Proyecto Fin de Carrera

“Forum Integration for Moodle”

AUTOR: Antonio Javier Martínez Vaillo.
DIRECTOR: José Juan Sánchez Manzanares.
CODIRECTOR: Bernhard Fischer.

Septiembre 2012
## Resumen

Los foros de Moodle no integran los correos electrónicos de forma bidireccional. Es decir, las respuestas desde una cuenta de correo electrónico hacia un mensaje de un usuario del foro no aparecen en la plataforma web. Para mejorar la usabilidad de estos foros, re-implementaremos el foro de Moodle de una forma apropiada.

Se plantearon varios mecanismos para resolver esta cuestión, como es el caso de de Mailman, una aplicación de software del proyecto GNU, que maneja listas electrotécnicas de correo o simplemente listas de correo.

Sin embargo, una vez centrados en las opciones de configuración de Moodle, averiguamos que el mantenimiento y control de los participantes en los foros es llevado a cabo de una forma efectiva a través de la base de datos de Moodle.

Haciendo uso de dichas opciones e implementando un proceso en php, llegamos a la solución del problema planteados.

<table>
<thead>
<tr>
<th>Autor</th>
<th>Antonio Javier Martínez Vaillo</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail del autor</td>
<td><a href="mailto:antoniojvaillo@hotmail.com">antoniojvaillo@hotmail.com</a></td>
</tr>
<tr>
<td>Director</td>
<td>José Juan Sánchez Manzanares</td>
</tr>
<tr>
<td>E-mail del director</td>
<td><a href="mailto:pepe.manzanares@si.upct.es">pepe.manzanares@si.upct.es</a></td>
</tr>
<tr>
<td>Codirector</td>
<td>Bernhard Fischer</td>
</tr>
<tr>
<td>Título del PFC</td>
<td>Forum Integration for Moodle</td>
</tr>
<tr>
<td>Titulación</td>
<td>Ingeniería de Telecomunicación</td>
</tr>
<tr>
<td>Departamento</td>
<td>Tecnologías de la Información y las Comunicaciones</td>
</tr>
<tr>
<td>Fecha de presentación</td>
<td>Septiembre 2012</td>
</tr>
</tbody>
</table>
## Contents

1. Introduction ............................................................................................................ 3
2. Moodle ..................................................................................................................... 3
3. eCampus in the FH................................................................................................. 5
4. Framework.............................................................................................................. 6
   4.1. Forums in eCampus .......................................................................................... 6
   4.2. The modified forum module .......................................................................... 8
   4.3. Technologies used .......................................................................................... 8
   4.4. Moodle database ............................................................................................ 8
5. Modification of the forum module ........................................................................ 9
   5.1. Where can we change the “To” in the reply of the users? ............................ 9
   5.2. The support email ........................................................................................... 11
   5.3. How does Moodle send the emails? ............................................................... 12
   5.4. Where do we insert the posts? ........................................................................ 12
       - Created and modified ..................................................................................... 12
       - Message and subject ..................................................................................... 13
       - Userid ............................................................................................................. 13
       - Discussion and parent ................................................................................... 13
   5.5. Reading emails ............................................................................................... 14
   5.6. Processing emails ........................................................................................... 15
   5.7. Check user ...................................................................................................... 16
   5.8. Insert post ....................................................................................................... 17
6. Operating example ............................................................................................... 18
7. Appendix I. Process.php ...................................................................................... 26

Bibliography.................................................................................................................. 29
List of Figures

Figure 1: Sites Moodle ................................................................. 4
Figure 2: eCampus FH ................................................................. 5
Figure 3: Forums in eCampus 1 .................................................... 6
Figure 4: Forums in eCampus 2 .................................................... 7
Figure 5: The modified forum module ......................................... 8
Figure 6: Forum's tables ......................................................... 8
Figure 7: Forum configuration .................................................. 10
Figure 8: Email configuration .................................................. 11
Figure 9: Table "mdl_forum_posts" ........................................... 12
Figure 10: Moodle and course of test ........................................ 18
Figure 11: Participants ............................................................. 19
Figure 12: New discussion ...................................................... 20
Figure 13: Posts from Moodle .................................................. 21
Figure 14: Email user account .................................................. 23
Figure 15: Email administrator account .................................... 23
Figure 16: Posts with the process.php ....................................... 25
Abstract

Moodle web forums do not integrate email bidirectional. This is, email responses to forum posts do not appear on the web. To enhance usability, we reimplement the Moodle forum in the appropriate way.

