Web技术开发

课程设计报告



**计算机科学与技术学院**

**梁振铎**

**3013216096**

**2015年7月10日**

## 课程设计的目的和意义

许多常用的Web上的服务都与Web数据库应用程序有很大关系。在线邮件收发系统、在线购物、论坛与BBS、企业网站、门户网站等服务背后都有数据库。数据库应用程序是建立现代化网站的必需品。为了学习使用广泛、简单实用的Web数据库应用程序构建方法，MySQL则是最合适的数据库管理系统，PHP是最常被采用的Apache Web服务器相关模块。本课程设计的目的和意义在于学会创建应用所需要的技术，并学习如何创建一个安全、可靠、易维护、具扩充性的Web应用程序。

## 课程设计的主要任务

1. 在线商店是一个电子商务网站，其应用程序不管理统计、存货控制、薪资、订货与其他任务。
2. 用户可以选择商品并放入购物车；从第一件物品放入购物车开始计算，购物车内只保留一天内放入的商品，超时即清空；用户若关闭浏览器或登出系统，购物车即清空。
3. 每位用户有一辆购物车，可随时添加、清空车内商品，任意改变商品数目。
4. 网站用户可以匿名，在决定购买之前可一直保持匿名状态；若要购买则必须登录；若要登录，必须已经注册。
5. 注册时需要提供昵称、国家、省市、地址、邮编、电话等信息，并设置登录账号和密码；输入账号、密码即可登录。
6. 用户购买商品时，其订单存放于数据库中；购物车是尚未与客户产生关联的订单，用户提交支付信息后该订单可转换为完整订单。
7. 商品信息等存放于数据库中，可以通过信息检索对商品进行筛选。
8. 用户只能购买有库存的商品；当用户将购物车的内容转换为订单时，系统会检查库存能否满足订单，若不足则通知用户修改购买数目；当用户确认订购后，库存随之减少。

## 软件平台的选择和搭建

Apache是Apache Software Foundation的开放源代码的服务器软件，由于其跨平台和安全性被广泛使用，是最流行的Web服务器之一；

PHP是一种开源的通用计算机脚本语言，尤其适用于网络开发并可嵌入HTML中使用，也是最常用的Apache Web服务器相关模块。PHP本身具有集成网站及数据库环境工具，尤其适合开发Web数据库应用程序；

PEAR是PHP扩展应用程序库，它提供了标准化、结构化、经过整理的程序包，可应用于数据验证、访问数据库服务器、付款机制、web service使用、图形处理、读写文件等众多常见任务；

MySQL是最为普遍的数据库管理系统，它既快捷又容易使用，由其制造商MySQL AB以开放源代码的形式授权；

Zend Studio是Zend Technologies开发的PHP语言集成开发环境，也是目前市面上最优秀的PHP语言IDE；

1、安装Apache

下载解压Apache 2.4，将httpd.conf文件中的SRVROOT改为安装目录并安装；

2、安装PHP

下载PHP 5.6.7并解压到指定路径下，将Apache的路径信息等加入php.ini中；

3、安装PEAR软件包

下载最新的go-pear.phar文件到PHP目录下，在该目录下执行php go-pear.phar命令安装，并安装DB、HTML\_Template\_IT、Mail、HTTP等包；

4、安装Zend Studio

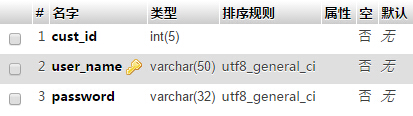
下载并安装Zend Studio 12.0，配置PHP信息，新建项目；

5、安装MySQL

下载并安装MySQL 5.6，设置账户信息并建立新数据库，项目中的db.inc文件中的$hostname、$databasename、$username、$password几项改为数据库信息以连接数据库；

