍了开发本系统要求的软硬件要求。

4. 系统设计与实现

根据第3章的功能需求分析，在本章我们将对系统的设计与实现给出详细的介绍。首先，我们会介绍数据库的设计，之后我们会选择系统中有针对性的功能模块做介绍，本系统的功能模块较多，限于篇幅的限制，我们只介绍有特点的几个模块。

4.1 数据库设计

数据库是一切计算机应用程序的基础，在本节中我们将详细介绍我们系统中的数据库设计。

在该系统中一共涉及11张表，下表列出了这些表的表名及其简单描述：

表4-1 数据表及描述

| 表名 | 描述 |
|-----------------|------------------------------|
| articlecategory | 文章分类表，记录文章类别名 |
| article | 文章表，记录文章标题、内容等信息 |
| category | 产品分类表，记录产品类别名 |
| goods | 产品表，记录产品的所有信息，如名称、价格、颜色等 |
| image | 产品图片图，记录产品的图片信息，包括产品id、图片地址等 |
| list | 订单内容详情表，记录每个订单包含的产品及个数等信息 |
| message | 留言表，记录用户留言及留言回复信息 |
| order | 订单表，记录订单信息，包括订单号、收件人、送货时间等 |
| storebox | 收藏夹表，记录会员收藏的商品信息 |
| user | 用户表，记录用户所有信息，用户名、密码、权限等 |
| Site_config | 网站配置表，网站的配置信息记录在此表中 |

表3-1给出了系统的数据表的信息及简单的功能描述，下面将对其中关键的表给出具体介绍。

4.1.1 产品类别表设计

产品类别表记录的产品类别的信息，同时该表还记录了产品子类与产品父类之间的关系，该表的结构设计如图4-1所示：

| 名 | 类型 | 长度 | 十进位 | 允许空?.. | |
|------|---------|----|-----|--------------------------|---|
| ▶ id | int | 11 | 0 | <input type="checkbox"/> |  |
| name | varchar | 20 | 0 | <input type="checkbox"/> | |
| fid | int | 11 | 0 | <input type="checkbox"/> | |

图4-1 产品分类结构表设计图

从图4-1可以看出，在该表中只有3个字段，id为该表的主键，该字段递增；name字段表示该产品类别的名称；fid字段表示该产品的父类，此字段为该表的外键，关联的字段为该表的主键，如果该字段为1，说明该分类的父类是id为1的分类。

4.1.2 产品表设计

该表记录了产品的基本信息，如产品编号、名称、类别、用途表述、价格等信息，具体设计如图4-2所示：

| 名 | 类型 | 长度 | 十进位 | 允许空?.. | |
|-------------|---------|-----|-----|-------------------------------------|---|
| ▶ id | int | 11 | 0 | <input type="checkbox"/> |  |
| no | varchar | 11 | 0 | <input checked="" type="checkbox"/> | |
| name | varchar | 20 | 0 | <input type="checkbox"/> | |
| category_id | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| use | varchar | 50 | 0 | <input checked="" type="checkbox"/> | |
| units | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| gprice | varchar | 7 | 0 | <input checked="" type="checkbox"/> | |
| mprice | varchar | 7 | 0 | <input checked="" type="checkbox"/> | |
| number | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| color | varchar | 20 | 0 | <input checked="" type="checkbox"/> | |
| material | varchar | 100 | 0 | <input checked="" type="checkbox"/> | |
| packaging | varchar | 100 | 0 | <input checked="" type="checkbox"/> | |
| florid | varchar | 200 | 0 | <input checked="" type="checkbox"/> | |
| vtimes | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| btimes | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| describe | text | 0 | 0 | <input checked="" type="checkbox"/> | |
| create_at | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| isshowed | tinyint | 1 | 0 | <input checked="" type="checkbox"/> | |
| ishot | tinyint | 1 | 0 | <input checked="" type="checkbox"/> | |
| recommended | tinyint | 1 | 0 | <input checked="" type="checkbox"/> | |