When it’s has to solve a problem like this, it raises various possible solutions. One of these likely solutions was through the use of Mailman, a computer software application from the GNU project for managing electronic mailing lists.

Mailman is free software for managing electronic mail discussion and e-newsletter lists. It runs on GNU/Linux and most Unix-like systems, and requires Python 2.1.3 or newer. GNU Mailman works with Unix style mail servers such as Postfix, Sendmail and qmail.

Features include:

- A Web browser interface for list administration, archiving of messages, spam filtering.
- A customizable home page for each mailing list.
- Integrated bounce detection and automatic handling of bouncing addresses.
- Integrated spam filters
- Majordomo-style email based commands.
- Multiple list owners and moderators.
- Per-list privacy features, such as closed-subscriptions, private archives, private membership rosters, and sender-based posting rules.
- Support for virtual domains.
- Web based subscribing and unsubscribing. Users can temporarily disable their accounts, select digest modes, hide their email addresses from other members, etc.

This option was considered at the beginning because it was necessary for a good management of the participants list in the forums and this we would give a good integration with Moodle.

Mailman allows us a flexible and quick reference to the list of participants and the emails sent to them.

However, once we were focused in the configuration options of Moodle, we could know that the management of the participants in the forums is carry out with an effective way with the database of Moodle. The management and storage of users and forums will be explained in various sections of this report, as in the case of the points 4.4 and 6.
Another question that we had to decide what solutions take is about to send the emails. Moodle, through the Server configuration, it allow us configure our SMTP Server, in an easy way. Section 5.2 of this document.

As well, the sending of the emails to the all users in the forums is automatic when we execute the cron.php. Section 5.3.

For these reasons, we have decided for the no integration of the Moodle application with the software Mailman. Although, I think that we can propose a “future line of investigation” that try to find a good integration of both systems.
1. Introduction

The development of Internet has produced a revolution in the distance education at all levels. It appears the virtual learning teaching or e-learning like a complement and sometimes like a replacement of the education classroom. We can define the virtual learning like the education and the knowledge acquisition and skills through the use technologies based on Internet.

A definition of virtual learning would be the next:

“Distance Education (DE) is a system of education characterized by physical separation between the teacher and the learner in which instruction is delivered through a variety of media including print and other ICTs to learner who may either have missed the opportunity earlier in life or have been denied the face-to-face formal education due to socio-economic, career, family and other circumstances. Jegede, (2003) defined distance education as education provided by a mode other than the conventional face-to-face method whose goals are similar to and just as noble and practical as those of on-campus full time face-to-face education.”

It is a new concept educative that join the use of technology and learning material to achieve the design and evolution of the distance courses. It has to be understood like the result of put in practice the news technologies to the education and learning, tore out his best advantage.

2. Moodle

Moodle is a software package for producing Internet-based courses and web sites. It is a global development project designed to support a social constructionist framework of education.

Moodle is provided freely as Open Source software (under the GNU Public License). Basically this means Moodle is copyrighted, but that you have additional freedoms. You are allowed to copy, use and modify Moodle provided that you agree to: provide the source to others; not modify or remove the original license and copyrights, and apply this same license to any derivative work.

Moodle can be installed on any computer that can run PHP, and can support an SQL type database (for example MySQL). It can be run on Windows and Mac operating systems and many flavors of Linux (for example Red Hat or Debian GNU). There are many knowledgeable Moodle Partners to assist you, even host your Moodle site.

The word Moodle was originally an acronym for Modular Object-Oriented Dynamic Learning Environment, which is mostly useful to programmers and education theorists. It's also a verb that describes the process of lazily meandering through something, doing things as it occurs to you to do them, an enjoyable tinkering that often leads to insight and creativity. As such it applies
both to the way Moodle was developed, and to the way a student or teacher might approach studying or teaching an online course. Anyone who uses Moodle is a Moodler.

The Moodle learning environment is based on constructivist pedagogical principles, with a modular design that makes it easy to add content to motivate the student.

