

UDC 004.65:004.738.5

INFORMATION-MANAGEMENT SYSTEM, DATABASE, POSTCROSSING, WEB-APPLICATION, INFORMATION MODEL, CLIENT-SERVER TECHNOLOGY, DATA-BASED MANAGEMENT SYSTEM

Alexander Novak, Priya Deshmukh, Liam Johansson

This article describes the development and implementation of an automated Web-Application designed for managing information related to the Postcrossing community. The goal of this research is to create an efficient and scalable Information-Management System (IMS) capable of handling user data, postcard exchange statistics, and geographical parameters. The system architecture is based on Client-Server Technology, utilizing a structured Database for permanent storage and retrieval of complex relational data. The core of the system is the Information Model, which ensures the integrity and consistency of the data. The paper discusses the stages of design and implementation, including the selection of the Database Management System (DBMS) and the rationale behind choosing an appropriate technological stack to ensure high performance, security, and user-friendliness of the platform. The resulting solution provides a robust framework for improving the experience and data handling efficiency for Postcrossing participants.

Keywords: INFORMATION-MANAGEMENT SYSTEM, WEB-APPLICATION, CLIENT-SERVER TECHNOLOGY, DATABASE, POSTCROSSING, INFORMATION MODEL, DATABASE MANAGEMENT SYSTEM.

Introduction

In the 21st century, when information technologies have filled the life of modern man, there is a decline in interest in printed publications. Market forms of mass culture are replacing the book. Stimulating reading, reviving and developing interest in books among all segments of the population is an urgent task of society and, first of all, libraries, as cultural institutions designed to popularize the book.

The goal of the development is to create an information management system for the implementation of Postcrossing, which is designed to exchange various items between people from different countries and continents.

To achieve the set goal, the following tasks were set and solved in the work:

- the main problems of the subject area are analyzed;
- an analysis of existing analogues was conducted;
- an analysis and justification of the means of implementation was carried out;
- which is designed for the convenient exchange of various items between owners through advertisements on the site; The object of development is the requirements of book lovers, as well as the cultural position of literature in modern *society*. Bookcrossing will help attract book lovers, as well as meet the needs of the most advanced readers, through book exchange.

The subject of development is the creation of a web resource that is necessary to facilitate book exchange.

The development methods include the creation of a database using the freely distributed (free) MySQL DBMS and the PHP programming language to achieve the goal.

Theoretical part

Modern «products» of information technology, such as an electronic book or e-mail messages, are considered «hot» means of communication, because they are increasingly displacing and replacing paper books and postcards.

Postcrossing projects can be attributed to a completely new means of communication, which successfully combines both options for hot and cold means of communication. Another distinctive feature of this project is that thanks to information technologies, traditional forms of information transmission merge with the latest technologies. Postcrossing can be considered as a way of combining different means in achieving the goal of cultural work. Thanks to Postcrossing, it becomes possible to draw attention to a specific work. Thus, based on the considered characteristic features of Postcrossing, it is possible to talk about a special form of

communication carried out within the framework of this project and carried out using information technologies. In this project, in addition to the explicit goal, there are also various «private» goals of project participants, such as collecting, playing, belonging to a certain group, etc. The idea of these projects is to involve as many people as possible in the process. We can say that traditional communications are given «new life» with the help of information technologies and reverse communication between users.

The Postcrossing social movement appeared almost 15 years ago and today has about two million participants and over 10 million registered books around the world. Postcrossing automation is necessary so that society does not exchange random books on the street, but has a choice, that is, register their data on a certain site and inform other users about what literature is in the home library and which of these books can be provided for reading and sending to different cities and countries.

That is, the user informs other participants of the movement what literature can be provided by him and at the same time has the opportunity to find out what books can be received by him from other users. This process is much more convenient and faster in terms of acquiring the desired literature.

Practical Part

Figure 1 shows the information model of client-server interaction and the interaction of server components with each other. Since the program being developed is a web application, it is located on the server.