图4-2 产品表结构设计图

如图4-2所示，id表示该表的主键，no字段表示产品编号，name字段表示产品名称，category_id表示产品分类，该字段是该表的外键，关联产品分类表，

use字段表示产品用途，units字段表示产品单位，gprice表示产品市场价，mprice表示会员价，isshowed表示该产品是否在前台显示，1为显示，0不显示，ishot表示该产品是否热卖，1为是，0为否，recommended表示是否为店长推荐，1为是，0为否。

4.1.3 订单表设计

订单表记录了订单的基本信息，该表的结构设计如图4-3所示：

| 名 | 类型 | 长度 | 十进位 | 允许空?.. | |
|-----------|---------|----|-----|-------------------------------------|--|
| id | int | 11 | 0 | <input type="checkbox"/> | |
| no | varchar | 9 | 0 | <input checked="" type="checkbox"/> | |
| user_id | varchar | 15 | 0 | <input checked="" type="checkbox"/> | |
| ids | varchar | 30 | 0 | <input checked="" type="checkbox"/> | |
| user | varchar | 10 | 0 | <input checked="" type="checkbox"/> | |
| ispayed | tinyint | 4 | 0 | <input checked="" type="checkbox"/> | |
| status | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| create_at | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| tel | varchar | 13 | 0 | <input checked="" type="checkbox"/> | |
| address | varchar | 50 | 0 | <input checked="" type="checkbox"/> | |
| time | date | 0 | 0 | <input checked="" type="checkbox"/> | |
| phone | varchar | 11 | 0 | <input checked="" type="checkbox"/> | |

表4-3 订单表结构设计图

如图4-3所示，在该表中，no字段表示订单号，user_id字段表示购买者id或购买者ip,ids字段表示该订单所购买的产品的id，user表示收货人姓名，status表示该订单的当前状态，0表示未发货，1表示已发货，2表示买家已收获。

4.1.4 留言表设计

该表记录了用户的留言以及管理员的回复信息，具体结构设计如图4-4所示：

| 名 | 类型 | 长度 | 十进位 | 允许空?.. | |
|----------|----------|------|-----|-------------------------------------|--|
| id | int | 11 | 0 | <input type="checkbox"/> | |
| user | varchar | 45 | 0 | <input checked="" type="checkbox"/> | |
| goods_id | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| date | datetime | 0 | 0 | <input checked="" type="checkbox"/> | |
| content | varchar | 1000 | 0 | <input checked="" type="checkbox"/> | |
| fid | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| level | int | 11 | 0 | <input checked="" type="checkbox"/> | |
| email | varchar | 25 | 0 | <input checked="" type="checkbox"/> | |

图4-4 留言表结构设计图

如图4-4所示，在留言表中，user字段表示留言者的id或ip，goods_id表示留

言针对的产品，fid表示该留言的回复对象，若该字段为空，表示该条留言是最顶级的留言，若不为空，假设为3，则表示该条留言是对id为3的留言的回复，level字段表示对商品评价的等级。

4.2 系统功能模块设计与实现

结合4.1节的数据库介绍，在此节，我们将详细叙述该系统中的关键或有技术特点的功能模块的设计与实现，其余模块的实现流程或原理与这些模块基本类似，因此不做过多的赘述。我们要介绍的模块主要包括购物模块、产品添加功能模块和订单管理模块。