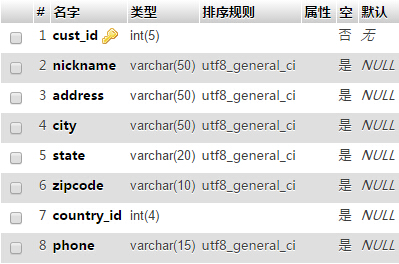
## 四、数据库设计

1、表users包含用户的帐户信息，有如下字段：



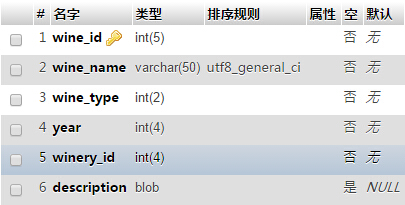
cust\_id表示客户号，user\_name表示用户名，password表示密码；

2、表customer包含用户的各项信息，有如下字段：



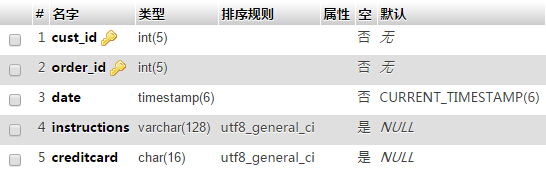
cust\_id表示客户号，nickname表示昵称，address表示地址，city表示城市，state表示省份，zipcode表示邮编，country\_id表示国家代号，phone表示电话；

3、表wine包含商品的各项信息，有如下字段：



wine\_id表示商品编号，wine\_name表示商品名称，wine\_type表示类型，year表示年份，winery\_id表示厂商代号，description表示描述；

4、表orders包含订单的各项用户信息，有如下字段：



cust\_id表示客户号，order\_id表示订单号，date表示交易日期，instructions表示备注，creditcard表示支付账户；

5、表items包含订单的各项商品信息，有如下字段：



cust\_id表示客户号，order\_id表示订单号，item\_id订单内商品号，wine\_id表示商品编号，qty表示数量，price表示价格；

6、表inventory包含库存信息，有如下字段：

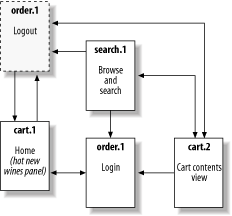
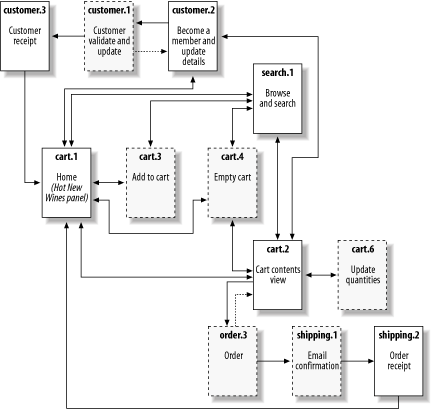


wine\_id表示商品编号，inventory\_id表示仓库编号，on\_hand表示库存量，cost表示售价；

7、此外，表countries包含国家代号信息，表region包含商品产地信息，winrey包含商品厂商信息，表wine\_type包含商品种类信息。

## 五、软件设计

在线商店首页为index.php，用户可在首页把推荐商品加入购物车或者使用其他功能。该应用程序由模板产生的可维护网页，并填入来自数据库的数据；用户驱动的查询及浏览，由用户提供限制搜索或浏览数据库的参数；数据输入与验证；以session管理技巧进行的用户追踪；用户验证与管理；错误处理的回执页面。



