ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ РФ

ЭЛЕКТРОСТАЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

(филиал) ФГОУ ВПО «Национальный исследовательский технологический университет «МИСиС»»

Кафедра прикладной информатики

**Курсовая научно-исследовательская работа:**

**«Система сбора и обработки данных на малом предприятии»**

Выполнил студент группы ДИС-06-1 Иванов Е.В.

Проверил Жучин А. В.

Электросталь 2010 г.

Оглавление

[Введение 3](#_Toc281046042)

[1. Испытательная лаборатория ООО «МАШПРОМЭКСПЕРТ» 5](#_Toc281046043)

[2. Требования к разрабатываемой системе 7](#_Toc281046044)

[3. База данных 8](#_Toc281046045)

[3.1 MySQL 8](#_Toc281046046)

[3.2 Класс DB для работы с базой данных 9](#_Toc281046050)

[3.3 Класс SqlData для работы с данными объекта 10](#_Toc281046051)

[4. Система авторизации пользователей 13](#_Toc281046052)

[4.1 Класс пользователя User 13](#_Toc281046053)

[4.2 Класс управления авторизацией AuthManager 14](#_Toc281046054)

[5. Ввод и обработка данных 17](#_Toc281046055)

[5.1 Формат RTF 17](#_Toc281046056)

[5.2 Обработка введенных данных 18](#_Toc281046057)

[Заключение 20](#_Toc281046058)

[Список используемой литературы 21](#_Toc281046059)

# Введение

В настоящее время все большее и большее распространение, как в производстве, так и в документообороте предприятий находит компьютерная техника, все шире и шире становится перечень охватываемых ею задач. Постоянно растет объем и сложность обрабатываемой информации, требуются все новые и новые виды ее представления. Как показывает практика, большинство пользователей уже не представляет себе, как бы они выполняли свою работу, с которой прекрасно справлялись еще 5 – 10 лет назад, без помощи компьютера. Вот только некоторые из преимуществ использования вычислительной техники при работе, с какими – либо данными, документами:

* возможность оперативного контроля за достоверностью информации, уменьшается число возможных ошибок, при генерировании производных данных;
* сразу после ввода, данные могут участвовать в различных операциях; возможность быстрого получения необходимых отчетов;
* необходимая информация может быть быстро найдена (в том числе с помощью контекстного поиска), поиск которой в обычной папке с документами затруднен;
* существенная экономия времени и людских ресурсов при выполнении операций, связанных с обработкой информации;
* позволяет быстро и в наиболее подходящем для конкретного человека виде просмотреть данные, что повышает их восприятие.

Все, выше изложенные преимущества, на сегодняшний день оценены не только рядовыми пользователями, но и коммерческими структурами, особенно это касается возможности контроля и поиска информации. В нашем случае испытательная лаборатория заинтересована иметь единую, электронную базу по всем проводимым сертификациям, так как наличие такой базы существенно упростит контроль за проведенными испытаниями и поможет формировать отчеты.

# 1. Испытательная лаборатория ООО «МАШПРОМЭКСПЕРТ»

Испытательная лаборатория ООО «МАШПРОМЭКСПЕРТ» аккредитована Федеральным Агентством по Техническому Регулированию и Метрологии на проведение сертификационных испытаний продукции в системе ГОСТ Р.

Аттестат аккредитации № РОСС RU.0001.21ММ18, действителен до 02.09.2015 г.

Лаборатория оснащена современным испытательным оборудованием и новейшими средствами измерений, что позволяет проводить испытания на передовом техническом уровне, с высокой степенью автоматизации процессов испытаний, измерений и оценки результатов.

Коллектив лаборатории – это эксперты, имеющие многолетний опыт испытательной практики, а также высококвалифицированные специалисты в области сертификации продукции и оборудования.

Лаборатория проводит широкий спектр электрических, климатических, механических, температурных, функциональных, специальных и др. испытаний для различных видов продукции. Основная специализация лаборатории – испытания энергетического и электротехнического оборудования.

Область аккредитации:

* Оборудование энергетическое
* Машины электрические
* Асинхронные, синхронные, коллекторные, постоянного тока, бесконтактные
* Оборудование и материалы электротехнические
* Трансформаторы и трансформаторное оборудование
* Аппаратура высоковольтная, силовая преобразовательная техника, хемотроника
* Аппараты электрические на напряжение до 1000 В. Комплектные устройства и электроустановки на напряжение до 1000 В
* Оборудование светотехническое , лампы электрические
* Изделия культурно-бытового назначения и хозяйственного обихода
* Источники тока химические, физические, генераторы электрохимические, термоэлектрические и термоэмиссионные
* Материалы и изделия электроизоляционные, электроугольные и электрометаллокерамические

# 2. Требования к разрабатываемой системе

Разрабатываемая система сбора и обработки данных на предприятии ООО «МАШПРОМЭКСПЕРТ» должна выполнять следующие функции:

1. Система авторизации пользователей.
   * Разграничение прав пользователей.
   * Добавление, редактирование и удаление пользователей.
   * Запрет доступа к системе незарегестирированным пользователям.
2. Система управления клиентами лаборатории.
   * Добавление, редактирование и удаление клиентов.
3. Система ввода и обработки данных.
   * Ввод данных различного типа, автозаполнение полей при определенных условиях.
   * Формирование документов на основе введенных данных.
4. Система истории.
   * Хранение информации о введенных данных.
   * Поиск документов по различным параметрам.
   * Отчеты о производительности работников.

