EWShop

程序设计说明书

www.eduwork.cn

电子商城系统（EWShop）

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | 融职-EWShop-03 |
| 当前版本： | 1.0 |
| 作 者： | 高洛峰 |
| 完成日期： | 2020-08-15 |

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作 者 | 参 与 者 | 起止日期 | 备 注 |
| 1.0 | 高洛峰 | 教学组成员 | 2020-07-15  2020-08-20 | 《细说PHP》 |

## EW.1 引言

根据分析，我们开发的软件是一个SHOP系统，也就是我们常说的电子商城系统。就是在开放的网络环境下，基于浏览器/服务器应用方式，买卖双方不谋面地进行各种商贸活动，实现消费者的网上购物、商户之间的网上交易和在线电子支付，以及各种商务活动、交易活动、金融活动和相关的综合服务活动的一种商业运营模式。所以我们的设计思想就是围绕着这个目标展开的。本系统是基于PHP开发的，为了便于代码重用，我们使用EWShop 2.0框架进行开发，并且统一管理。

### EW.1.1 编写目的

本说明是EWShop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。本文档的新程序员对象为程序员和系统设计人员。软件开发小组的每一位参与人员应该仔细阅读本说明，以清楚产品在技术方面的要求和实现策略。

### EW.1.2 背景

本系统可以作为新程序员和融职IT教育学员的项目参考资料。除了学习项目开发参考使用，也可以通过本系统建设自己的电子商城系统。

软件系统的名称：EWShop电子商务管理系统。

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：学习猿地学员。

### EW.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出如下。

SHOP：电子商务系统。

API：插件管理（第三方开放的插件管理）。

PHP：本系统应用的自定义PHP开发框架。

### EW.1.4 使用技术

开发技术：PHP 7.4.0，Composer自己定义框架，JavaScript（JS、Ajax、jQuery）。

数据库技术：MySQL 8.0.18。

程序控制软件：PHPStorm。

### EW.1.5 参考文献

《细说PHP》。

软件需求说明书。

编码规范（北京智校云网络科技有限公司）。

项目进度文档。

API接口文档。

数据库设计说明书。

## EW.2 系统的结构

系统总体结构设计采用了MVC的设计模式，以及完全使用面向对象的思想开发，并根据自己定义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。

### EW.2.1 项目的目录结构

在需求阶段，我们将SHOP分为了前台和后台两个应用。因为是通过自己定义框架进行开发的，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（入口），也是路一个【路由】文件，并存放到项目目录EWShop（该目录存放在Web服务器的根目录下）下面。EWShop项目结构目录，如下所示：

|  |
| --- |
| |-- **EWShop**目录 #项目根目录  |**-- index.php** #主入口路由文件  |**-- config.inc.php** #项目的主配置文件  |**-- .htaccess** #项目根下的Apache配置文件  |**-- vendor** #Composer组件程序目录  |**-- app** #项目的应用程序目录  |**--controllers** #MVC模式控制器目录  |**--admin** #后台控制目录  |**--home** #前台控制器目录  |**--BaseControllers.php** #控制器的基类  |**--models** #MVC模式Model目录  |**--BaseDao.php** #数据库操作对象的基类  |**--views** #MVC模式的视图目录  |**--admin** #后台模板文件目录  |**--home** #前台第一套模板目录  |**--home2** #前台第一套模板目录  |**--public** #前后台公共资源（CSS,JS,images）目录  |**--helpers.php** #项目的自定义函数库文件  |**-- class** #项目自己定义类库文件  |**-- uploads** #资源上传保存的目录 |

按照自己定义框架的要求，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class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根目录下的uploads/文件夹中。

后台应用的目录为项目根目录下的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，存放在项目根目录中的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home相关的目录，用户可以进行模板风格切换。