## 六、界面设计

整个站点的内容居中显示，左上角显示购物车按钮，右上角显示登录状态，底部显示功能按钮，各页面显示如下：

1、站点首页界面：



2、注册界面：



3、登陆界面：



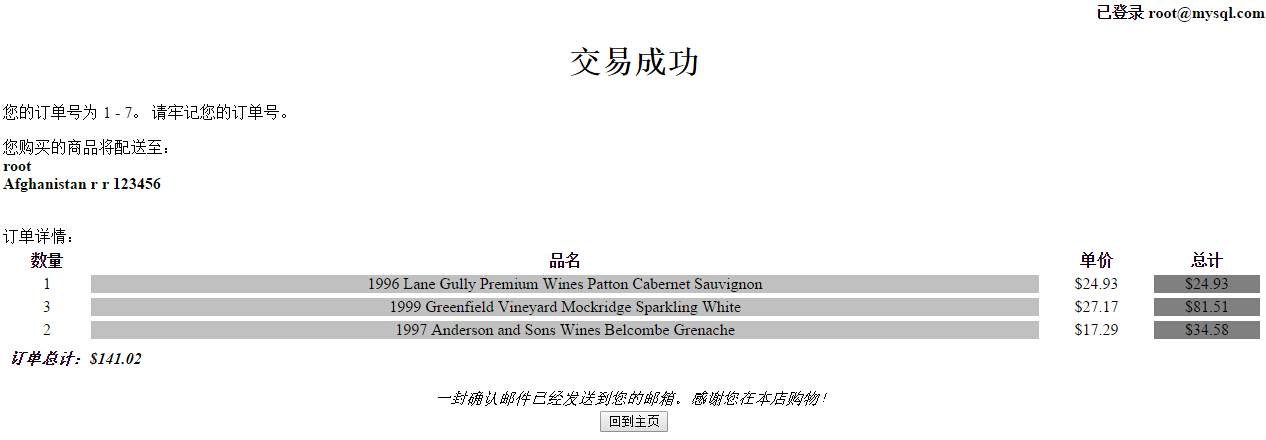
4、购物车界面：



5、支付界面：



6、交易成功界面：



## 算法设计（15分）

主要算法包含输入检测算法（检测是否输入为空、输入是否符合要求）、数据库查询（查询商品信息）、数据库修改算法（购物车内商品与数据库的交互）等。具体代码详见第十一节。

## 八、编程实现

在线商店应用程序开发是为了满足前述需求，是完整的PHP与MySQL Web数据库应用程序，并使用PEAR ITX模板及PEAR DB以获得HTML显示及数据库层。本应用有五个独立模块：客户管理、购物车、下单与购物、浏览与搜索、验证，源文件分别放在/customer、/cart、/order、/search、/auth下面。应用程序中有一组共享组件，包括扩展包PEAR ITX、验证函数、自定义错误处理器、验证函数、通用函数与常量。模板文件在/templates下，inc文件在/include下。具体代码详见第十一节。

## 九、所遇到的主要问题

模板中一些老版本PHP的函数已经弃用，ereg() 使用 preg\_match() 替代， ereg\_replace()和eregi() 使用 preg\_replace() 替代，mysql替换为mysqli等。