4.2.1 购物模块设计与实现

购物模块是一个购物网站的核心模块，因此我们首先给出购物模块的设计与实现过程，购物模块流程图如图4-5所示：

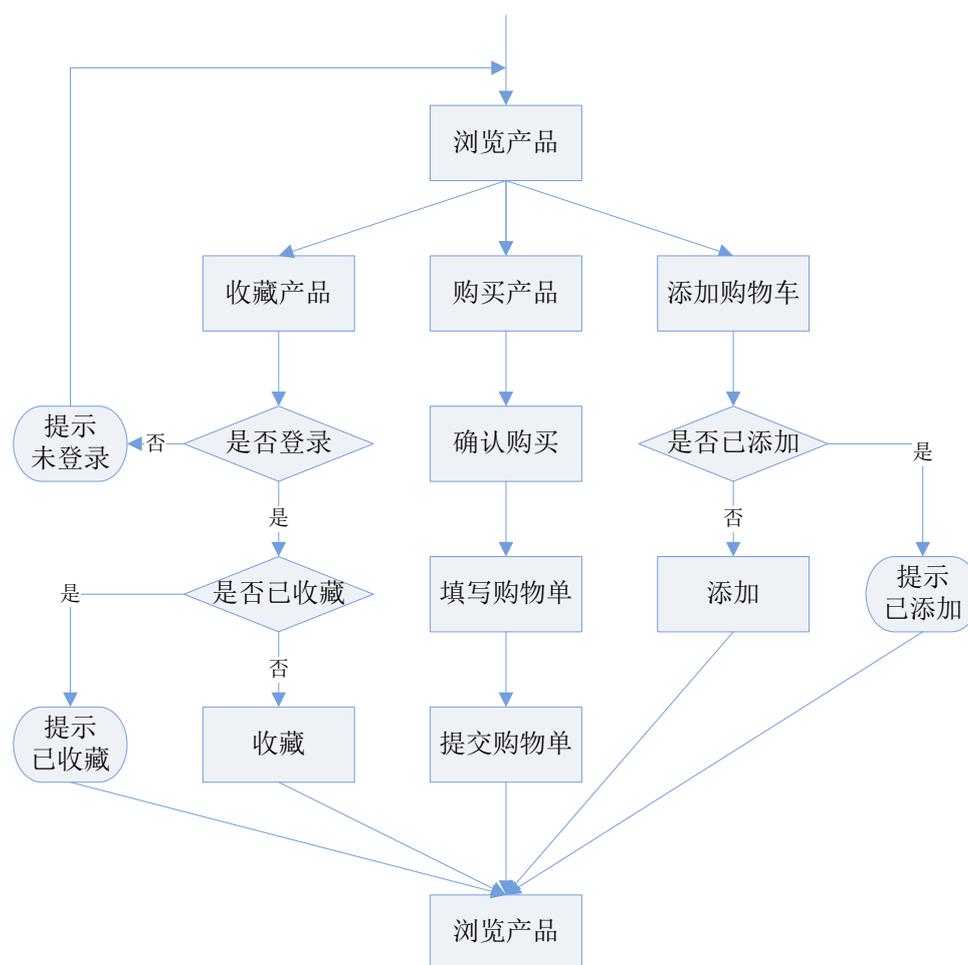


图4-5 购物功能模块流程图

从图4-5中可以看出，当客户在浏览产品时，主要可以执行3个动作：收藏

产品、购买产品和添加购物车。下面将分别对这3个动作进行介绍：

(1) 收藏产品

该功能主要是为登录用户提供收藏的功能，以此方便客户在以后购买。当客户浏览产品并点击收藏按钮时，系统首先会判断客户是否已登录，若用户未登录，则提示客户未登录无法收藏产品；否则，进一步判断该客户是否已经收藏过该产品，若用户已经收藏过该产品，则提示客户已经收藏过，无法继续收藏，若用户未收藏过该产品，则添加该产品至客户的收藏记录中。在此过程中，不管用户是否最终收藏成功，或是出现错误提示，最终客户都将继续浏览产品。

该过程的具体实现过程如下所示：

```
if(!isset($_SESSION['uid'])){
    $smarty->assign('nologin', true);
}else{
    if(!isset($_GET['id']) || $_GET['id']<1 || !is_numeric($_GET['id'])){
        $smarty->assign('err', true);
    }else{
        $r = $db->Execute('select id from goods where id = '.$_GET['id']);
        if(!$r->fields){
            $smarty->assign('nogoods', true);
        }else{
            $r = $db->Execute('select id from storebox where goods_id =
'.$_GET['id'].' and user_id='.$_SESSION['uid']);
            if($r->fields){
                $smarty->assign('hasadded', true);
            }else{
                $db->Execute("insert into storebox (goods_id, user_id)
values($_GET[id], $_SESSION[uid]) ");
            }
        }
    }
}
```

在以上代码中，我们首先通过`isset($_SESSION['uid'])`函数判断客户是否已登录，若未登录，则通过`$smarty->assign('nologin', true)`告诉模板该用户未登录；否则，我们会在数据库中查询该用户的收藏记录中是否已有该产品的记录，若有，则通过`$smarty->assign('hasadded', true)`告诉模板，否则，添加一条该客户收藏该产品的记录。