### EW.2.2 模块结构

按照需求分析的结果，将后台应用分为17个模块，前台应用分为6个模块。根据主要功能确定前后和后台应用中每个模块的操作及操作权限，如表EW-1和表EW-2所示。

表EW-1 后台应用的模块操作说明

|  |  |  |
| --- | --- | --- |
| 模 块 | 操 作 | 权 限 |
| 登录管理 | 获取登录界面及处理登录、退出、获取验证码操作 | 无 |
| 操作界面管理 | 加载后台主操作页面，并提供默认信息 | 登录用户 |
| 商品管理 | 商品列表、商品添加、商品修改、商品删除、商品上下架、商品推荐和取消操作 | 登录用户 |
| 商品分类管理 | 商品分类列表、增加分类、修改分类、删除分类、对分类进行排序 | 登录用户 |

续表

|  |  |  |
| --- | --- | --- |
| 模 块 | 操 作 | 权 限 |
| 商品咨询管理 | 咨询列表、咨询回复、咨询删除 | 登录用户 |
| 商品评价管理 | 商品评价列表、对评价进行回复、咨询评价 | 登录用户 |
| 订单管理 | 获取订单列表、修改订单、删除订单、更改订单状态 | 登录用户 |
| 支付管理 | 获取支付列表、修改支付信息、删除支付方式、对支付方式进行排序 | 登录用户 |
| 文章分类管理 | 文章分类列表、增加分类、修改分类、删除分类、对分类进行排序 | 登录用户 |
| 文章管理 | 文章列表、增加文章、修改文章、删除文章 | 登录用户 |
| 单页管理 | 单页列表、增加单页、修改单页、删除单页 | 登录用户 |
| 会员管理 | 会员列表、修改会员、删除会员 | 登录用户 |
| 管理员管理 | 管理员列表、增加管理员、修改管理员信息、删除管理员 | 登录用户 |
| 基本信息管理 | 修改平台的基础信息 | 登录用户 |
| 缓存管理 | 获取缓存内容列表、更新缓存 | 登录用户 |
| 友情链接管理 | 查询友情链接列表、添加、修改、删除、排序 | 登录用户 |
| 广告管理 | 查询广告列表、添加、修改、删除、排序 | 登录用户 |

表EW-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 | 无 |
| 商品管理 | 获取商品内容、商品列表页面、添加商品收藏、添加商品咨询、添加商品评价 | 无 |
| 文章管理 | 获取文章内容、文件列表 | 无 |
| 用户管理 | 处理用户登录、退出和注册，获取用户订单列表、查看订单详情和删除订单，处理收藏和删除收藏，获取咨询列表，获取评价列表，设置用户基本信息和修改密码 | 登录用户 |
| 订单管理 | 将商品加入购物车、更改购物车中商品的数量、增加订单、选择支付方式、订单查询 | 登录用户 |
| 单页管理 | 获取并显示单页内容 | 无 |

### EW.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。首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表EW-3和表EW-4所示。

表EW-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() | 获取登录界面的处理登录操作 |
| logout() | 用户退出的操作 |
| vcode() | 获取验证码信息 |
| 操作界面管理 | Index | index() | 获取管理平台操作界面，并显示平台默认信息 |
| 商品管理 | Product | index() | 查询指定的商品列表并带分页 |
| add() | 添加一条新的商品数据 |
| mod() | 修改指定的商品信息 |
| del() | 按指定条件删除商品 |