## 十、参考文献

《PHP & MySQL Web数据库应用开发指南》 ，

Hugb E. Williams, David Lane著 ，东南大学出版社

## 十一、源代码

主页源码：

|  |  |
| --- | --- |
| 1 | <?php |
| 2 | require\_once "DB.php"; |
| 3 | require\_once "includes/winestore.inc"; |
| 4 | require\_once "includes/template.inc"; |
| 5 |  |
| 6 | set\_error\_handler("customHandler"); |
| 7 |  |
| 8 | *function* showPanel($connection, &$template) |
| 9 | { |
| 10 | $query = "SELECT  wi.winery\_name, w.year, |
| 11 | w.wine\_name, w.wine\_id, w.description |
| 12 | FROM wine w, winery wi, inventory i |
| 13 | WHERE w.winery\_id = wi.winery\_id |
| 14 | AND w.wine\_id = i.wine\_id |
| 15 | AND w.description IS NOT NULL |
| 16 | GROUP BY w.wine\_id |
| 17 | ORDER BY i.date\_added DESC LIMIT 3"; |
| 18 |  |
| 19 | $result = $connection->query($query); |
| 20 | if (*DB*::isError($result)) |
| 21 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 22 |  |
| 23 | while ($row =$result->fetchRow(DB\_FETCHMODE\_ASSOC)) |
| 24 | { |
| 25 | $template->setCurrentBlock("row"); |
| 26 | $template->setVariable("YEAR", $row["year"]); |
| 27 | $template->setVariable("WINERY", $row["winery\_name"]); |
| 28 | $template->setVariable("WINE", $row["wine\_name"]); |
| 29 | $template->setVariable("DESCRIPTION", |
| 30 | $row["description"]); |
| 31 | $template->setVariable("VARIETIES",showVarieties |
| 32 | ($connection, $row["wine\_id"])); |
| 33 | $price = showPricing($connection, $row["wine\_id"]); |
| 34 | $template->setVariable("BOTTLE\_PRICE", |
| 35 | sprintf("%.2f", $price)); |
| 36 | $template->setCurrentBlock("link"); |
| 37 | $template->setVariable("SCRIPT", S\_ADDTOCART); |
| 38 | $template->setVariable("QTY", "1"); |
| 39 | $template->setVariable("WINE\_ID", $row["wine\_id"]); |
| 40 | $template->setVariable("STRING", "加入购物车"); |
| 41 | $template->parseCurrentBlock("link"); |
| 42 | $template->setCurrentBlock("row"); |
| 43 | $template->parseCurrentBlock("row"); |
| 44 | } |
| 45 | } |
| 46 |  |
| 47 | session\_start(); |
| 48 |  |
| 49 | $template = new *winestoreTemplate*(T\_HOME); |
| 50 | $connection = *DB*::connect($dsn, true); |
| 51 | if (*DB*::isError($connection)) |
| 52 | trigger\_error($connection->getMessage(), E\_USER\_ERROR); |
| 53 |  |
| 54 | showPanel($connection, $template); |
| 55 | $template->showWinestore(SHOW\_ALL, B\_ALL & ~B\_UPDATE\_CART & |
| 56 | ~B\_HOME & ~B\_PASSWORD & |
| 57 | ~B\_PURCHASE & ~B\_EMPTY\_CART); |
| 58 | ?> |