(2) 添加购物车

该功能主要是将客户打算购买的产品统一存放起来，方便客户一次性购买。当客户点击添加至购物车按钮时，系统首先判断该产品是否已经被添加过。若添加过，则提示用户已添加过该产品，否则添加该产品至客户的购物车，运行结果如图4-6所示：

| 商品列表 | | | | | |
|---|------|------|--------------------------------|------|--------------------|
| 商品名称 | 市场价 | 会员价 | 购买数量 | 小计 | 操作 |
|  jh11 | ¥12元 | ¥32元 | <input type="text" value="1"/> | ¥32元 | 删除 |

购物金额小计 ¥32元 [清空购物车](#) [更新购物车](#)

图4-6 添加产品到购物车

具体实现代码如下所示：

```
if(isset($_POST['id']) && isset($_POST['number'])) {
    $number =
is_numeric($_POST['number'])?$_POST['number']:intval($_POST['number']);
    $id = $_POST['id'];
    $number = $number < 1 ? 1:$number;
    if(!isset($_SESSION['cart'][$id])) {
        $_SESSION['cart'][$id]['id'] = $_POST['id'];
    }
    $_SESSION['cart'][$id]['number'] = $number;
}
```

由以上代码可以看出，在我们的网上鲜花订购系统中，购物车采用SESSION实现。SESSION是一种存储在服务器端的会话，在我们的系统中将购物车信息保存在名为\$_SESSION['cart']的数组中，我们首先使用isset(\$_SESSION['cart'][\$id])判断id等于\$id的产品是否已经在购物车中，若不存在，我们将该产品的id和购买数量加入购物车，否则，只更新购物车中该产品的购买数量。

(3) 购买产品

购买的流程相对简单，没有太多的判断。当客户点击购买按钮时，系统会首先让用户确认购买，在这一步中，主要是将购物车中所有的产品展示给客户，客户可以选择全部购买或者选择一件产品进行购买，在确定了购买方式之后，系统会要求客户填写购物单，填写完购物单之后，提交购物单，至此以此网上

鲜花订购过程完成，客户可以继续浏览产品。

点击购买的运行结果如图4-7和图4-8所示：

| 商品名称 | 市场价 | 会员价 | 购买数量 | 小计 | 购买 |
|---|------|------|--------------------------------|------|--------------------|
|  jh11 | ¥12元 | ¥32元 | <input type="text" value="1"/> | ¥12元 | 购买 |

图4-7 [点击购买产品](#)

订单填写

收件人

地址

电话 区号+号码

手机

送货时间

图4-8 [填写订单信息](#)

该过程实现的代码如下所示：

```

//展示所有购物车中的产品
<{foreach from=$goods item=s}>
<tr>
<td>
<a href="goods.php?id=<{$s.id}>">
<{$s.name}></a>
</td>

<td>¥<{$s.gprice}>元</td>

<td>¥<{$s.mprice}>元</td>

<td>
<input type="text" value="<{$s.number}>" name="goods_number[<{$s.id}>]"
id="goods_number_<{$s.id}>" />
</td>

<td>¥<{$s.money}>元</td>

<td>
<a href="javascript:if (confirm('您确实要把该商品移出购物车吗? '))
location.href='cart.php?action=delete&id=<{$s.id}>';" class="f6">删除</a>
</td>
</tr>
</foreach>
//订单提交过程
$error = array();
if(trim($_POST['user']) == "" || trim($_POST['address'])==" ||

