

# 第 30 章 程序设计说明书

www.brophp.com

## 内容管理系统(BroCMS)

文件状态： [ ] 草稿 [ ✓ ] 正式发布 [ ] 正在修改	文件标识：	LAMP 兄弟连-BroCMS-02-Program
	当前版本：	2.0
	作 者：	兄弟连
	完成日期：	2011-12-15

## 版本历史

版本/状态	作者	参与者	起止日期	备注
1.0	兄弟连	教学组成员	2009-11-05 2009-11-20	《细说 PHP》第一版
2.0	兄弟连	教学组成员	2012-04-15 2012-04-30	《细说 PHP》第二版

## 30.1 引言

根据分析，我们开发的软件是一个 CMS 系统，也就是我们常说的新闻发布系统，CMS 系统就是能够自动地发布各种信息的平台，所以我们的设计思想就是围绕着这个目标展开。本系统是基于 PHP 开发的，为了便于代码重用，我们使用 BroPHP 框架进行开发,并且统一管理。

### 30.1.1 编写目的

本说明是 CMS 软件产品的程序设计说明，记录了系统整体实现上技术层面上的设计。程序设计说明书是进行系统编码的依据，编写本文档的目的在于为程序员的编码提供详细的说明，使程序员能根据详细设计的框图进行正确的编码。并且以需求说明作为依据，同时该文档将作为产品实现、特性要求和控制的依据。本文档的读者对象为程序员，系统设计人员。软件开发小组的每一位参与开发成员应该阅读本说明，以清楚产品在技术方面的要求和实现策略。

### 30.1.2 背景

本系统是《细说 PHP》最后一部分的项目实例，可以作为读者和 LAMP 兄弟连学员的项目参考资料。除了学习项目开发参考使用之外，也可以通过本系统建设自己的网站。

- 软件系统的名称：BroCMS 网站内容管理系统 for NT(v2.0)
- 该软件系统开发项目的任务提出者：《细说 PHP》作者
- 该软件系统的用户：公司客户

### 30.1.3 定义

为了便于表达及避免歧义，现将本说明书中使用的专门术语的定义和外文首字母组词的原词组列表如下：

CMS：内容管理系统。Content Management System

API：插件管理（第三方开放的插件管理）

BroPHP：本系统应用的超轻量级 PHP 开发框架

## 30.1.4 使用技术

开发技术: PHP, BroPHP 框架, JavaScript

数据库技术: MySQL

程序控制软件: VIM

## 30.1.5 参考资料

- 《细说 PHP》
- 《BroPHP 框架》手册
- 软件需求说明书（本书第 28 章）
- 编码规范(本书附录 A)
- 项目进度文档
- API 接口文档
- 数据库设计说明书

## 30.2 系统的结构

系统总体结构设计采用了 MVC 的设计模式，和完全使用面向对象的思想开发，并根据 BroPHP 框架的规则去规划视图、控制器和实体类及调用关系。提高代码的易维护性、易读性，增加类内部的纯度、类之间调用的灵活性。

### 30.2.1 项目的目录结构

在需求阶段，我们将 CMS 分为了前台和后台两个应用。因为是通过 BroPHP 框架进行开发，所以只需要为每个应用单独声明一个入口文件，分别命名为 `index.php`(前台入口)和 `admin.php`(后台应用入口)，并存放到项目目录 `brocms`（该目录存放在 Web 服务器的根目录下）下面。在前台应用的入口文件 `index.php` 中指定前台的应用目录为“home”目录，在后台应用的入口文件 `admin.php` 中指定后台的应用目录为“admin”目录。分别访问两个入口文件 BroPHP 自动生成项目结构目录，如下所示：

<b>-- brocms 目录</b>	#项目根目录
<b>-- brophp 目录</b>	#BroPHP 框架目录
<b>-- index.php 文件</b>	#前台主入口文件（可以使用其他名称）
<b>-- home 目录</b>	#自定义的前台项目应用目录
<b>-- controls 目录</b>	#声明控制器类的目录（前台控制器目录）
<b>-- models 目录</b>	#声明业务模型类的目录（前台模型目录）
<b>-- views 目录</b>	#声明视图的目录（前台视图目录）
<b>-- admin.php 文件</b>	#后台主入口文件（可以使用其他名称）
<b>-- admin 目录</b>	#自定义的后台项目应用目录
<b>-- controls 目录</b>	#声明控制器类的目录（后台控制器目录）
<b>-- models 目录</b>	#声明业务模型类的目录（后台模型目录）
<b>-- views 目录</b>	#声明视图的目录（后台视图目录）
<b>-- config.inc.php 文件</b>	#项目的配置文件
<b>-- classes 目录</b>	#用户自定义的扩展类目录
<b>-- commons 目录</b>	#用户自定义的扩展函数目录
<b>-- public 目录</b>	#项目的所有应用公用的资源目录
<b>-- runtime 目录</b>	#项目运行时自动生成文件存放目录（可以随时删除）