组件定义：

|  |  |
| --- | --- |
| 1 | <?php |
| 2 | require\_once 'db.inc'; |
| 3 | require\_once 'customHandler.inc'; |
| 4 | define("D\_INSTALL\_PATH", "D:/Program Files/Apache24/htdocs"); |
| 5 | define("D\_WEB\_PATH", "/store/"); |
| 6 | define("D\_CART", D\_WEB\_PATH . "cart/"); |
| 7 | define("D\_CARTIMAGES", D\_CART . "images/"); |
| 8 | define("D\_CUSTOMER", D\_WEB\_PATH . "customer/"); |
| 9 | define("D\_AUTH", D\_WEB\_PATH . "auth/"); |
| 10 | define("D\_ORDER", D\_WEB\_PATH . "order/"); |
| 11 | define("D\_SEARCH", D\_WEB\_PATH . "search/"); |
| 12 | define("D\_TEMPLATES", D\_INSTALL\_PATH . D\_WEB\_PATH . "templates/"); |
| 13 | define("S\_MAIN", D\_WEB\_PATH . "index.php"); |
| 14 | define("S\_ADDTOCART", D\_CART . "addtocart.php"); |
| 15 | define("S\_EMPTYCART", D\_CART . "emptycart.php"); |
| 16 | define("S\_SHOWCART", D\_CART . "showcart.php"); |
| 17 | define("S\_UPDATECART", D\_CART . "updatecart.php"); |
| 18 | define("S\_ORDER\_1", D\_ORDER . "order-step1.php"); |
| 19 | define("S\_ORDER\_2", D\_ORDER . "order-step2.php"); |
| 20 | define("S\_ORDER\_3", D\_ORDER . "order-step3.php"); |
| 21 | define("S\_ORDER\_4", D\_ORDER . "order-step4.php"); |
| 22 | define("S\_ORDERRECEIPT", D\_ORDER . "receipt.php"); |
| 23 | define("S\_SEARCH", D\_SEARCH . "search.php"); |
| 24 | define("S\_SEARCHFORM", D\_SEARCH . "searchform.php"); |
| 25 | define("S\_DETAILS", D\_CUSTOMER . "details.php"); |
| 26 | define("S\_VALIDATE", D\_CUSTOMER . "validate.php"); |
| 27 | define("S\_CUSTRECEIPT", D\_CUSTOMER . "receipt.php"); |
| 28 | define("S\_LOGOUT", D\_AUTH . "logout.php"); |
| 29 | define("S\_LOGIN", D\_AUTH . "login.php"); |
| 30 | define("S\_LOGINCHECK", D\_AUTH . "logincheck.php"); |
| 31 | define("S\_PASSWORD", D\_AUTH . "password.php"); |
| 32 | define("S\_CHANGEPASSWORD", D\_AUTH . "changepassword.php"); |
| 33 | define("S\_PASSWORDRECEIPT", D\_AUTH . "receipt.php"); |
| 34 | define("T\_SKELETON", "winestore.tpl"); |
| 35 | define("T\_HOME", "index.tpl"); |
| 36 | define("T\_SHOWCART", "showcart.tpl"); |
| 37 | define("T\_DETAILS", "details.tpl"); |
| 38 | define("T\_CUSTRECEIPT", "custreceipt.tpl"); |
| 39 | define("T\_LOGIN", "login.tpl"); |
| 40 | define("T\_PASSWORD", "password.tpl"); |
| 41 | define("T\_PASSWORDRECEIPT", "passwordreceipt.tpl"); |
| 42 | define("T\_EMAIL", "email.tpl"); |
| 43 | define("T\_ORDERRECEIPT", "orderreceipt.tpl"); |
| 44 | define("T\_SEARCH", "search.tpl"); |
| 45 | define("T\_SOURCE", "source.tpl"); |
| 46 | define("I\_CART\_OFF", D\_CARTIMAGES . "cart\_off.jpg"); |
| 47 | define("I\_CART\_ON", D\_CARTIMAGES . "cart\_on.jpg"); |
| 48 | define("B\_EMPTY\_CART", 1); |
| 49 | define("B\_SHOW\_CART", 2); |
| 50 | define("B\_UPDATE\_CART", 4); |
| 51 | define("B\_PURCHASE", 8); |
| 52 | define("B\_SEARCH", 16); |
| 53 | define("B\_HOME", 32); |
| 54 | define("B\_DETAILS", 64); |
| 55 | define("B\_LOGINLOGOUT", 128); |
| 56 | define("B\_PASSWORD", 256); |
| 57 | define("B\_ALL", 511); |
| 58 | define("NO\_CART", 1); |
| 59 | define("SHOW\_ALL", 2); |
| 60 | define("SEARCH\_ROWS", 12); |
| 61 | define("ERROR\_FILE", "/tmp/php\_error\_log"); |
| 62 | define("FILE\_ERRORS", false); |
| 63 | define("SCREEN\_ERRORS", true); |
| 64 | $dsn = "mysqli://{$username}: |
| 65 | {$password}@{$hostname}/{$databasename}"; |
| 66 |  |
| 67 | *function* pearclean($array, $index, $maxlength, $connection) |
| 68 | { |
| 69 | if (isset($array["{$index}"])) |
| 70 | { |
| 71 | $input = trim(substr($array["{$index}"], 0, $maxlength)); |
| 72 | $link = mysqli\_connect("localhost","root", |
| 73 | "19950608","winestore"); |
| 74 | $input = mysqli\_real\_escape\_string($link, $input); |
| 75 | return ($input); |
| 76 | } |
| 77 | return NULL; |
| 78 | } |
| 79 |  |
| 80 | *function* getCust\_id($user\_name, $connection = null) |