**3. База данных**

**3.1 MySQL**

MySQL - свободная система управления базами данных (СУБД). MySQL является собственностью компании Sun Microsystems, осуществляющей разработку и поддержку приложения. Распространяется под GNU General Public License и под собственной коммерческой лицензией, на выбор.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

На сегодняшний день СУБД MySQL является одной из самых известных, надежных и быстрых из всего семейства существующих СУБД.

Одной из причин являются правила ее распространения — за нее не надо платить деньги и распространяется она вместе со своими исходными текстами, другая причина – это то, что MySQL относительно быстрая СУБД. PostgreSql, например, также распространяется под лицензией GNU GPL, но она не получила столь широкого распространения. Одна из причин — это заметная медлительность. Итак, две главные причины популярности MySQL: цена и производительность.

MySQL написан под десятки видов операционных систем. Это и FreeBSD,  
OpenBSD, MacOS, OS/2, SunOS, Win9x/00/NT и Linux. Сегодня MySQL особенно распространена на платформах Linux и Windows.

Принцип работы СУБД MySQL аналогичен принципу работы любой СУБД, использующей SQL (Structured Query Language, язык структурированных запросов) в качестве командного языка для создания/удаления баз данных, таблиц, для пополнения таблиц данными, для осуществления выборки данных.

**3.2 Класс DB для работы с базой данных**

Для работы с базой данных был разработан класс DB на основе шаблона проектирования Singleton. Данный шаблон гарантирует, что у класса есть только один экземпляр, и предоставляет к нему глобальную точку доступа. Это идеально подходит для работы с базой данных, т.к. в любом месте программы мы имеем доступ к базе и при этом всегда будем уверены, что к базе существует только одно соединение.

Рассмотрим работу данного класса. В конструкторе класса происходит создание подключения к базе данных и указатель на подключение записывается в переменную $connection. Данные о настройках подключения подгружаются из файла sql\_config.php.

private function \_\_construct()

{

include ($\_SERVER['DOCUMENT\_ROOT'].'/inc/sql\_config.php');

$this->connection = mysql\_connect($mysql\_host, $mysql\_login, $mysql\_psswd) or die(mysql\_error);

mysql\_select\_db($mysql\_dbname);

mysql\_query("SET NAMES 'cp1251'");

}

Для получения доступа к классу необходимо вызвать функцию DB::get().

Она проверяет существование экземпляра класса и если его не существует, то вызывает конструктор класса. Указатель на экземпляр класса записывается в статичную переменную $instance.

public static function get()

{

if ( !(self::$instance instanceof DB) )

{

self::$instance = new DB();

}

return self::$instance;

}

Запросы к базе данных осуществляются через функцию DB::sql().

public function sql($query)

{

$result = mysql\_query($query, $this->connection) or die(mysql\_error());

return $result;

}

Для более простого обращения к базе данных можно использовать глобальную функцию sql(), которая является оберткой над функцией DB::sql().

function sql($query)

{

return DB::get()->sql($query);

}

**3.3 Класс SqlData для работы с данными объекта**

Предполагается, что объекты в разрабатываемой системе будут хранить свои данные в базе, и для облегчения доступа к ним был написан класс SqlData. Данный класс использует функцию sql() для обращения к базе данных. Для создания экземпляра класса нам необходимо знать следующие параметры:

$table – таблица в которой производится выборка.

$selector – Поле, которое точно идентифицирует данный объект.

$selector\_value – Значение, которое является уникальным для данного объекта.