按 BroPHP 框架的要求，程序的配置文件使用项目目录下的 config.inc.php 文件，自己开发的实体类存放在 classes 目录中，公用的图片、js 和 css 等资源存放在 public 目录中，用户上传的图片存放在 public/uploads/ 目录中。

后台应用的目录为项目目录下的 admin 目录，控制器、视图和模型分别写在 controls 目录、views 目录和 models 目录中。前台应用也是独立的，存放在项目目录下的 home 目录下，也有对应的控制器、视图和模型目录。本系统为前台应用开发了两套模板，所以在前台的 views 目录下有两个目录，用户可以进行模板风格切换。

### 30.2.2 模块结构

按需求分析的结果，将后台应用分为 12 个模块，前台应用分为 8 个模块。根据主要功能确定前后台应用中每个模块的操作和操作权限，如下表 30-1 和 30-2 所示：

表 30-1 后台应用的模块操作说明

模块	操作	权限
登录管理	获取登录界面、处理登录、退出、获取验证码操作	无
操作界面管理	主页、顶部、菜单、主区域、底部	后台登录用户
常规管理	获取系统信息、信息设置界面、设置网站信息、更新缓存	网站编辑权限
公告管理	查询公告列表、获取添加界面、添加、获取修改界面、修改、排序、删除，上传图片	网站编辑权限
友情链接管理	查询友情链接列表、获取添加界面、添加、获取修改界面、修改、排序、删除	网站编辑权限
相册管理	查询相册列表、获取添加界面、添加、获取修改界面、修改、删除	网站编辑权限
图片管理	查询图片列表、获取上传界面、添加、删除、弹出列表	网站编辑权限

栏目管理	查询栏目列表、获取添加界面、添加、获取修改界面、修改、排序、删除、设置栏目显示	网站编辑权限
文章管理	查询文章列表、获取添加界面、添加、获取修改界面、修改、删除、设置显示状态、设置是否允许评论	管理文章权限
幻灯片管理	查询幻灯片列表、获取添加界面、添加、获取修改界面、修改、删除	管理文章权限
用户组管理	查询用户组列表、获取添加界面、添加、获取修改界面、修改、删除	管理用户权限
用户管理	查询用户列表、获取添加界面、添加、获取修改界面、修改、删除	管理用户权限

表 30-2 前台应用的模块操作说明

模块	操作	权限
首页管理	获取首页全部内容、获取公告内容	无
列表管理	获取某一个栏目的全部信息	无
内容管理	获取某一文章内容，获取文章评论信息	发表评论权限
搜索管理	搜索和显示	无
登录注册	获取注册界面、添加、登录处理、退出、唯一性检查	无
个人空间管理	显示个人空间首页平台、顶部处理、菜单、用户信息修改、密码设置、头像设置、设置关注、取消关注、获取关注列表、获取粉丝列表	登录用户
消息管理	获取消息列表、写消息、查看单个消息、删除消息	登录用户并有发消息权限
动态管理	获取用户动态列表、删除动态、获取收藏或推荐、设置收藏数、设置推荐、添加评论、修改评论、删除评论、上传图片、上传 flash 等	登录用户

### 30.2.3 程序结构

根据反复讨论的需求说明和上面的模块规划，并结合 MVC 设计模式的思想，为每个模块声明一个控制器类来操作。先确定每个控制器和其中每个操作的名称，这样就可以大概确定了程序的结构。前台和后台每个应用模块的控制器类及其说明如下表 30-3 和 30-4 所示：

表 30-3 后台应用每个控制器类的结构说明

模块	控制器类	操作方法	简要说明
登录管理	Login	index()	获取登录界面的操作
		prologin()	处理登录的操作
		logout()	用户退出的操作
		code()	获取验证码信息
操作界面管理	Index	index()	获取后台主页分框结构
		top()	获取后台主页顶部分框页面
		menu()	获取后台主页左部分框菜单页面