```

```

$_POST['tel']=="" || $_POST['phone']=="" || $_POST['time']=="" ) {
    $error[] = '信息不完整';
}
if(!preg_match('/^\d{3,4}-\d{7,8}$/', $_POST['tel'])){
    $error[] = '电话格式错误';
}
if(!preg_match('/^\d{11}$/', $_POST['phone'])){
    $error[] = '手机格式错误';
}
if(!preg_match('/^\d{4}-\d{1,2}-\d{2}$/', $_POST['time'])){
    $error[] = '时间格式错误';
}
if(empty($error)){
    $now = time();
    $no = $now. rand(1000, 9999);
    $uid = isset($_SESSION['uid']) ? $_SESSION['uid'] :
$_SERVER['REMOTE_ADDR'];
    $rs = $db->Execute("insert into `order` (no, user_id, ids, user, ispayed,
status, create_at,tel, phone, address, time ) values ('$no', '$uid', '$_POST[id]',
'$_POST[user]', 1, 0, $now,'$_POST[tel]','$_POST[phone]',
'$_POST[address]', '$_POST[time]') ");
    if($rs){
        $arr = explode(',', $_POST['id']);
        $num = explode(',', $_POST['num']);
        $order_id = $db->Insert_ID();
        foreach ($arr as $k=> $v) {
            $db->Execute("insert into list (flower_id, number, order_id)
values ($v, $num[$k], $order_id");
            if(isset($_SESSION['cart'][$v])){
                unset($_SESSION['cart'][$v]);
            }
        }
        $smarty->assign('order', $no);
        $smarty->assign('success', 1);
    }else{
        $smarty->assign('err', 1);
    }
}
}
else{
    $smarty->assign('id', $_POST['id']);
    $smarty->assign('num', $_POST['num']);
    $smarty->assign('error', $error);
}
}

```

在上述代码中，在确认购买页面中，我们首先通过在模板中遍历\$goods数

组向客户展示购物车中所有的产品，在客户确认了购买之后，客户开始填写订单。由于订单信息的正确性对客户很重要，因此在客户填写完订单提交之后，程序要对订单的信息进行检验，在程序中，我们对数据的必填性、电话格式的正确性、手机格式的正确性和时间格式的正确性进行了验证。如果发现任意一条不符合要求，系统将给出错误提示，否则开始对用户的信息入库。我们除了对订单信息进行了入库，还对订单中购买的每一种产品都进行了入库记录。

4.2.2 产品添加功能模块

产品添加功能允许管理员在后台向系统中添加新的产品，该过程的流程图如图4-9所示：

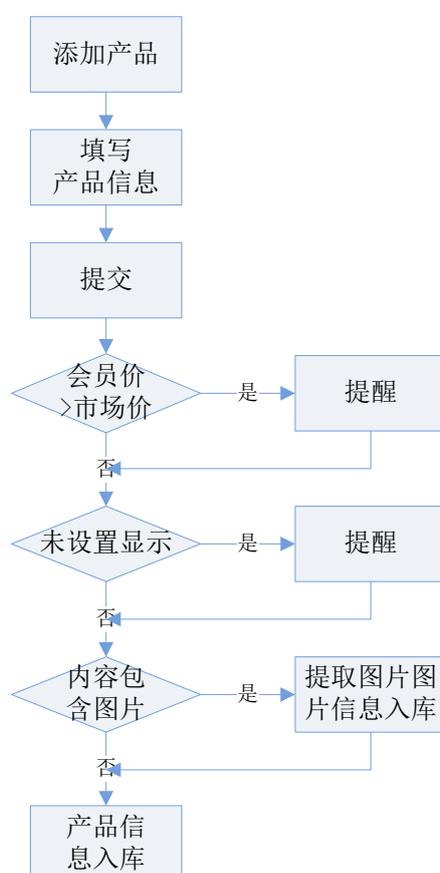


图4-9添加产品流程图

如图4-98所示，当管理员填写完产品的基本信息后，点击提交按钮，系统首先会对会员价格和市场价格进行比较，若会员价大于市场价，系统会发出提醒。接着，若发现未设置产品为显示状态，系统同样发出提醒。最后，系统会判断在产品描述中是否包含图片，若发现包含图片，则将图片提取出来，对图片信息单独入库，最后将产品信息入库。在产品添加页面中，我们使用了