The activities are the heart of course management system. Moodle was designed by an educator and computer, based on the principles of "social constructivism". Constructionism asserts that learning is particularly effective when done by sharing with others. This experience can be anything from a spoken sentence or an internet posting, or elements more complex like a painting, a house or a software application.

Nowadays there are 49547 currently active sites that have registered from 213 countries.

Moodle is a virtual learning environment ideal for schools, colleges, universities, businesses, schools, hospitals...

Moodle has a lot of properties that made it like a perfect e-Learning platform. Some of these properties are educational adequacy, adapting teaching, simple installation and maintenance, large support, open source software...
3. eCampus in the FH

The University of Applied Sciences in St. Pölten chose Moodle as their e-Learning Platform. The appearance of this platform in the FH of St. Pölten is:

![Figure 2: eCampus FH](image)

It’s been a great surprise know that in the FH St. Pölten all community of the university use a lot eCampus. For example, in my university, in Cartagena, use Moodle since September of 2008, and I remember that at the beginning the lectures didn’t use so much this platform. Almost of them continued using their own servers to upload the information and the material for the lessons. That was awful for the students because you had to know three or four internet address to get all the things for your subjects. Now it’s better because the lectures use Moodle and they’ve understood that this is the best way for the all university community.

The appearance of FH eCampus made it a interface very easy to use and all university community use this platform no just for learning, you can check a lot information in this eCampus, like sports, events, look for a jobs,…
4. Framework

4.1. Forums in eCampus

Moodle has four kinds of forums each with a slightly different layout and purpose.

The most usual is the standard forum for general use. In this forum type, participants will see the introduction text in a separate space above the discussion field, in which you will see the information such as the title of the discussion (which means the forum’s title), its author, the number of replies and the date of the last post.

When the user writes a message in a Forum, the message is written in the database and sent to the email of the others participants in the forum.

If a person wants reply any post has to introduce in eCampus and reply the post from eCampus.
Below there are some properties of the forum module:

- Different types of forums are available, such as teacher-only, course news, open-to-all, and one thread per user.
- There are several options for emailing forum posts to members of the course.
- All postings have the author’s photo attached.
- Discussion can be viewed nested, flat or threaded, oldest or newest first.
- Robust subscription methods for each forum
  - Individual forums can be subscribed to by each person.
  - Teacher can force subscription for all members of the course, either initially or permanently.
- Group’s features allow options for more entry and viewing limitations for students.
- The teacher can choose not to allow replies to their posts.
- Discussion threads can be moved between forums or split by the teacher.
- Attachments can be made to posts and shown as part of message.
- Forum ratings can be used; these can be restricted to a range of dates.
4.2. The modified forum module

We want that the user don’t need access to Moodle to reply the entry on the Forums. The users will be able to send the reply from their personal email.

![Diagram](image.jpg)

Figure 5: The modified forum module

4.3. Technologies used

- Moodle 1.9.8.
- MySQL.
- PHP.
- Ubuntu.
- VirtualBox.

4.4. Moodle database

Moodle uses a database in MySQL. Moodle has tables for each activity. For example here are the tables used for the forum module activity.

![Tables](image2.jpg)

Figure 6: Forum's tables
5. Modification of the forum module

When somebody have to modify something like that, you have to think in every step that you have to do. Then, I will explain all the steps that I’ve followed.

5.1. Where can we change the “To” in the reply of the users?

According with the properties of the forum activity in Moodle, when one user tries to answer an email received from the forum, the receiver of this email is the user who wrote in the forum. We think that the emails should be sent to an administrator account, because we need process every email, and if the emails are sent to the user account we won’t be able to get these emails. Our process will access to this administrator account and it will process the received emails.

This has been configured changing the property Modules→Activities→Forum→forum_replytouser, that the default value is “Yes”. Unchecking this property, emails will be replied to the administrator account instead to the user account.
Forum configuration

Figure 7: Forum configuration
5.2. The support email

On the other hand, we have to configure the properties for Moodle can send the emails. That is configured in Server\rightarrow Email.

![Email configuration](image)

Figure 8: Email configuration
5.3. How does Moodle send the emails?

Next question: How does Moodle send the emails to the users? We don’t have to worry about this question because Moodle uses a cron.php. Cron is a program that runs predefined task at regular intervals. When the cron.php is run one of the task that do is send the email from the forum module to the participants of the forum. We can find this file in /opt/lamp/htdocs/moodle/admin/cron.php or we can run it directly from the browser: http://example.com/moodle/admin/cron.php.