		main()	获取后台主页右部主体分桢页面
		bottom()	获取后台主页底部分桢页面
常规管理	Base	sysinfo()	获取网站服务器的系统信息
		baseset()	获取网站设置的表单界面
		set()	接收数据并设置网站信息
		upcache()	更新网站缓存
公告管理	Notice	index()	查询公告列表
		add()	获取添加公告表单界面
		insert()	接收公告数据并插入到数据库中
		mod()	获取修改公告表单界面
		update()	接收公告信息修改数据库原有的记录
		order()	对公告显示顺序进行排列
		del()	删除指定的公告信息
友情链接管理	Flink	index()	查询友情链接列表
		add()	获取添加友情链接表单界面
		insert()	接收友情链接数据并插入到数据库中
		mod()	获取修改友情链接表单界面
		update()	接收友情链接信息修改数据库原有的记录
		order()	对友情链接显示顺序进行排列
		del()	删除指定的友情链接信息
相册管理	Al bum	index()	查询相册列表
		add()	获取添加相册表单界面
		insert()	接收相册数据并插入到数据库中
		mod()	获取修改相册表单界面
		update()	接收相册信息修改数据库原有的记录
		del()	删除指定的相册信息
图片管理	Image	index()	查询图片列表
		add()	获取上传图片表单界面
		insert()	接收图片到服务器中并插入到数据库
		open()	显示弹出式图片列表
		del()	从服务器文件和数据库中删除指定的信息
栏目管理	Column	index()	查询栏目列表
		add()	获取添加栏目表单界面
		insert()	接收栏目数据并插入到数据库中
		mod()	获取修改栏目表单界面
		update()	接收栏目信息修改数据库原有的记录
		del()	删除指定的栏目信息
		dis()	设置栏目的显示状态
		order()	对栏目显示顺序进行排列
文章管理	Article	index()	查询文章列表
		add()	获取添加文章表单界面
		insert()	接收文章数据并插入到数据库中

		mod()	获取修改文章表单界面
		update()	接收文章信息修改数据库原有的记录
		del()	删除指定的文章信息
		fpro()	批量处理文章的各种状态
		status()	单个文章状态的设置
幻灯片管理	Play	index()	查询幻灯片列表
		add()	获取添加幻灯片表单界面
		insert()	接收幻灯片数据并插入到数据库中
		mod()	获取修改幻灯片表单界面
		update()	接收幻灯片信息修改数据库原有的记录
		order()	对幻灯片显示顺序进行排列
		del()	删除指定的幻灯片信息
用户组管理	Group	index()	查询用户组列表
		add()	获取添加用户组表单界面
		insert()	接收用户组数据并插入到数据库中
		mod()	获取修改用户组表单界面
		update()	接收用户组信息修改数据库原有的记录
		del()	删除指定的用户组信息
用户管理	User	index()	查询用户列表
		add()	获取添加用户表单界面
		insert()	接收用户数据并插入到数据库中
		mod()	获取修改用户表单界面
		update()	接收用户信息修改数据库原有的记录
		del()	删除指定的用户信息
全局管理	Common	init()	公用操作，用于处理登录和控制权限操作
		mess()	处理用户提示消息
		upimage()	处理文章和公告的图片上传
		upflash()	处理文章和公告的 Flash 上传

30-4 前台应用每个控制器类的结构说明

模块	控制器类	操作方法	简要说明
首页管理	Index	index()	输出网站首页信息
		notice()	输出网站公告页面信息
列表管理	List	index()	以分页格式输出一个栏目记录信息
文章内容管理	Article	index()	输出用户请求的文章详细信息
		comment()	以分页显示当前文章的评论信息
搜索管理	Search	index()	处理用户搜索并显示搜索结果
登录注册	Login	index()	提供用户登录表单
		logout()	处理用户退出
		register()	提供用户注册表单
		insert()	将用户注册信息插入到数据库中
		unique()	检查用户名称是否惟一

		code()	提供用户注册的验证码
		vcode()	验证用户输入的验证码
个人空间管理	<b>User</b>	index()	提供个人空间的操作界面
		top()	个人空间顶部信息的操作
		menu()	个人空间左部菜单的操作
		set()	设置用户信息
		pset()	用户密码设置
		tset()	用户头像设置
		follow()	设置用户关注
		delfollow()	取消用户关注
		allfollowed()	获取粉丝列表
		allfollowing()	获取关注列表
消息管理	<b>Message</b>	index()	显示用户的全部消息列表
		write()	发用户消息
		view()	查看某个消息内容
		del()	删除指定的消息
动态管理	<b>Dynamic</b>	index()	显示一个用户的所有动态信息
		Del()	删除指定的用户动态
		Gts()	获取收藏或推荐数
		Collection()	设置收藏数
		Recommnd()	设置推荐
		Addc()	添加用户评论
		Modc()	修改用户评论
		Delc()	删除用户评论
		Add()	获取文章添加表单
		Insert()	向数据表中添加新的文章
		Mod()	获取文章修改表单
		Update()	更新已有一篇文章记录
		Upimage()	处理添加文章时的图片上传
		Upflush()	处理添加文章时的 Flush 动画上传
全局管理	<b>Common</b>	init()	公用操作，用于处理登录和控制权限操作

按 BroPHP 框架的规则，将前台的控制器类声明在项目目录下的 `home/controls` 中，后台控制器类声明在 `admin/controls` 目录中。并且控制器类所在的文件名要以“控制器类名.class.php”格式命名，文件名称一律小写。前台和后台两个应用控制器类之间的继承关系如图 30-1 和 30-2 所示：