| 81 | { |
| 82 | global $dsn; |
| 83 | if (!isset($connection)) |
| 84 | { |
| 85 | $connection = *DB*::connect($dsn, false); |
| 86 | if (*DB*::isError($connection)) |
| 87 | trigger\_error($connection->getMessage(), E\_USER\_ERROR); |
| 88 | } |
| 89 |  |
| 90 | $query = "SELECT cust\_id FROM users WHERE |
| 91 | user\_name = '{$user\_name}'"; |
| 92 |  |
| 93 | $result = $connection->query($query); |
| 94 | if (*DB*::isError($result)) |
| 95 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 96 |  |
| 97 | $row = $result->fetchRow(DB\_FETCHMODE\_ASSOC); |
| 98 |  |
| 99 | return($row["cust\_id"]); |
| 100 | } |
| 101 |  |
| 102 | *function* showWine($wineId, $connection = null) |
| 103 | { |
| 104 | global $dsn; |
| 105 |  |
| 106 | $wineQuery = "SELECT year, winery\_name, wine\_name |
| 107 | FROM winery, wine |
| 108 | WHERE wine.winery\_id = winery.winery\_id |
| 109 | AND wine.wine\_id = {$wineId}"; |
| 110 |  |
| 111 | if (!isset($connection)) |
| 112 | { |
| 113 | $connection = *DB*::connect($dsn, false); |
| 114 | if (*DB*::isError($connection)) |
| 115 | trigger\_error($connection->getMessage(), E\_USER\_ERROR); |
| 116 | } |
| 117 |  |
| 118 | $result = $connection->query($wineQuery); |
| 119 |  |
| 120 | if (*DB*::isError($result)) |
| 121 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 122 |  |
| 123 | $row = $result->fetchRow(DB\_FETCHMODE\_ASSOC); |
| 124 | $output = "{$row["year"]} |
| 125 | {$row["winery\_name"]} {$row["wine\_name"]}"; |
| 126 | $output .= showVarieties($connection, $wineId); |
| 127 | return $output; |
| 128 | } |
| 129 |  |
| 130 | *function* showVarieties($connection, $wineID) |
| 131 | { |
| 132 | $query = "SELECT gv.variety |
| 133 | FROM grape\_variety gv, wine\_variety wv, wine w |
| 134 | WHERE w.wine\_id = wv.wine\_id |
| 135 | AND wv.variety\_id = gv.variety\_id |
| 136 | AND w.wine\_id = {$wineID} |
| 137 | ORDER BY wv.id"; |
| 138 |  |
| 139 | $result = $connection->query($query); |
| 140 |  |
| 141 | if (*DB*::isError($result)) |
| 142 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 143 |  |
| 144 | $varieties = ""; |
| 145 | while ($row = $result->fetchRow(DB\_FETCHMODE\_ASSOC)) |
| 146 | $varieties .= " {$row["variety"]}"; |
| 147 |  |
| 148 | return $varieties; |
| 149 | } |
| 150 |  |
| 151 | *function* showPricing($connection, $wineID) |
| 152 | { |
| 153 | $query = "SELECT min(cost) FROM inventory |
| 154 | WHERE wine\_id = {$wineID}"; |
| 155 |  |
| 156 | $result = $connection->query($query); |
| 157 |  |
| 158 | if (*DB*::isError($result)) |
| 159 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 160 |  |
| 161 | $row = $result->fetchRow(DB\_FETCHMODE\_ASSOC); |
| 162 |  |
| 163 | $price = $row["min(cost)"]; |
| 164 |  |
| 165 | return $price; |
| 166 | } |
| 167 |  |
| 168 | *function* showCountry($country\_id, $connection) |
| 169 | { |
| 170 | $query = "SELECT country FROM countries WHERE |
| 171 | country\_id = {$country\_id}"; |
| 172 |  |
| 173 | $result = $connection->query($query); |
| 174 |  |
| 175 | if (*DB*::isError($result)) |
| 176 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 177 |  |
| 178 | $countryRow = $result->fetchRow(DB\_FETCHMODE\_ASSOC); |
| 179 |  |
| 180 | return($countryRow["country"]); |
| 181 | } |
| 182 |  |
| 183 | *function* showTitle($title\_id, $connection) |
| 184 | { |
| 185 | $query = "SELECT title FROM titles WHERE |
| 186 | title\_id = {$title\_id}"; |
| 187 |  |
| 188 | $result = $connection->query($query); |
| 189 |  |
| 190 | if (*DB*::isError($result)) |
| 191 | trigger\_error($result->getMessage(), E\_USER\_ERROR); |
| 192 |  |
| 193 | $titleRow = $result->fetchRow(DB\_FETCHMODE\_ASSOC); |
| 194 |  |
| 195 | return($titleRow["title"]); |
| 196 | } |
| 197 |  |
| 198 | ?> |