Конструктор класса:

public function \_\_construct($table, $selector, $selector\_value)

{

$this->\_table = $table;

$this->\_selector = $selector;

$this->\_selector\_value = $selector\_value;

}

Функция для выборки данных getData() является основной функцией данного класса. В ней реализован механизм кеширования для предотвращения лишних запросов к базе данных. Закешированные данные хранятся в переменной класса $\_data. В зависимости от входных параметров возвращается значение конкретного поля или значения всех полей объекта.

public function getData($field='')

{

if ($field == $this->\_selector) return $this->\_selector\_value;

else

{

if (empty($this->\_data))

{

$query = "SELECT \* FROM {$this->\_table} WHERE {$this->\_selector} = '{$this->\_selector\_value}'";

$this->\_data = mysql\_fetch\_assoc(sql($query));

}

if (empty($field)) return $this->\_data;

else if (isset($this->\_data[$field])) return $this->\_data[$field];

else

{

$query = "SELECT $field FROM {$this->\_table} WHERE {$this->\_selector} = '{$this->\_selector\_value}'";

$result = mysql\_fetch\_assoc(sql($query));

$this->\_data[$field] = $result[$field];

return $this->\_data[$field];

}

}

}

Также присутствуют функции:

setData() – для выставления известных данных объекту, без занесения их в базу данных.

updateData() – для изменения данных объекта и сохрания их в базе данных.

clear() – очищает закешированные данныею

public function setData($data)

{

$this->\_data = $data;

}

public function updateData($data)

{

$query = update\_sql\_query($data, $this->\_table, "WHERE {$this->\_selector} = '{$this->\_selector\_value}'");

sql($query);

$this->\_data = array\_merge($this->\_data, $data);

}

public function clear()

{

$this->\_data = null;

}

**4. Система авторизации пользователей**

**4.1 Класс пользователя User**

Класс User хранит информацию о конкретном пользователе. Он позволяет получать информацию о пользователе и проверять уровень доступа к различным разделам системы.

Для создания экземпляра класса необходимо знать только id пользователя.

public function \_\_construct($id)

{

$this->\_id = $id;

}

Для определения уровня доступа пользователя используются функции:

checkPermission() – проверяет доступ пользователя к компоненту системы.

diePermission() – проверяет доступ пользователя к компоненту системы и прекращает выполнение программы если доступ запрещен.

getGroupInfo() – получение информации о группе пользователя.

changeGroup() – изменение группы пользователя.

public function checkPermission($name)

{

return $this->getGroupInfo($name);

}

public function diePermission($name)

{

if (!$this->checkPermission($name)) die;

}

public function getGroupInfo($name='')

{

$id = $this->getInfo('group\_id');

if (empty($this->\_group\_data)) $this->\_group\_data = mysql\_fetch\_assoc(sql("SELECT \* FROM groups WHERE id = $id"));

if (empty($name)) return $this->\_group\_data;

else return $this->\_group\_data[$name];

}

public function changeGroup($group\_id)

{

sql("UPDATE users SET group\_id = $group\_id WHERE id = {$this->\_id}");

$this->\_data = '';

$this->\_group\_data='';

}

**4.2 Класс управления авторизацией AuthManager**

Данный класс предназначен для авторизации пользователя и получения объекта класса User. Для создания экземпляра класса не требуются входные параметры:

public function \_\_construct() {}

Для авторизации пользователя используются следующие функции:

checkArrays() – проверка суперглобальных массивов.

checkPost() – проверка суперглобального массива $\_POST.

checkSession() – проверка суперглобального массива $\_SESSION.

checkCookie() – проверка суперглобального массива $\_COOKIE.

authorise() – авторизация пользователя и отображение формы авторизации в случае провала.

userAuth() – регистрация пользователя в системе.

public function authorise()

{

session\_start();

$this->checkPost();

$this->checkArrays();

if ($this->\_user == null) $this->showAuthForm();

}

private function checkArrays()

{

if ( !isset($\_SESSION['password']) && !isset($\_COOKIE['password']) )

{

$this->showAuthForm();

}

if (isset($\_SESSION['password'])) $this->checkSession();

if (isset($\_COOKIE['password'])) $this->checkCookie();

}

private function checkPost()

{

if (isset($\_POST['password']))

{

$name = $\_POST['name'];

$password = $\_POST['password'];

$password = md5($name.$password);

if ($this->isUserExist($password))

{

$this->userAuth($password);

header('Refresh: 2; URL=http://'.$\_SERVER['SERVER\_NAME'].$\_SERVER['REQUEST\_URI']);

die('Пароль верный.');

}

else

{

header('Refresh: 2; URL='.$\_SERVER['REQUEST\_URI']);

die('Неверный пароль.');

}

}

}

private function checkSession()

{

$password = $\_SESSION['password'];

if ($this->isUserExist($password)) $this->userAuth($password);

}

private function checkCookie()

{

$password = $\_COOKIE['password'];

if ($this->isUserExist($password)) $this->userAuth($password);

}

private function showAuthForm()

{

include($\_SERVER['DOCUMENT\_ROOT']."/inc/login\_form.php");

die();

}

private function userAuth($password)

{

$\_SESSION['password'] = $password;

if (isset($\_POST['remember'])) setCookie('password', $password, time() + 3000000); // на месяц

$res = mysql\_fetch\_assoc( sql("SELECT \* FROM users WHERE password='$password'"));

$this->\_user = new User($res['id']);

}

Вспомогательные функции:

getUser() – возвращает объект класса User в случае успешной авторизации.

isUserExist() – проверка существования пользователя в базе данных.

private function isUserExist($password)

{

$query = sql("SELECT \* FROM users WHERE password='$password'");

$len = mysql\_num\_rows($query);

return ($len == 1);

}

public function getUser()

{

return $this->\_user;

}

**5. Ввод и обработка данных**

Ввод данных будет осуществляться с помощью web-формы. Введенные данные будут переданы скрипту обработки для занесения данных в базу данных и формирования отчета в формате RTF.