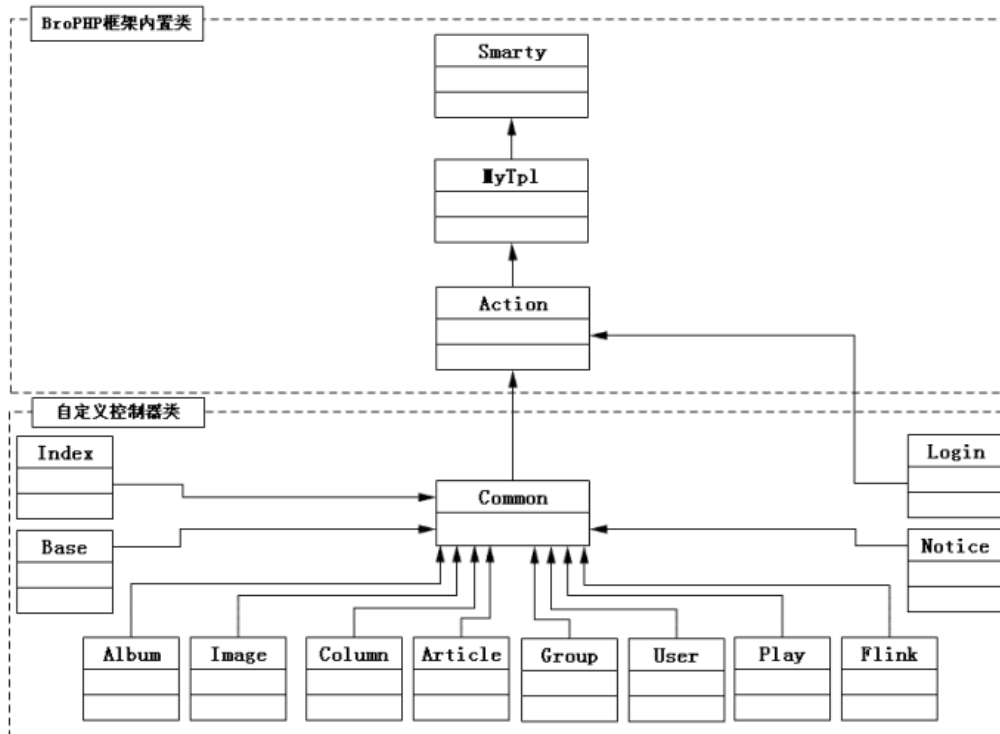


图 30-1 BroCMS 后台控制器类图继承关系

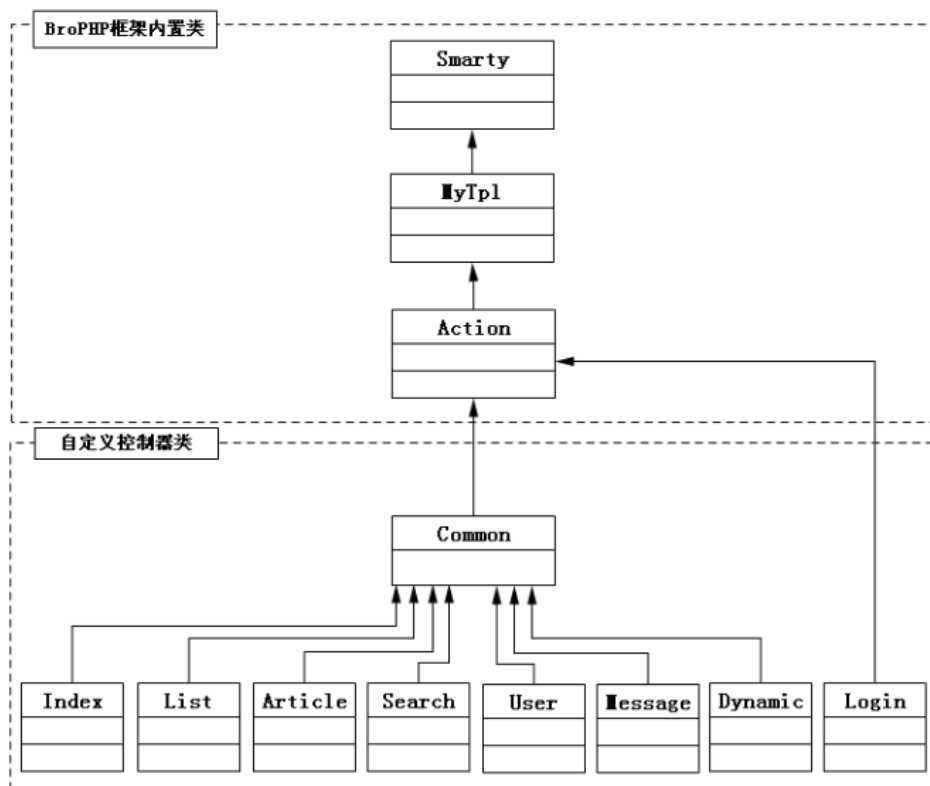


图 30-1 BroCMS 前台控制器类图继承关系

## 30.3 用户管理模块设计说明

从本节开始，将逐个地给出各个模块的设计考虑，但由于篇幅限制本书只给出一个用户模块的设计参考。

### 30.3.1 功能

网站在发展过程中需要用户互动来促进网站发展，另一方面也需要防止用户对网站信息的越权访问或随意发布各类信息。因此需要针对不同的服务对象（非注册用户、注册用户、不同的注册用户类型），根据其需求开设不同的信息栏目、提供不同范围的信息服务。根据网站的管理需要设定不同权限的管理员，以方便协同管理网站。当用户在网站中注册为注册会员，则相当于在网站中有了一个通行证，会员可用于辨别属于自己的信息、访问或发布权限允许内的信息。用户模块的具体操作如下所示：

#### 1. 添加用户

除了新用户自己注册以外，还可以通过用户管理系统提供的表单界面添加新的用户。