输入验证：

|  |  |
| --- | --- |
| 1 | <?php |
| 2 | *function* checkMandatory |
| 3 | ($field, $errorString, $errors, $formVars) |
| 4 | { |
| 5 | if (!isset($\_SESSION["{$formVars}"]["{$field}"]) || |
| 6 | empty($\_SESSION["{$formVars}"]["{$field}"])) |
| 7 | { |
| 8 | $\_SESSION["{$errors}"]["{$field}"] = |
| 9 | "The {$errorString} field cannot be blank."; |
| 10 | return false; |
| 11 | } |
| 12 | return true; |
| 13 | } |
| 14 |  |
| 15 | *function* checkMinAndMaxLength |
| 16 | ($field, $minlength, $maxlength, |
| 17 | $errorString, $errors, $formVars) |
| 18 | { |
| 19 | if (isset($\_SESSION["{$formVars}"]["{$field}"]) && |
| 20 | (strlen($\_SESSION["{$formVars}"]["{$field}"])<$minlength|| |
| 21 | strlen($\_SESSION["{$formVars}"]["{$field}"])>$maxlength)) |
| 22 | { |
| 23 | $\_SESSION["{$errors}"]["{$field}"] = |
| 24 | "The {$errorString} field must be greater than or equal to". |
| 25 | "{$minlength} and less than or equal to {$maxlength} ". |
| 26 | "characters in length."; |
| 27 | return false; |
| 28 | } |
| 29 | return true; |
| 30 | } |
| 31 |  |
| 32 | *function* checkZipcode |
| 33 | ($field, $errorString, $errors, $formVars) |
| 34 | { |
| 35 | if (isset($\_SESSION["{$formVars}"]["{$field}"]) && |
| 36 | !preg\_match("/^([0-9]{5,6})$/", |
| 37 | $\_SESSION["{$formVars}"]["{$field}"])) |
| 38 | { |
| 39 | $\_SESSION["{$errors}"]["{$field}"] = |
| 40 | "The zipcode must be 5 or 6 digits in length"; |
| 41 | return false; |
| 42 | } |
| 43 | return true; |
| 44 | } |
| 45 |  |
| 46 | *function* checkPhone |
| 47 | ($field, $errorString, $errors, $formVars) |
| 48 | { |
| 49 | $validPhoneExpr = |
| 50 | "/^([0-9]{2,3}[ ]?)?[0-9]{4}[ ]?[0-9]{4}$/"; |
| 51 |  |
| 52 | if (isset($\_SESSION["{$formVars}"]["{$field}"]) && |
| 53 | !preg\_match($validPhoneExpr, |
| 54 | $\_SESSION["{$formVars}"]["{$field}"])) |
| 55 | { |
| 56 | $\_SESSION["{$errors}"]["{$field}"] = |
| 57 | "The {$field} field must be 8 digits in length, " . |
| 58 | "with an optional 2 or 3 digit area code"; |
| 59 | return false; |
| 60 | } |
| 61 | return true; |
| 62 | } |
| 63 |  |
| 64 | *function* emailCheck |
| 65 | ($field, $errorString, $errors, $formVars) |
| 66 | { |
| 67 | $validEmailExpr = |
| 68 | "^[0-9a-z~!#$%&\_-]([.]?[0-9a-z~!#$%&\_-])\*". |
| 69 | "@[0-9a-z~!#$%&\_-]([.]?[0-9a-z~!#$%&\_-])\*$"; |
| 70 |  |
| 71 | if (!eregi($validEmailExpr, |