5.4. Where do we insert the posts?

We must know in what table of the Moodle database the posts written for the users in the forum are inserted, because we will use this table in our process to insert the reply from the email account of the users. The table used for this issue is “mdl_forum_post” that has the following fields.

![Figure 9: Table "mdl_forum_posts"

The fields that we have to insert with our program are, “discussion”, “parent”, “userid”, “created”, “modified”, “subject” and “message”.

Created and modified

The fields “created” and “modified”, we will insert them using the function UNIX_TIMESTAMP(), that returns a Unix timestamp (seconds since ‘1970-01-01 00:00:00’ UTC) as an unsigned integer.

```
$sqlinsert= "INSERT into mdl_forum_posts
discussion,parent,userid,created,modified,subject,message)"
values('$discussion','$parent','$userid',UNIX_TIMESTAMP(),UNIX_TIMESTAMP(),'$subject','$message')";
```
Message and subject

About the fields “message” and “subject” we will get them using the IMAP functions for PHP.

```php
$message = imap_fetchbody($mbox,$i,'1');

$Subject_insert = substr($header->Subject,0,$pos_first);
```

Userid

To get “userid” we’ll do a SQL query using the user email which has sent the mail.

```php
$query = sprintf("SELECT * FROM mdl_user WHERE email = '$mail'");
$result = mysql_query($query);
if(!$result){
    die('Invalid query: ' . mysql_error());
}

$num_rows = mysql_num_rows($result);
$row = mysql_fetch_assoc($result);
return $row['id']; //userid
```

Discussion and parent

For the fields “discussion” and “parent”, we have modified the file that compose the email for the forum Moodle. We have inserted these fields in the Subject of the email. Our program will get these fields examining the Subject of the received emails.

The file that we have to modified is /moodle/mod/forum/lib.php. I've concatenated these fields in the email Subject:

```php
$postsubject = "$course->shortname: ".
    format_string($post->subject,true).
    "?id=".$post->id .
    "?dis=" .$post->discussion .
    "?pa=" .$post->parent .
    "?usu=" .$post->userid;
```

With the process.php, we will get the fields processing the Subject:

```php
$values = strstr($header->Subject,'?');
$valuesarray = str_split($values);
$eq=0;
$que=0;
```
$valin = "";
$arrayeq = array();
$arrayque = array();
foreach ($valuesarray as $index => $value) {
    if ($value == "=") {
        $arrayeq[$eq] = $index;
        ++$eq;
    }
    if ($value == "?") {
        $arrayque[$que] = $index;
        ++$que;
    }
}

// Get discussion
$dis = substr($values, $arrayeq[1]+1, $arrayque[2]-$arrayeq[1]-1);

// Get parent
$parent = substr($values, $arrayeq[2]+1, $arrayque[3]-$arrayeq[2]-1);

5.5. Reading emails

Now, we focus in the issue of read the emails. For that, we'll use the IMAP PHP library. Through the next code lines we can connect to the server and read the server's directories:

$host = '{sxstudents.fhstp.ac.at}INBOX/Moodle';
$user = 'is090002';
$pass = '************';
echo $host;
$mbox = imap_open($host, $user, $pass) or die("can't connect: " . imap_last_error());

$list = imap_getmailboxes($mbox, $host, "*");
if (is_array($list)) {
    foreach ($list as $key => $val) {
        echo "($key) ";
        echo "Folder: ";
        echo imap_utf7_decode($val->name) . ", ";
        echo "$\"" . $val->delimiter . ", ";
        echo "$\"" . $val->attributes . "<br />
        }
    } else {
        echo "imap_getmailboxes failed: ". imap_last_error(). "\n";
    }
}
5.6. Processing emails