#### 2. 查询用户

可以根据多种条件筛选出需要处理的用户列表，并通过分页管理多条数据。

#### 3. 编辑用户

和添加用户类似，对已经注册的用户进行编辑修改。

#### 4. 删除用户

可以删除一些非法的用户记录，可以单条删除，也可以通过复选框选择多项一起删除。

### 30.3.2 流程逻辑

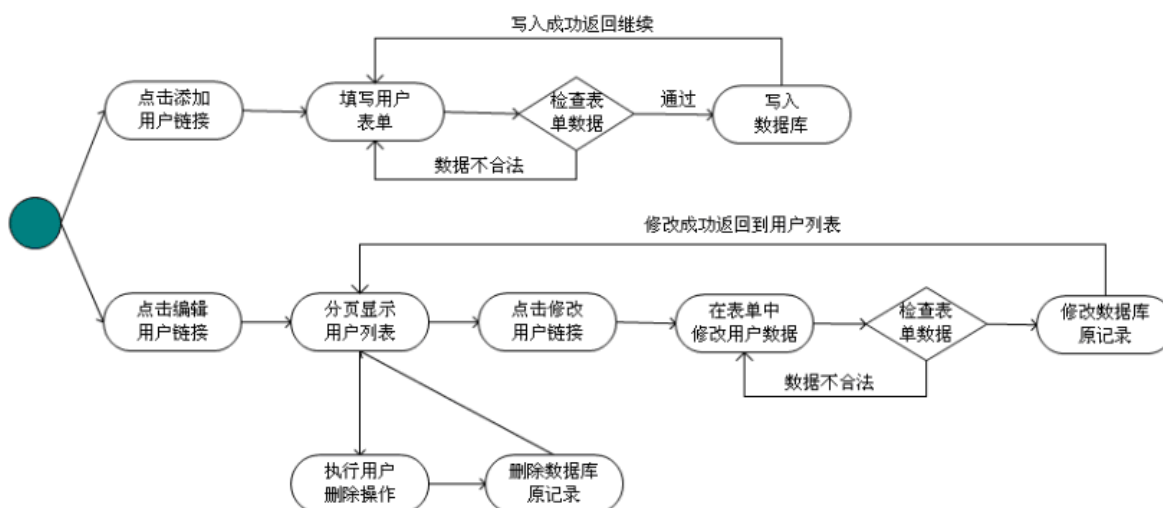


图 30-3 用户模块操作流程图

### 30.3.3 接口

根据 BroPHP 框架的规则，在设计用户管理模块时需要在指定的位置一个控制器类(User)。并在控制器类中声明 6 个操作方法，包括用户列表操作 index()、获取添加界面的操作 add()、数据入库的操作 insert()、获取修改页面的操作 mod()、修改数据库记录的操作 update()和删除操作 del()。其中有 3 个操作方法需要通过模板显示数据，并在一些操作中需要通过模型类及实体类完成一些业务。用户管理模块的 MVC 模式结构如图 30-4 所示：