fckEditor, fckEditor是一款在线编辑器, 它支持上传图片功能, 因此我们使用它为产品添加图片。

根据上面的对该过程的流程描述, 下面我们给出了这一过程中的实现代码, 如下所示:

```
$error = false;
if (trim($_POST['name']) == "") {
    $smarty->assign('err1', true);
    $error = true;
}
if (!$POST['category']) {
    $smarty->assign('err2', true);
    $error = true;
}
if ($_POST['gprice'] < $_POST['mprice']) {
    $smarty->assign('warning1', true);
}
if (!$POST['isshowed'][0]) {
    $smarty->assign('warning2', true);
}
if (!$error) {
    $color2 = trim($_POST['color2']);
    if (!empty($color2)) {
        $color = $color2;
    } else {
        $color = $_POST['color'];
    }
    $ishot = $_POST['ishot'][0] == 1 ? 1 : 0;
    $recommended = $_POST['recommended'][0] == 1 ? 1 : 0;
    $isshowed = $_POST['isshowed'][0] == 1 ? 1 : 0;
```

```

$smarty->assign('action', '添加');

preg_match("/^common\\.\\.images\\flowers\\(.*)" alt/",
$_POST['describe'], $images);

$des = preg_replace('/<img.*?>/', "", $_POST['describe']);
unset($images[0]);

$no = time();

$sql = "insert into `goods` (`name`,`no`,`category_id`,`use`,`units`,
`gprice`,`mprice`,`number`,`color`,`material`,`packaging`,`florid`,
`describe`,`ishot`,`recommended`,`isshowed`,`create_at`,vtimes,btimes) ";

$sql.= "values ('$_POST[name]','$no', '$_POST[category]',
$_POST[use]', '$_POST[unit]', '$_POST[gprice]', '$_POST[mprice]',
$_POST[number]', '$color', '$_POST[material]', '$_POST[packaging]',
$_POST[florid]', '$des', $ishot, $recommended, $isshowed, ".time()." ,0,0 )";

$rs = $db->Execute($sql);

if (!$rs) {
    $smarty->assign('error', true);
} else {
    if(isset($images)){
        $goods_id = $db->Insert_ID();
        foreach ($images as $k => $v) {
            $sql = "insert into `image` (`goods_id`,`path`) values
($goods_id, '$v)";
            $db->Execute($sql);
        }
    }
    $smarty->assign('success', true);
}
}

```

在上述代码中，我们首先判断会员价格是否大于市场价格，若是，则通过

\$smarty->assign('warning1', true)告知模板。接着，判断产品是否未显示，

若是，则通过\$smarty->assign('warning2', true)告知模板。然后，判断产品描述内容中是否含有图片信息，该过程使用正则表达式实现：

```
preg_match("/^\/common\/..\images\/flowers\/(.*)" alt",
$_POST['describe'], $images);
```

该方法通过匹配产品描述中是否含有图片地址判断是否含有图片信息，若发现，则对图片进行入库。

此外，因为产品类别有父类别和子类别，在添加产品时产品的子类别通过ajax获得，实现过程如下所示：

```
//基于jquery的ajax
<script type="text/javascript">
    function getCategory(){
        var id = document.getElementById('fcategory').value;
$.ajax({
        url: 'ajax/getScategory.php?id='+id,
        type: 'GET',
        dataType: 'text',
        timeout: 1000,
        error: function(){
            alert('暂时无法获取子类别，请重试');
        },
        success: function(date){
            var a = document.getElementById('scategory');
            a.innerHTML = date;
        }
    });
}
</script>
```

```

// getCategory.php
$Id = checkInput($_GET['id']);
$sql = "select id, name from category where fid = $Id";
$rs = $db->Execute($sql);
$category = array();
while (!$rs->EOF) {
    echo '<option value="'. $rs->fields['id'].'">'. $rs->fields['name']. '</option>';
$rs->MoveNext();
}

```

当选择类型的父类时，触发getCategory函数，该函数使用异步的方式获取子类别。Ajax请求getCategory.php页面，getCategory.php返回数据给ajax，当出错时，弹出错误提示框，否则，返回子类的下拉菜单。

4.2.3 订单管理功能模块

当有客户在前台下了订单之后，管理员在后台会收到通知，该过程的具体实现如下所示：

```

//获取新订单数
$sql = 'select count(*) as newOrder from `order` where status = 0 or status is null';
$q = $db -> Execute($sql);
$smarty->assign('newOrder', $q->fields['newOrder']);
//模板中显示
<{if $newOrder gt 0}>
<blink><{$newOrder}>新订单 </blink>
<{else}>
最新订单
</if>