After that, we have to examine the emails for get of the necessary fields, that it's been explained in the point 4 of this section. This fields are necessary for insert the new post in the table “mdl_forum_posts”. We do that using a “for”, that it is executed as many times as the number of messages in the server, obtaining the information of everyone of the received emails:

```php
$message_count = imap_num_msg($mbox);
for ($i = 1; $i <= $message_count; ++$i) {
    $header = imap_header($mbox, $i);
    echo "<br />
";
    echo "USER: ";
    echo $header->fromaddress . "<br />
";
    echo "SUBJECT: ";
    echo $header->Subject . "<br />
";
    echo "IMAP_HEADER: ";
    print_r($header->from);
    $mailbox = $header->from[0]->mailbox;
    $host = $header->from[0]->host;
    echo "<br />
";
    $fromfinal = $mailbox . "@" . $host;
    echo "EMAIL USER: ";
    echo $fromfinal . "<br />
";
    $compruebausu = check_user("$fromfinal");
    echo "<br />
";
    echo "Checking User....." . "<br />
";
    echo "User Id: " . $compruebausu . "<br />
";

    if($compruebausu){
        echo "The user exists";
        echo "<br />
";
    }
    if(!$compruebausu){
        echo "The user doesn't exit";
        echo "<br />
";
    }
    $pos_first = strpos($header->Subject,'?');
    $Subject_insert = substr($header->Subject,0,$pos_first);
    $values = strstr($header->Subject,'?');
    $valuesarray = str_split($values);
    $eq=0;
    $que=0;
    $valin = "";
    $arrayeq = array();
    $arrayque = array();
    foreach ($valuesarray as $index => $value){
        if($value == "="){
            $arrayeq[$eq] = $index;
            ++$eq;
        }
        if($value == "?"){
            $arrayque[$que] = $index;
            ++$que;
        }
    }
```

5.7. Check user

As well, we have created two functions. The first one is the function that check if the user is in Moodle, and also returns us the user id, that we need to insert the register in "mdl_forum_posts":

```php
function check_user($mail){
    $link = mysql_connect('localhost', 'root', '');
}
```
### 5.8. Insert post

The other function is where it's run the insert of the new record:

```php
function insert_post($discussion, $parent, $usu, $userid, $subject, $message)
{
    echo "Insert.....";
    $link = mysql_connect('localhost', 'root', '');
    if(!$link){
        die('Could not connect: ' . mysql_error());
    }
    $db_selected = mysql_select_db('moodle', $link);
    if(!$db_selected){
        die('Can\'t use moodle: ' . mysql_error());
    }
    ++$parent;
    $sqlinsert= "INSERT into mdl_forum_posts(discussion,parent,userid,created,modified,subject,message)
values(\'$discussion\',\'$parent\',\'$userid\',UNIX_TIMESTAMP(),UNIX_TIMESTAMP(),\'$subject\',\'$message\'\')";
    mysql_query($sqlinsert,$link);
    echo "<br />
";
    echo "INSERTED ROW";
    echo "<br />
";
}
```
6. Operating example

Now we come to explain the Moodle that we have used to do the modification of the forum module. In this point I will explain the configuration of the course with some pictures, as well the adding of the forum and discussion that we’ll use to test the program that we have done.

In the first illustration we can see the courses that there are in our Moodle. We can look too, the homepage of the Bakkalaureatsarbeit, in which there are two forum. We will use the forum “Forum for the Bakkalaureatsarbeit”. In the other part of the image is viewed the insert of the forum in the table “mdl_forum”.

![Moodle homepage](image1)

![Course Bakkalaureatsarbeit](image2)

![Table “mdl_forum”](image3)

Figure 10: Moodle and course of test
Another important element in a Moodle course, are the users. In the next image we can look the participants in the Bakkalaureatsarbeit course, and these users will participate in the “Forum for the Bakkalaureatsarbeit” for this operating example. Antonio Javier Martínez Vaillo is the “professor” of the course; Maria José Martínez Belmonte y Thanh Nguyen Huy, are “students”.

**Bakkalaureatsarbeit**

![Participants](image)

Table “mdl_user”

**Figure 11: Participants**
Then we can view how is adding a new discussion in the forum:

![Adding new discussion](image)

The next step that we have to do is write some posts in the forum for look where are inserted in the database, an important point to know because in this table is where we will insert the received replies in our email account.