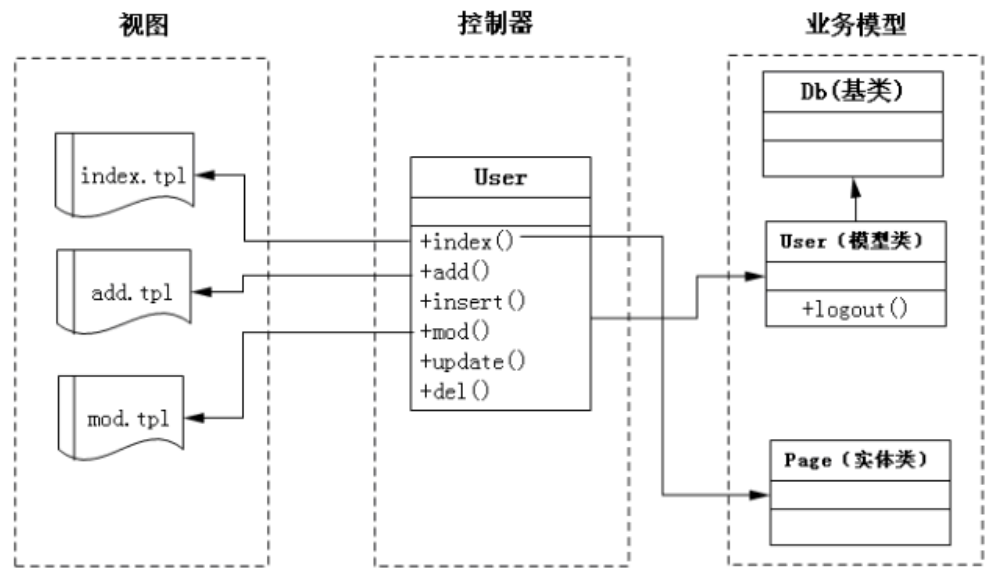


图 30-4 用户模块的 MVC 模式设计结构

### 30.3.4 存储分配

本模块只需要对一个用户表操作即可，所有数据操作都通过 Model 类来完成。数据表结构详见《数据库设计说明书》（第 29 章）。

### 30.3.5 注释设计

- 在控制器类、模型类和实体类的首部添加注释说明；
- 在类中的每个方法上方添加注释说明；
- 对各变量的功能、范围、缺省条件等加上注释；
- 对使用的逻辑算法加上注释；

### 30.3.6 限制条件

在添加和修改数据时用户名称、用户密码和电子邮箱必须填写。管理员用户不能删除，否则将不能登录系统，用户被删除时，他的文章及评论等所有信息也会一同删除。

## 30.3.7 测试计划

通过测试达到以下目标：

- 测试已实现的模块是否达到设计的要求，各个操作点是否以实现。
- 产品规定的操作和运行是否稳定。
- Bug 数和缺陷率控制在可接收的范围之内。
- 业务流程，及数据流从软件中的一个模块流到另一个模块的过程中的正确性。

## 30.3.8 尚未解决的问题

无（在本程序的设计中尚未解决而设计者认为在软件完成之前应解决的问题）。

## 30.3.9 获取添加用户的界面操作 add()

### 30.3.9.1 程序描述

如果需要向数据库中添加一条用户记录，就需要先为管理员提供一个添加表单。操作用户先通过管理平台的菜单项单击“添加用户”链接，就会通过链接中的 URL 访问到用户模块的 add() 操作中。在该操作中，需要从模型中获取用户组的列表发送给对应的模板中显示，并输出模板文件（add.tpl），提供给操作用户一个添加用户界面。

### 30.3.9.2 输入项

本操作的访问 URL 为：/brocms/admin.php/user/add

### 30.3.9.3 输出项

用户请求时并不需要提供其他的参数输入，但需要从用户组模型中获取一个用户组列表。需要请求 Group 模型中的“formselect(“gid”, 2)”，第一个参数为用户组表单列表的名称，第二个参数是默认从第二个开始显示。将从 Group 模型中获取的用户组列表通过变量名“select”发送到模板中，输出添加用户的模板。

### 30.3.9.4 算法

**第一步：**提示用户添加的规则

**第二步：**从用户组模型中获取用户组下拉列表分配到模板中

**第三步：**加载并输出添加模板 add.tpl

### 30.3.9.5 模板设计

需要为该操作声明一个添加用户的模板文件，在模板中只需要声明一个 HTML 表单。在模板设计时需要用到的一些信息如下所示：

**模板文件：**声明在视图目录中，并在用户模块目录（user）下，文件名称为 add.tpl

**子模板：**包含公用的头部(header.tpl)和尾部/footer.tpl)模板文件

**表单提交位置：**当前模块的 insert()操作中

**提交方法：**POST

**表单项的设计：**如表 30-5 所示

表 30-5 添加用户表单内容项

表单项	类型	名称	说明
用户组	checkbox	gid	提供用户组下拉列表
用户名称	text	username	可以使用中文，但禁止除[@][.]以外的特殊符号
用户密码	password	userpwd	初使密码为 brophp
确认密码	password	repwd	
电子邮箱	text	email	请正确添写你的电子邮件地址
用户性别	radio	sex	男、女、保密
是否禁用	checkbox	disable	禁用该用户