续表

|  |  |  |  |
| --- | --- | --- | --- |
| 模 块 | 控制器类 | 操作方法 | 简要说明 |
|  |  | state() | 设置商品上下架的操作 |
| tuijian() | 对商品进行推荐和取消操作 |
| 商品分类管理 | Category | index() | 获取商品分类列表 |
| add() | 增加商品分类操作 |
| mod() | 修改指定的商品分类 |
| del() | 删除符合条件的商品分类 |
| order() | 对商品分类按指定的编号进行排序 |
| 商品咨询管理 | Ask | index() | 获取商品咨询列表 |
| mod() | 修改指定的商品咨询项 |
| del() | 删除符合条件的商品咨询项 |
| 商品评价管理 | Comment | index() | 获取商品评价列表 |
| mod() | 修改指定的商品评价项 |
| del() | 删除符合条件的商品评价项 |
| 订单管理 | Order | index() | 按指定条件获取特定的订单 |
| mod() | 修改订单的部分信息 |
| del() | 删除一些失效的订单 |
| state() | 修改订单的管理状态 |
| 支付管理 | Payway | index() | 获取支付方式列表 |
| mod() | 对支付方式进行修改 |
| del() | 删除不需要的支付方式 |
| order() | 对已有的支付方式按指定的序号进行排序 |
| 文章分类管理 | Aclass | index() | 获取文章分类列表 |
| add() | 增加文章分类操作 |
| mod() | 修改指定的文章分类 |
| del() | 删除符合条件的文章分类 |
| order() | 对文章分类按指定的编号进行排序 |
| 文章管理 | Article | index() | 获取文章列表 |
| add() | 增加文章操作 |
| mod() | 修改指定的文章信息 |
| del() | 删除符合条件的文章 |
| 单页管理 | Page | index() | 获取所有单页列表 |
| add() | 增加单页操作 |
| mod() | 修改指定的单页信息 |
| del() | 删除单页操作 |
| 会员管理 | User | index() | 获取注册会员列表 |
| mod() | 修改指定会员的基本信息 |
| del() | 删除指定条件的会员 |
| 管理员管理 | Admin | index() | 获取平台管理员列表 |
| add() | 增加管理员 |
|  |  | mod() | 修改管理员基本信息 |
| del() | 删除管理员 |
| 基本信息管理 | Setting | mod() | 设置系统平台基本信息 |
| 缓存管理 | Cache | index() | 获取所有缓存使用的列表 |
| update() | 清除指定的缓存 |
| 友情链接管理 | Link | index() | 查询友情链接列表 |
| add() | 增加友情链接操作 |
| mod() | 修改友情链接 |
| del() | 删除友情链接 |
| order() | 对友情链接按指定的编号进行排序 |
| 广告管理 | Ad | index() | 查询广告列表 |
| add() | 增加广告操作 |
| mod() | 修改广告 |
| del() | 删除广告 |
| order() | 对广告按指定的编号进行排序 |

表EW-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() | 获取首页全部内容并通过模板显示 |
| 商品管理 | Product | index() | 获取商品内容并显示 |
| plist() | 获取商品列表并显示 |
| collectadd() | 添加商品收藏 |
| askadd | 添加商品咨询 |
| commentadd() | 添加商品评价 |
| 文章管理 | Article | index() | 获取文章内容 |
| plist() | 获取文件列表 |
| 用户管理 | User | login() | 处理用户登录 |
| logout() | 处理退出 |
| register() | 处理用户注册 |
| order() | 获取用户订单列表 |
| orderview() | 查看订单详情 |
| orderdel() | 删除用户订单 |
| collect() | 处理收藏 |
| collectdel() | 删除收藏 |
| ask() | 获取咨询列表 |
| comment() | 获取评价列表 |
| base() | 设置用户基本信息 |
| pw() | 用户密码修改处理 |
| 订单管理 | Order | cartadd() | 将商品加入购物车 |
| cartnum() | 更改购物车中商品的数量 |
|  |  | add() | 增加订单 |
| pay() | 选择支付方式 |
| plist() | 订单查询 |
| 单页管理 | Page | index() | 获取并显示单页内容 |

按照自己定义框架的规则，将前台的控制器类声明在项目根目录下的app/controllers/home目录中，后台控制器类声明在app/controllers/admin目录中。并且控制器类所在的文件名要以“控制器类名.php”格式命名，文件名称一律小写。

## EW.3 用户管理模块设计说明

从本节开始，将逐个给出各个模块的设计考虑，但限于篇幅，本文档只给出一个用户模块的设计参考。

### EW.3.1 功能

网站在发展过程中一方面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），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。用户在网站中注册成为会员后，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，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