![New discussion](image)

"Table mdl_forum_discussions"

Figure 12: New discussion
Looking the table “mdl_forum_posts” the most remarkable thing for our aim, is the field “mailed”, whose default value is inserted with “0”. This values means that the posts has not been send yet by email to the forum’s users. To send the emails we have to run the cron.php and after that, the field “mailed” will have the value “1”. The result showed in the browser when we run the cron is:

```
Server Time: Sun, 23 May 2010 14:38:26 +0200

Starting activity modules
Processing module function assignment_cron ...done.
Processing module function chat_cron ...done.
Processing module function forum_cron ...Processing user 7
Sending post 41: New forum integration for Moodle
Sending post 42: Re: New forum integration for Moodle
Sending post 43: Re: New forum integration for Moodle
Sending post 44: Re: New forum integration for Moodle
Sending post 45: Re: New forum integration for Moodle
Processing user 4
Sending post 41: New forum integration for Moodle
Sending post 42: Re: New forum integration for Moodle
Sending post 43: Re: New forum integration for Moodle
Sending post 44: Re: New forum integration for Moodle
Sending post 45: Re: New forum integration for Moodle
```
After running the cron.php, the users will receive in their email accounts, the emails. From their email accounts they will be able to reply the post that they want. In this case, we’ll show the email account of María José Martínez Belmonte. It’s possible to see the received emails, an email, and the reply of this email:
As Maria José, the user Thanh will send his answer to the administrator account. In the administrator account we can look at these emails:
After receiving emails from users, we run the process created. In the browser we can see the next result:

sxstudents.fhstp.ac.at}INBOX/Moodle (0)
Folder:{sxstudents.fhstp.ac.at}INBOX/Moodle,'/',0

USER: Nguyen-Huy Thanh
IMAP_HEADER: Array ( [0] => stdClass Object ( [personal] => Nguyen-Huy Thanh [mailbox] => is090004 [host] => fhstp.ac.at ) )
EMAIL USER: is090004@fhstp.ac.at

Checking User.....
User Id: 8
The user exists
Id: 44
Discussion: 5
Parent: 42
Message: thank you!
Insert.....
INSERTED ROW

USER: Martinez-Belmonte Maria-Jose
IMAP_HEADER: Array ( [0] => stdClass Object ( [personal] => Martinez-Belmonte Maria-Jose [mailbox] => is090001 [host] => fhstp.ac.at ) )
EMAIL USER: is090001@fhstp.ac.at

Checking User.....
User Id: 4
The user exists
Id: 45
Discussion: 5
Parent: 43
Message: Thank you for the answer.
Insert.....
INSERTED ROW
The result in the forum is the adding of two new posts. Also, in this image we can see these posts in the database:

Posts inserted whit the process.php

<table>
<thead>
<tr>
<th>id</th>
<th>discussion</th>
<th>parent</th>
<th>userid</th>
<th>created</th>
<th>modified</th>
<th>mailed</th>
<th>subject</th>
<th>message</th>
<th>format</th>
<th>attachment</th>
<th>totalscore</th>
<th>mainlor</th>
</tr>
</thead>
<tbody>
<tr>
<td>68</td>
<td>X</td>
<td>44</td>
<td>8</td>
<td>1274642267</td>
<td>1274642267</td>
<td>0</td>
<td>RE: BAKK2: Re: New forum integration for Moodle</td>
<td>thank you!</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>X</td>
<td>45</td>
<td>4</td>
<td>1274642268</td>
<td>1274642268</td>
<td>0</td>
<td>RE: BAKK2: Re: New forum integration for Moodle</td>
<td>Thank you for the answer.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table “mld_forum_posts”