**说明：**如果需要连续录入多个用户，可以添加一下用户组的“记住选项”。

### 30.3.10 用户数据入库的操作 insert()

#### 30.3.10.1 程序描述

操作者在添加用户表单中录入一个用户信息后，将用户所有录入的数据以 POST 方法传递到 insert()操作中，处理数据并插入到数据表 user 中。

#### 30.3.10.2 输入项

本操作的访问 URL 为：/brocms/admin.php/user/insert

本操作的输入项是用户在表单中录入的全部信息，包括用户组编号、用户名称、用户密码确认密码、电子邮箱、用户性别和禁用状态。以 HTTP 的 POST 方法传递到本操作中，所以全部保存在超全局数据\$\_POST 中，其中需要对用户密码进行加密处理（使用 md5()函数）。数组\$\_POST 的格式如下所示：

```
$_POST=array(
    'username'=>'user',           //用户名
    'userpwd'=>'123456',          //用户密码
    'repwd'=>'123456',            //确认密码
    'sex'=>'1',                    //用户性别
    'disable'=>'0',                //禁用状态
```

```
'email'=>'user@brophp.com',    //用户电子邮箱
'sub'=>'添加用户'              //提交按钮
);
```

### 30.3.10.3 输出项

本操作中的输出项是要插入到数据库中的数据，是通过验证并处理过的数据。

### 30.3.10.4 算法

**第一步：**处理用户提交的数据；

**第二步：**使用 BroPHP 框架中的自动验证方式进行数据验证；

**第三步：**如果验证通过则将数据插入到数据库 `user` 表中，插入成功后再返回到添加表单界面，让操作者可以继续添加下一个用户；

**第四步：**如果验证数据失败也是返回到添加表单去重新添加数据，但要保留上一次录入过的数据。

**第五步：**数据添加成功或是失败都要提示操作者。

## 30.3.11 查询用户列表操作 `index()`

### 30.3.11.1 程序描述

操作者通过“编辑用户”链接进入到用户的列表中，用户的记录列表全部从数据库中获取并以分页形式显示。操作者可以通过用户组的下拉列表选择对应某个用户组下面的全部用户记录，也可以通过用户名称的模糊搜索快速定位到需要的查找的用户，并且通过分页切换页面时也要保持查询条件。

### 30.3.11.2 输入项

本操作的访问 URL 为：`/brocms/admin.php/user/index`

1. 用户组编号 （通过 GET 方法传递），格式为 `$_GET['gid']`；
2. 搜索的用户名 （通过 GET 方法传递），格式为 `$_GET['search']`；

### 30.3.11.3 输出项

1. 用户列表，以一个变量形式分配到模板中，类型为一个二维数组，变量名为 `$users`；
2. 分页内容，也是以一个变量形式分配到模块中，类型为一个字符串，变量名为 `$fpage`；

### 30.3.11.4 算法

**第一步：**提示用户操作方式；

**第二步：**当操作者从下拉列表中选择一个用户组后，用户的记录列表则为当前用户组下面的所有用户，所以需要按组编号组合一个查询的 **WHERE** 条件，并要根据查询条件组合出分页需要的传递参数；

**第三步：**同样在搜索用户时也需要按用户称组合一个查询的 **WHERE** 条件和分页参数，并且要保持在特定用户组下的进行搜索，只查询当前用户组下的符合条件的用户记录；

**第四步：**如果没有指定用户组或搜索条件为空，则查询全部记录；

**第五步：**加载输出 index.tpl 模板显示用户记录；

### 30.3.11.5 模板设计

在用户列表模块中，需要遍历用户数组列表，及其它一些操作链接按钮，使用 **DIV+CSS** 布局页面模板。在模板设计时需要用到的一些信息如下所示：

**模板文件：**声明在视图目录中，并在用户模块目录（user）下，模板文件名称和操作名称相同为 index.tpl

**子模板：**包含公用的头部(header.tpl)和尾部(footer.tpl)模板文件；

**表单提交位置：**其中删除会用到表单，提交到当前模块的 del()操作中，提交方法使用 post；

**模板中用到的变量：**主要有两个，一个是保存列表数据的二维数组 \$users，另一个是保存分页字符串变量 \$fpage。其中 \$users 的格式如下：

```
$user = array (
    [0]=>array(
        'id'=>'12',           //用户编号
        'username'=>'user',   //用户名
        'regtime'=>'1312323323', //添加时间
        'email'=>'user@brophp.com', //用户电子邮箱
        'disable'=>'0',       //禁用状态
    ),
    [1]=>array(
        )
    id,username,regtime,email,disable
    .....
);
```

# 30.3.12 获取修改用户的界面操作 mod()

## 30.3.12.1 程序描述

如果需要修改数据库中一条用户记录，和添加用户相似需要先为操作者提供一个修改表单。操作者通过单击用户列表中的“修改”链接，就会通过链接中的 URL 访问到用户模块的 mod()操作中。在该操作中加载一个修改模板并输出，为操作者提供一个修改用户界面。

## 30.3.12.2 输入项

本操作的访问 URL 为：/brocms/admin.php/user/mod

- 1. 用户编号 （通过 GET 方法传递），格式为\$\_GET['id'];