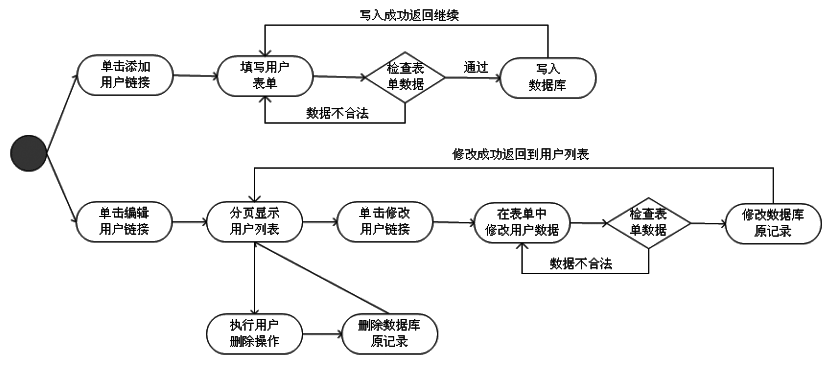
和添加用户类似，对已经注册的用户进行编辑修改。

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。

### EW.3.2 流程逻辑

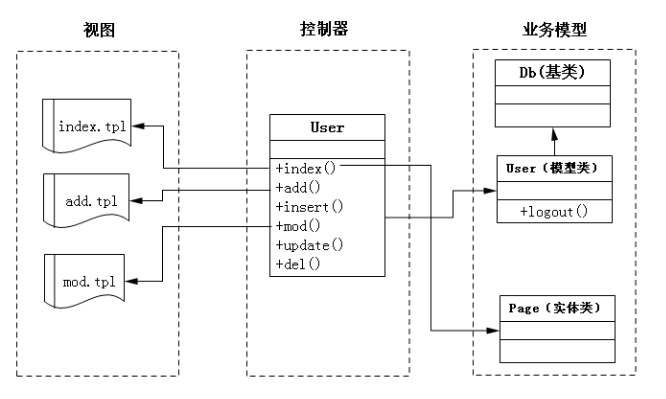
用户模块操作流程图如图EW-3所示。



图EW-3 用户模块操作流程图

### EW.3.3 接口

根据自己定义框架的规则，在设计用户管理模块时，需要在指定的位置设置一个控制器类（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EW-4所示。



图EW-4 用户模块的MVC模式设计结构

### EW.3.4 存储分配

本模块只需要对一张用户表操作即可，所有数据操作都通过Model类来完成。数据表结构详见《数据库设计数明书》。

### EW.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、默认条件等加上注释。

对使用的逻辑算法加上注释。

### EW.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。用户被管理员删除时，他的文章及评论等所有信息也会一同删除。

### EW.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，各个操作点是否已经实现。

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受的范围之内。

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。

### EW.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）。

### EW.3.9 获取添加用户的界面操作add()

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.html），提供给操作用户一个添加用户界面。

2．输入项

本操作的路由为Macaw**::**any(**"/admin/user/add"**, **'admin\User@add'**);

3．输出项

用户请求时并不需要提供其他的参数输入，输出添加用户的模板。

4．算法

**第一步：**提示用户添加的规则。

**第二步：**加载并输出添加模板add.html。

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.html。

**子模板：**包含公用的头部（header.html）和尾部（footer.html）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EW-5所示。

表EW-5 添加用户表单内容项

|  |  |  |  |
| --- | --- | --- | --- |
| 表 单 项 | 类 型 | 名 称 | 说 明 |
| 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 | pw | 初始密码为EWShop |
| 真实姓名 | text | tname | 用户的实名 |
| 电子邮箱 | text | email | 请正确填写你的电子邮件地址 |
| 用户电话 | text | phone | 用户的手机号码 |
| 联系地址 | text | address | 用户的联系地址 |