```

在上述代码中，当在后台打开任意页面，系统都会去数据库中查询最新订单的条数，当数目大于0时，将在页面中显示新订单的具体条数，并且使用HTML语言的blink标签作为提示。

管理员发现有新订单后会对订单进行查看，然后执行发货等操作，这些过程的流程图如图4-10所示：



图4-10订单处理流程

如图4-10所示，当有新订单到达时，管理员会查看订单详情，订单详情信息包括订单中购买的产品名、产品数量、购买人、收件人、收件人地址、电话、送货时间等信息。管理员开始发货，发完货之后修改订单的状态至已发货状态。当收件人收到货物后，管理员再将订单状态修改至已接收状态。至此，一次订单处理流程结束。

上述订单的运行结果图如下图所示：

| Order / 订单管理 | 订单号 | 订单者 | 下单时间 | 送货时间 | 操作 |
|--------------|-----------|-----|------------------|------------|---------|
| 最新订单 | 130536355 | 322 | 2011-05-14 16:59 | 0000-00-00 | 已发货 已接收 |
| 历史订单 | 130536168 | 432 | 2011-05-14 16:28 | 0000-00-00 | 已发货 已接收 |
| 订单追踪 | 130536165 | 432 | 2011-05-14 16:27 | 0000-00-00 | 已发货 已接收 |

图 4-11 查看最新订单

| 订单号 | 订单者 | 下单时间 | 送货时间 | 状态 | 操作 |
|-----------|-----|------------------|------------|-----|----|
| 130611661 | 233 | 2011-05-23 10:10 | 2011-05-01 | 已送达 | |

图 4-12 发货

| | | | | | | |
|--------------------------|-----------|-----|------------------|------------|-----|-----|
| <input type="checkbox"/> | 130536373 | 322 | 2011-05-14 17:02 | 0000-00-00 | 已发货 | 已接收 |
| <input type="checkbox"/> | 130536361 | 322 | 2011-05-14 17:00 | 0000-00-00 | 已发货 | 已接收 |

图 4-13 买家收货

上述订单处理过程的关键代码，如下所示：

```
//查询订单详情
$rs = $db->Execute('select * from `order` where id = '.$_GET['id']);
if(!$rs){
    header('location: orderlist.php');
}elseif(empty ($rs->fields)){
    $smarty->assign('notexist', 1);
}else{
    $smarty->assign('order', $rs->fields);
    $order_id = $rs->fields['id'];
    $r = $db->Execute('select l.number,g.name,g.id from list as l left join goods
as g on l.flower_id = g.id where l.order_id = '.$order_id);
    $goods = array();
    while (!$r->EOF){
        $goods[] = $r->fields;
        $r->MoveNext();
    }
    $smarty->assign('goods', $goods);
}
//修改订单状态
if(isset($_GET['id']) && isset ($_GET['status'])){
    $db->Execute('update `order` set `status` = '.$_GET['status'].' where id =
'.$_GET['id']);
    header('location: neworder.php');
}
```

上述代码中，在查询订单详情时，首先会判断该订单号是否存在。若不存在，则给出错误提示；否则，根据订单号查询该订单所购买到的所有产品，在这个查询过程中，我们使用左连接的查询方式，这样的查询方式可以的效率较高，可以降低数据库服务器的压力。在修改订单状态时，直接根据订单GET参数修改指定的订单。

4.3 本章小结

在该章中，我们首先介绍了数据库的设计过程，之后，我们挑选出有特色的3个模块，对这3个模块的设计与实现做了详细介绍。通过该章的介绍使读者

对该系统的具体实现过程有所了解。

5. 总结和展望

5.1 论文总结

在互联网高速发展的今天，Internet 已成为人们获取、发布和传递信息的重要渠道，它在人们的日常工作、学习和生活中正发挥着越来越重要的作用。而网站作为互联网上信息交流的主要手段之一，其在商业领域应用的重要性更是不可言喻，它已成为商业组织、政府和企事业单位信息化建设中的重要组成部分，从而倍受人们的重视。另一方面，当前人们的生活节奏越来越快，越来越多的人希望可以在互联网通过点击鼠标完成复杂的事务，免除一些繁琐的过程。而本课题的设计正是网站在便捷式商业交流领域应用的一个较好体现。