## 30.3.12.3 输出项

- 1. 一条用户数据，以一个变量形式分配到模板中，类型为一个一维数组，变量名为\$users;
- 2. 用户组列表，以一个变量形式分配到模板中输出，类型为一个字符串，变量名为\$select;

## 30.3.12.4 算法

**第一步：**提示用户修改的规则；

**第二步：**通过 URL 的 GET 方式传递进来要修改的用户记录 ID，除了和获取添加界面操作一样，需要获取用户组的下列表分配到模板中使用外，并按用户 ID 从数据库的用户表中获取这个用户的全部信息数据，分配到 mod.tpl 模板中；

**第三步：**在表单中对应的位置回填这些数据；

**第四步：**加载并输出修改模板 mod.tpl

## 30.3.12.5 模板设计

需要为该操作声明一个修改用户的模板文件，在模板中只需要声明一个 HTML 表单，并将控制器中分配的用户信息对应的回填到每个表单项中。在模板设计时需要用到的一些信息如下所示：

**模板文件：**声明在视图目录中，并在用户模块目录（user）下，文件名称为 mod.tpl

**子模板：**包含公用的头部(header.tpl)和尾部/footer.tpl)模板文件

**表单提交位置：**当前模块的 update()操作中

**提交方法：**POST

**表单项的设计：**如表 30-6 所示

表 30-6 修改用户表单设计项

表单项	类型	名称	默认值	说明
用户编号	hidden	id	\$user.id	当前用户的编号



用户组	checkbox	gid	控制器中指定	提供用户组下拉列表
用户名称	text	username	\$user.username	可以使用中文，但禁止除[ @][.][_]以外的特殊符号
用户密码	password	userpwd	无	
确认密码	password	repwd	无	
电子邮箱	text	email	\$user.email	请正确添写你的电子邮件地址
用户性别	radio	sex	\$user.sex	男、女、保密
是否禁用	checkbox	disable	\$user.disable	禁用该用户

## 30.3.13 用户数据修改的操作 update()

### 30.3.13.1 程序描述

操作者在用户修改表单中修改一些信息以后，将新修改的数据以 POST 方法传递到 update() 操作中，处理数据后并修改数据表 user 中当前的记录。

### 30.3.13.2 输入项

本操作的访问 URL 为：/brocms/admin.php/user/update

本操作的输入项是用户在表单中录入的全部信息，以 HTTP 的 POST 方法传递到本操作中，所以全部保存在超全局数据 \$\_POST 中，其中需要对用户密码进行加密处理(使用 md5() 函数)。数组 \$\_POST 的格式如下所示：

```
$_POST=array(
    'id'=>'user',           //用户编号，作为修改的条件
    'username'=>'user',     //用户名
    'userpwd'=>'123456',    //用户密码
    'repwd'=>'123456',     //确认密码
    'sex'=>'1',             //用户性别
    'disable'=>'0',         //禁用状态
    'email'=>'user@brophp.com', //用户电子邮箱
    'sub'=>'添加用户'       //提交按钮
);
```

### 30.3.13.3 输出项

本操作中的输出项是要修改到数据库中的数据，是通过验证并处理过的数据。

### 30.3.13.4 算法

**第一步：**处理用户提交的数据；

**第二步：**使用 BroPHP 框架中的自动验证方式进行数据验证；

**第三步：**如果验证通过则修改数据表(user)中这条记录，并返回到用户记录列表页面；

**第四步：**如果修改失败则返回到修改表单界面重新进行修改，还要将表单中的数据值还原；

**第五步：**数据修改成功或是失败都要提示操作者。

## 30.3.14 删除用户操作 del()

### 30.3.14.1 程序描述

操作者可以通过每条用户记录后面提供的“删除”链接，单击删除一条用户记录，也可以通过每条记录前面的复选框选择多条记录删除。删除操作成功或是失败都要再返回到用户列表中。

### 30.3.14.2 输入项

本操作的访问 URL 为：/brocms/admin.php/user/del

1. 输入为两种类型的用户编号：

1) 一种是单个用户编号，通过 GET 方法传递格式为\$\_GET['id']。

2) 另一种是通过 POST 方法传递多个用户编号，格式为\$\_POST['id']。

### 30.3.14.3 输出项

无

### 30.3.14.4 算法

**第一步：**提示用户删除的规则；

**第二步：**组合用户的删除条件，如果是 GET 方式传递的用户 ID，则删除一条记录。如果是通过 POST 方式传递的数据则要删除多条记录。并且还要组合删除后的跳转位置；

**第三步：**删除指定的用户和用户的全部资源（包括用户发布的文章、评论、短消息和用户动态等）；

**第四步：**返回到删除操作前的用户列表；

**第五步：**处理删除成功或失败的提示状态；