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”。

### EW.3.10 用户数据入库的操作insert()

1．程序描述

操作者在添加用户表单中录入一条用户信息后，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。

2．输入项

本操作的路由为Macaw**::**any(**"/admin/user/add"**, **'admin\User@add'**);

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、用户密码和确认密码、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  **'pw'=>'123456',**  //用户密码  **'tname'=>'gaoluofeng',**  //用户真实姓名  **'email'=>'g@eduwork.cn',**  //用户电子邮件  **'phone'=>'15801685668',**  //用户手机号  **'address'=>'haidian beijing'**  //用户联系地址  ); |

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。

4．算法

**第一步：**处理用户提交的数据。

**第二步：**使用自己定义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。

### EW.3.11 查询用户列表操作index()

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查找的用户，并且通过分页切换页面时也要保持查询条件。

2．输入项

本操作的路由为Macaw**::**any(**"/admin/user"**, **'admin\User@index'**);

搜索的用户名通过GET方法传递，格式为$\_GET['search']。

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。

4．算法

**第一步：**提示用户操作方式。

**第二步：**用户的记录列表为所有用户，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组合一个查询的WHERE条件和分页参数。

**第四步：**如果搜索条件为空，则查询全部记录。

**第五步：**加载输出index.html模板显示用户记录。

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他一些操作链接按钮，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.html。

**子模板：**包含公用的头部（header.html）和尾部（footer.html）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$users，另一个是保存分页的字符串变量$fpage。其中二维数组$users的格式如下：

|  |
| --- |
| **$user = array (**  **[0]=>array(**  **'id'=>'12',** //用户编号  **'username'=>'user',**  //用户名  **'tname'=>'gaoluofeng',**  //用户真实姓名  **'email'=>'g@eduwork.cn',**  //用户电子邮件  **'phone'=>'15801685668',**  //用户手机号  **'address'=>'haidian beijing'**  //用户联系地址  **'regtime'=>'1312323323',** //添加时间  **),**  **[1]=>array(**  **)**  **…**  **);** |

### EW.3.12 获取修改用户的界面操作mod()

1．程序描述

如果需要修改数据库中的一条用户记录，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。

2．输入项

本操作的路由为Macaw**::get**(**"/admin/user/mod/(:num)"**, **'admin\User@mod'**);

用户编号：通过GET方法传递，格式为$\_GET['id']。

3．输出项

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。

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是一样的操作，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.html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.html。

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地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.html。

**子模板：**包含公用的头部（header.html）和尾部（footer.html）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EW-6所示。

表EW-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 | pw |  | 初始密码为EWShop |
| 真实姓名 | text | tname | $user.tname | 用户的实名 |
| 电子邮箱 | text | email | $user.id | 请正确填写你的电子邮件地址 |

续表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 单 项 | 类 型 | 名 称 | 默 认 值 | 说 明 |
| 用户电话 | Text | phone | $user.id | 用户的手机号码 |
| 联系地址 | text | address | $user.id | 用户的联系地址 |

### EW.3.13 用户数据修改的操作update()

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，修改数据表user中当前的记录。

2．输入项

本操作的路由为Macaw**::post**(**"/admin/user/doupdate"**, **'admin\User@doupdate'**);

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  **'tname'=>'gaoluofeng',**  //用户真实姓名  **'email'=>'g@eduwork.cn',**  //用户电子邮件  **'phone'=>'15801685668',**  //用户手机号  **'address'=>'haidian beijing'',**  //用户联系地址  **'sub'=>'**添加用户**'** //提交按钮  ); |

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。

4．算法

**第一步：**处理用户提交的数据。

**第二步：**使用自己定义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）中的这条记录，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。

### EW.3.14 删除用户操作del()

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一起删除。删除操作成功或是失败都要返回到用户列表中。

2．输入项

本操作的路由为Macaw**::get**(**"/admin/user/del/(:num)"**, **'admin\User@del'**);

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']。

3．输出项

无

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。如果是通过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。