网上鲜花订购系统，是一个将传统的商务交流场景转移到 Web 浏览器上的系统。该系统解决了传统的商务交易中买卖双方必须面对面对话的约束，交易双方只要可以连接网络，便可以在任何时间、任何地点下订单或接受订单，通过网站使交易方式更方便、更快捷。

PHP 作为一种内嵌式的服务器端脚本语言，其在 Web 领域的应用正越来越广泛，PHP 发展迅速、应用广泛，如今 PHP 已成为最受欢迎的 Web 开发语言之一。作为一门成熟的开源语言，PHP 有许多开源的优秀模板和框架，通过使用这些模板和框架，不但节省开发时间、减少重复编码的开发，更有助于建立更稳定的应用，同时，还可实现程序逻辑与美工的分离，使程序更利于维护。本系统采用 PHP 界有名的 Smarty 模板实现，在阐述网站在商业领域的重要性的同时，也向读者阐述模板分离为程序带来的好处。

5.2 今后工作展望

近年来，随着 Internet 的迅速崛起，互联网已日益成为收集提供信息的最佳渠道并进入传统的流通领域。于是电子商务开始流行起来，一种全新的购物理念开始形成并逐步发展。

网上订购系统作为 B2B, B2C (Business to Customer,即企业对消费者),

C2C (Customer to Customer,即消费者对消费者) 电子商务的前端商务平台, 在其商务活动全过程中起关举足轻重的作用。

网上购物是一种具有交互功能的商业信息系统。它向用户提供静态和动态两类信息资源。所谓静态信息是指那些比经常变动或更新的资源, 如公司简介、管理规范和公司制度等等; 动态信息是指随时变化商务是通过电子化的手段来完成商业贸易活动的一种方式。

作为网站建设者, 我们必须根据客户需求, 做出更加完善和美化的购物系统, 同时也要结合消费者的爱好和需求, 使购物功能更加完善和便捷。

参考文献

- [1] W.Jason Gilmore. PHP 与 MySQL5 程序设计. 人民邮电出版社,2007 年 4 月.
- [2] 前沿科技 曾顺. 精通 CSS+DIV 网页样式与布局. 人民邮电出版社,2009 年 11 月.
- [3] David Flanagan. JavaScript 权威指南. 清华大学出版社, 2003 年 4 月.
- [4] 邹天思, 孙鹏. PHP 从入门到精通. 北京:清华大学出版社,2008.
- [5] 明日科技. PHP 函数参考大全. 北京:人民邮电出版社,2007.
- [6] NARAMORE[美].PHP5、APACHE、MYSQL 网络开发 [M].北京: 电子工业出版社, 2005.
- [7] W.Jason Gilmore,W.J. PHP 与 MySQL 程序设计. 人民邮电出版社,2009 年 6 月.
- [8] 康贤昆. PHP 开发框架的研究与实现. 2007.
- [9] 潘凯华, 邹天思. PHP 开发实战宝典. 北京:清华大学出版社,2010.
赵鹤芹. 设计动态网站的最佳方案:Apache+PHP+MySQL. 计算机工程与设计 2007 28(4).
- [10] 兰旭辉 熊家军 邓刚. 基于 MySQL 的应用程序设计. 计算机工程与设计 2004 25(3).
- [11] 陈湘扬, 陈国益. PHP5+MySQL 网页系统开发设计. 北京:电子工业出版社, 2007.
- [12] Michael kofler. MySQL 5 权威指南. 北京:人民邮电出版社,2006.
- [13] 欧阳峰, 陈朝荣.电子商务技术.中国财政经济出版社.2001.8.1
- [14] 严洪范. OA92-综合办公系统[J].计算机世界,电子商务出版社。1993.7.
- [15] 邓问渊, 陈惠贞,陈俊荣. ASP 与网页数据库设计.中国铁道出版社, 2001.2.
- [16] 黄河编著.网页设计实务全攻略.北京大学出版社.2000.8.