| 72 | $\_SESSION["{$formVars}"]["{$field}"])) |
| 73 | { |
| 74 | $\_SESSION["{$errors}"]["{$field}"] = |
| 75 | "The email must be in the name@domain format."; |
| 76 | return false; |
| 77 | } |
| 78 |  |
| 79 | if (function\_exists("getmxrr") && |
| 80 | function\_exists("gethostbyname")) |
| 81 | { |
| 82 | $maildomain = |
| 83 | substr(strstr($\_SESSION["{$formVars}"] |
| 84 | ["{$field}"], '@'), 1); |
| 85 |  |
| 86 | if (!(getmxrr($maildomain, $temp) || |
| 87 | gethostbyname($maildomain) != $maildomain)) |
| 88 | { |
| 89 | $\_SESSION["{$errors}"]["{$field}"] = |
| 90 | "The email domain does not exist."; |
| 91 | return false; |
| 92 | } |
| 93 | } |
| 94 |  |
| 95 | return true; |
| 96 | } |
| 97 |  |
| 98 | *function* checkCard($field, $errors, $formVars) |
| 99 | { |
| 100 | if (!preg\_match("/^[0-9 ]\*$/", |
| 101 | $\_SESSION["{$formVars}"]["{$field}"])) |
| 102 | { |
| 103 | $\_SESSION["{$errors}"]["{$field}"] = |
| 104 | "Card number must contain only digits and spaces."; |
| 105 | return false; |
| 106 | } |
| 107 |  |
| 108 | $\_SESSION["{$formVars}"]["{$field}"] = preg\_replace |
| 109 | ('[ ]', '', $\_SESSION["{$formVars}"]["{$field}"]); |
| 110 |  |
| 111 | $firstFour = intval(substr( |
| 112 | $\_SESSION["{$formVars}"]["{$field}"], 0, 4)); |
| 113 | $type = ""; |
| 114 | $length = 0; |
| 115 |  |
| 116 | if ($firstFour >= 8000 && $firstFour <= 8999) |
| 117 | { |
| 118 | $type = "SurchargeCard"; |
| 119 | $length = 16; |
| 120 | } |
| 121 |  |
| 122 | if (empty($type)) |
| 123 | { |
| 124 | $\_SESSION["{$errors}"]["{$field}"] = |
| 125 | "Please check your card details."; |
| 126 | return false; |
| 127 | } |
| 128 |  |
| 129 | if (strlen($\_SESSION["{$formVars}"] |
| 130 | ["{$field}"]) != $length) |
| 131 | { |
| 132 | $\_SESSION["{$errors}"]["{$field}"] = |
| 133 | "Card number must contain {$length} digits."; |
| 134 | return false; |
| 135 | } |
| 136 |  |
| 137 | $check = 0; |
| 138 |  |
| 139 | for($x=$length-1;$x>=0;$x-=2) |
| 140 | $check += intval(substr( |
| 141 | $\_SESSION["{$formVars}"]["{$field}"], $x, 1)); |
| 142 |  |
| 143 | for($x=$length-2;$x>=0;$x-=2) |
| 144 | { |
| 145 | $double = intval(substr($\_SESSION["{$formVars}"] |
| 146 | ["{$field}"], $x, 1)) \* 2; |
| 147 | if ($double >= 10) |
| 148 | $check += $double - 9; |
| 149 | else |
| 150 | $check += $double; |
| 151 | } |
| 152 | ?> |