Figure 16: Posts with the process.php
7. Appendix I. Process.php

```php
<?php
$host = '{sxstudents.fhstp.ac.at}INBOX/Moodle';
$user = 'is090002';
$pass = '************';

$host;
$mbox = imap_open($host, $user, $pass) or die("can't connect: ", imap_last_error());

$list = imap_getmailboxes($mbox, $host, "*"); if (is_array($list)) {
    foreach ($list as $key => $val) {
        echo "($key) ";
        echo "Folder:";
        echo imap_utf7_decode($val->name). ",";
        echo "" . $val->delimiter . ",";
        echo $val->attributes."\n";
    }
} else {
    echo "imap_getmailboxes failed: " . imap_last_error() . "\n";
}

$message_count = imap_num_msg($mbox);

for ($i = 1; $i <= $message_count; ++$i) {
    $header = imap_header($mbox, $i);
    echo "<br />\n";
    echo "USER: ";
    echo $header->fromaddress . "\n";
    echo "SUBJECT: ";
    echo $header->Subject . "\n";
    echo "IMAP_HEADER: ";
    print_r($header->from);
    $mailbox = $header->from[0]->mailbox;
    $host = $header->from[0]->host;
    echo "<br />\n";
    $fromfinal = $mailbox . "@" . $host;
    echo "EMAIL USER: ";
    echo $fromfinal . "\n";
    $compruebausu = check_user("$fromfinal");
    echo "<br />\n";
    echo "Checking User....." . "<br />\n";
    echo "User Id: " . $compruebausu . "\n";
    if($compruebausu){
        echo "The user exists";
        echo "<br />\n";
    } if(!$compruebausu){
        echo "The user doesn't exit";
        echo "<br />\n";
    }
    $pos_first = strpos($header->Subject,'?');
    $Subject_insert = substr($header->Subject,0,$pos_first);
    $values = strstr($header->Subject,'?');
    $valuesarray = str_split($values);
```
$eq=0;
$que=0;
$valin = ""
$arrayeq = array();
$arrayque = array();
foreach ($valuesarray as $index => $value){
    if($value == "="){
        $arrayeq[$eq] = $index;
        ++$eq;
    }
    if($value == "?"){
        $arrayque[$que] = $index;
        ++$que;
    }
}

//Get id mdl forum_post
$id = substr($values, $arrayeq[0]+1,$arrayque[1]-
$arrayeq[0]-1);
echo "\n";
echo "Id: " . $id;

//Get discussion
$dis = substr($values, $arrayeq[1]+1,$arrayque[2]-
$arrayeq[1]-1);
echo "\n";
echo "Discussion: " . $dis;

//Get parent
$parent = substr($values, $arrayeq[2]+1,$arrayque[3]-
$arrayeq[2]-1);
echo "\n";
echo "Parent: " . $parent;

//Get user
$usu = substr($values, $arrayeq[3]+1);
echo "\n";
echo "User post: " . $usu;

//Get Message
$message = imap_fetchbody($mbox,$i,'1');
$subject = strstr($message, 'From', true);
echo "\n";
echo "Message: " . $message;

if(check_user($fromfinal)){
    insert_post($dis, $parent, $usu,
    check_user($fromfinal), $Subject_insert, $message);
}

if(!check_user($fromfinal)){
    echo "THE USER IS NOT IN DDBB";
}

//Delete email
imap_delete($mbox, $i);

imap_close($mbox);
function check_user($mail){
    $link = mysql_connect('localhost', 'root', '');
    if(!$link){
        die('Could not connect: ' . mysql_error());
    }

    $db_selected = mysql_select_db('moodle', $link);
    if(!$db_selected){
        die('Can\'t use moodle: ' . mysql_error());
    }

    $query = sprintf("SELECT * FROM mdl_user
                      WHERE email = '$mail'" );
    $result = mysql_query($query);
    if(!$result){
        die('Invalid query: ' . mysql_error());
    }

    $num_rows = mysql_num_rows($result);
    $row = mysql_fetch_assoc($result);
    return $row['id'];
}

function insert_post($discussion, $parent, $usu, $userid, $subject, $message){
    echo "Insert.....";
    $link = mysql_connect('localhost', 'root', '');
    if(!$link){
        die('Could not connect: ' . mysql_error());
    }

    $db_selected = mysql_select_db('moodle', $link);
    if(!$db_selected){
        die('Can\'t use moodle: ' . mysql_error());
    }

    ++$parent;
    $sqlinsert= "INSERT into
               mdl_forum_posts(discussion,parent,userid,created,modified,subject,message)
               values ('$discussion','$parent','$userid',UNIX_TIMESTAMP(),UNIX_TIMESTAMP(),'$subject','$message');"
    mysql_query($sqlinsert,$link);
    echo "<br />
    echo "INSERTED ROW";
    echo "<br />
    echo "<br />
    ?>
Bibliography


[5]  PHP de Luxe Fortgeschrittene PHP-Programmierung
    Richard Samar, Christian Stocker
    ISBN 3-8266-0799-6
    Ed. mitp.

[6]  Die Referez PHP4
    Jörg Krause
    ISBN 3-446-21687-1
    Ed. Hanser.