**5.1 Формат RTF**

Данный формат был определен фирмой Microsoft как стандартный формат для обмена текстовыми документами. RTF поддерживается многими продуктами фирмы Microsoft. Так, например, начиная с версии 2.0, он введен в Windows в качестве Clipboard-формата, благодаря чему возможен обмен данными между различными прикладными программами Windows. Кроме того, RTF-формат поддерживается программами WORD для Macintosh, начиная с версии 3.X, и WORD для PC, начиная с версии 4. X.

В RTF для обмена документами используются только представимые символами коды из ASCII-, MAC- и PC-символьного набора. Кроме текста, файл в RT-формате в читаемой форме содержит команды управления.

Документ состоит преимущественно из команд управления настройки программы чтения файлов в RTF-формате. Эти команды можно разделить на управляющие слова (control words) и управляющие символы (control symbols).

Управляющее слово представляет собой последовательность символов с разделителем (delimiter) в конце:

\lettersequence <delimiter>

Перед управляющим словом вводится обратная косая черта "\" (backslash). В качестве разделяющих могут использоваться следующие символы:

* Пробел (space), причем этот символ относится к управляющему слову;
* Цифра или знак "-". После этих символов должен следовать параметр с разделителем. В качестве разделителя может быть использован пробел или другие символы, кроме цифр и букв;
* Все символы, отличные от цифр и букв. Эти символы не относятся к управляющему слову.

В RTF-формате для задания управляющей последовательности используются буквы от "А" до "Z" и от "а" до "z", а также цифры от "0" до "9". Национальные символы к управляющей информации не относятся.

**5.2 Обработка введенных данных**

После заполнения web-формы данные передаются в скрипт-обработчик make\_rtf.php. Для формирования отчета используются rtf-шаблоны, в которых проставлены метки в необходимых местах, для того чтобы заменить эти метки на введенные данные. Метка имеет вид AAAимя\_меткиAAA. Все данные, полученные от web-формы, содержаться в массиве $\_POST['data']. Для перекодирования текста в формат RTF используется функция vutf():

// Функция преобразует строковый параметр в строку юникод символов параметра для сохранения в rtf

function vutf($str\_par)

{

$str\_par = urlencode($str\_par);

$count = substr\_count($str\_par,'%');

for ( $i=0; $i < $count; $i++)

{

$str\_par = substr\_replace($str\_par, '\\\'', strpos($str\_par, '%'),1);

}

//$str\_par = strtolower($str\_par);

$str\_par = strtr($str\_par, '+', ' ');

return $str\_par;

}

Процесс замены меток выглядит следующим образом:

$data = $\_POST['data'];

foreach ($data as $k => $v)

{

$file = str\_replace('AAA'.$k.'AAA', vutf(stripcslashes($v)), $file);

}

После обработки файл сохраняется на диск и выводится в браузер пользователя.

header('Content-Disposition: attachment; filename="'.$filename.'.rtf"');

header('Content-type: application/msword');

header("Expires: 0");

header("Cache-Control: must-revalidate, post-check=0, pre-check=0");

header("Pragma: public");

echo $file;

**Заключение**

В настоящее время информационная система для сбора и обработки данных на ООО «МАШПРОМЭКСПЕРТ» находится в разработке и не выполняет полный перечень необходимых функций. Несмотря на это она значительно упростила процесс формирования отчетов для работников ООО «МАШПРОМЭКСПЕРТ». В данной работе были рассмотрены функции авторизации пользователей, сбора информации и ее обработка для формирования отчета.

Система постоянно развивается, в разработке находится модуль истории и отчеты о производительности работников.

# Список используемой литературы

1. Котеров Д. В., Костарев А. Ф., " PHP 5 В Подлиннике", 2006 г.

2. Харрис Э., "PHP/MySQL для начинающих", 2005 г.

3. Грекул В.И. , Денищенко Г.Н. , Коровкина Н.Л., "Проектирование информационных систем", 2008 г.

4. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес, "Приемы ООП. Паттерны проектирования", 2005 г.

5. Лешек А. Мацяшек, "Анализ и проектирование информационных систем", 2006 г.