**3.8 ConnectDB design**

**3.8.1 Module description**

This module is about the connection to the database. All operations to the database need this module, and other modules just invoking the functions in this module.

**3.8.2 Profile**

The name of profile is ***ConnectConfig.php***, and these are what in the file:

*<?*

*$database=Course\_Resource*

*$acount=root*

*$password=root*

*$server=localhost*

*$port=3306*

*?>*

**3.8.3 Performance**

In each module, the connection to the database is need, and each operation related to the database is finished in this module, including connecting, disconnecting, querying, and even changing data.

**3.8.4 The definition of ConnectDB**

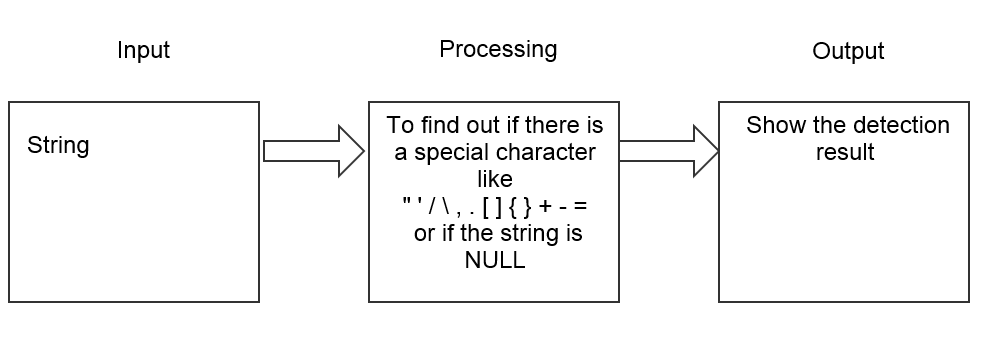
|  |  |
| --- | --- |
| **Function Name** | **Function** |
| open() | Use the function mysql\_connect() inside the php to create a connection to the database. If succeed, return true, else return false. |
| close() | Use the function mysql\_close() inside the php to close the connection to the database. If succeed, return true, else return false. |
| query  (sql) | Use the function mysql\_query(sql) inside the php to make a SQL query, and the SQL statement is in the constant string ‘sql’. If succeed, return the query result, else return an error. |
| fetch\_assoc() | Use the function mysql\_fetch\_assoc() inside the php to get an associative array from the result. If succeed, return the associative array, else return an error. |
| LoginQuery  (user\_id, password) | Make a SQL query to find out if there is a user whose id is ‘user\_id’ with a password ‘password’. If there is such a user, store the login status in the session and return true, else return false. |
| SecurityCheck() | Verify the user’s information. If the user passes the verification, return true, else return false. |
| getUsertype() | Get the type of the user, and return the result with a string. |
| AddAssignment  (form) | All information of assignment is in the form. Add the assignment to the database, if succeed, return true, else return false. |
| generateTmpName  (filename) | Get a new name for the file ‘filename’, and return the new name. |
| getFromDB  (fileID) | Query the information of the file with an ID ‘fileID’, and return all the information. |
| AddFileInfoToDB  (filename,filetype,filesize,  tmpname,course\_id) | Add the information of the file to the database. The original name of the file is ‘filename’, the type of the file is ‘filetype’, the size of the file is ‘filesize’, the name in the database of the file is ‘tmpname’, and the ID of the course the file related to is ‘course\_id’. |

**3.9** **Data security detection module**

**3.9.1 Module description**

This module is used to detect the data via the client and make sure the data security. Some users may user SQL injection or some other methods to get some data or change the data in the database illegally. To avoid this condition, security detection is needed.

**3.9.2 Function**

****

The IPO graph for module 9

**3.9.3 Performance**

After getting a string, to make sure that the string is legal, make a detection on the string and give out the result.

**3.9.4 Input**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Signal** | **Type and Format** | **Input Format** |
| String to be detected | StrCheck | string | Function parameter |

**3.9.5 Output**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Signal** | **Type and Format** | **Output Format** |
| Detection result | CheckResult | string | Function return value |

**3.9.6 Algorithm**

<?

function bool CheckStringSafety(string StrCheck)

{

char example[]={“,’,/,\,,,.,[,],=,-,+};

if (none of example in StrCheck)

return true;

else

return false;

}

?>

**3.9.7 Test plan**

Test with other modules which need detection.

|  |  |
| --- | --- |
| **Input data** | **Expected Result** |
| NULL | Return warning |
| A string with illegal character | Return warning |
| A legal string | Operate succeed and return result |

**6. Running design**

**6.1 Module combination**

In this subsystem, the modules are divided by their different functions. Each module contains client interface, client script and background program in the server. Different modules don’t share an interface except for the main interface, and different client scripts just share a common class XMLHttpRequest. The background programs have connection by the database connection.

**6.2 Running control**

The user choose different function in the main interface:

**Login Module**:

The user inputs the User\_ID and Password to log in, and then other functions are allowed.

**Personal Resource Management Module:**

After logging in, the user are granted to do some operations. In this module, the user can view, download and change the resources online.

**Resource Upload Module:**

After logging in, the user are granted to do some operations. In this module, the user can upload a file as a resource to the server.

**Search Module:**

After logging in, the user are granted to do some operations. In this module, the user can search and download a file from all the homeworks and resources.

**Homework Assign Module:**

After logging in, the user are granted to do some operations. In this module, a instructor can assign homework for the courses, while a student can do nothing.

**Homework Upload Module:**

After logging in, the user are granted to do some operations. In this module, students can view their courses and homework and select a homework to upload.

**Homework Check Module:**

After logging in, the user are granted to do some operations. In this module, a instructor can check their students’ homework and download it.

**6.3 Running time**

With the using of AJAX, we can use static update technique in the client, and full page operations can also be reduced, so we can achieve a high server bandwidth utilization. In this way, the running time is greatly reduced.

One thing influence the running time a lot is the connection and disconnection to the database. Each operation needs to link the database will touch off these two steps, and these operations can be found in every module.

What’s more, when doing querying operations, for example, when looking up the user’s information in the database, if the table is very big or the input is wrong, the program has to traverse the whole table and cost a lot of time.

**7. System Error Handling**

**7.1 Errors**

|  |  |  |
| --- | --- | --- |
| **Type of Error** | **The Reason** | **Handling** |
| Unable to connect  the database | The profile is not right and cannot connect the database, or too much users access to the database at the same time. | Correct the profile to make sure the connection. Make a limit on the interval between two requirements of the same user. |
| SQL error | Some users insert SQL statement in the form and try to destroy the database or change the data illegally, which may lead to error in the database. | Detect the data in the form in the server first, and drop the bad request. |
| Information missing on the website | Some users try to get access to the website even the background without logging in, and destroy the website or get some information illegally. | Some websites should be banned to access directly. |
| The server crashed | The server works for a long time with high work load, and meet a lot of bugs in the running, finally the server crashed. | Have regular server maintenance. |
| Account Information leaked | Computer viruses in the client or other reasons lead to account information leaked, and the password changed. Or hacker attack the server and get everything in the database. | Use antivirus software to protect the computer, and The users should never log in with other people’s computer. And the passwords in the database should be encrypted by algorithms like MD5. |
| The hard disk crashed | Reading and writing too often and too many operations to the database lead to the damage of the hard disk and the lost of data. | Backup regularly. |
| Illegal instructions cannot be handled. | No resource exist or other bad requests. | Check and make sure the instruction is right before executing the statement. |
| Get random or mess code from the database | Having reading or writing errors when having database operation. | Use the same code or encode different kind of code. |

**7.2 Remedy**

**7.2.1 Backup system technology**

Backup the database regularly to avoid losing.

Set different database in different computer, so that the data won’t lost at the same time.

Save the same data at more than one computer, so that we can recover the data from other computers.

**7.2.2 Use a less effective technology**

When the client does not work, we can make operations with manual work. For example, we can download the resource directly from the server, or change the resource information manually.

**7.2.3 Recover and reboot technology**

If the server or the database crashed, we can rerun it from the breaking point or just start it at the very beginning.

**7.3 System maintenance design**

In the database operation, when connecting or disconnecting the database or making an SQL query, catch every exception and get the warning.

Check the IP accessed to the system, don’t let a IP visit the website too often to avoid attack like DDOS.

Have a record of administers’ operation to the server and database, so that dangerous operations can be checked out very soon.