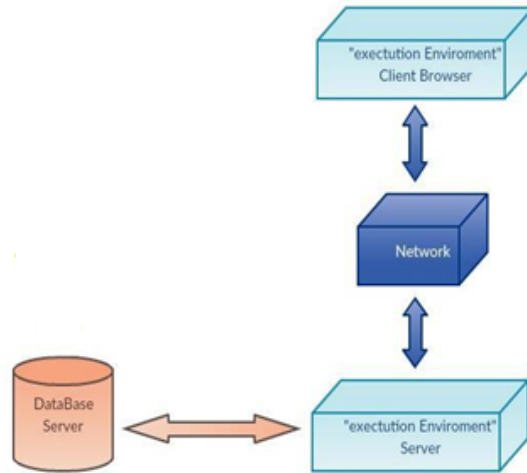


Figure 1 – information model

Client Browser – client agent (aka browser) for interacting with the server application using hypertext transfer;

Network – global Internet network;

Server – a software environment for the functioning of the application server and database, responds to client requests;

Database Server – a database server (for example, MySQL) that houses a database built on a model that meets the needs of the application.

The database stores all the necessary information for the full functioning of the system. It consists of five tables related to each other. The database is reduced to the third normal form and contains one-to-many relationships. The database diagram is shown in Figure 2:

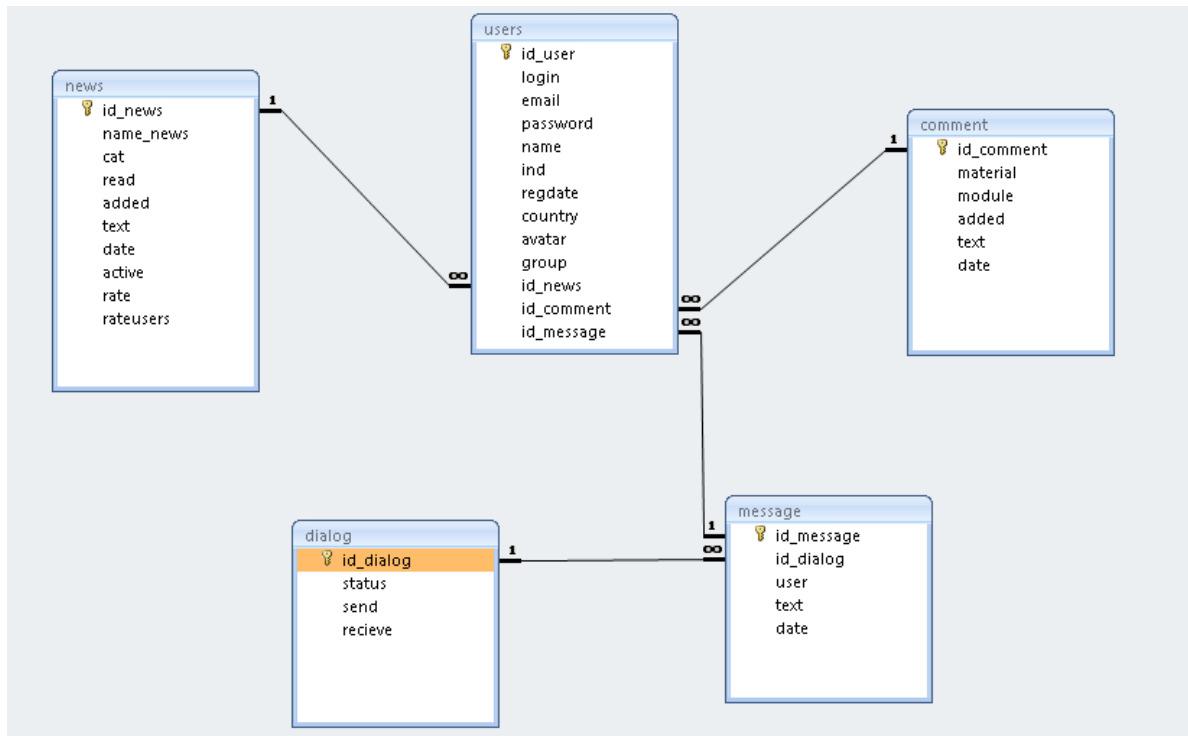


Figure 2 – Database schema

mysql_connect - Opens a connection to the MySQL server

List of parameters:

server

MySQL server. May also include a port number, such as "hostname:port" or a path to a local socket, such as ":/path/to/socket" for a local server.

If the PHP directive mysql.default_host is not defined (by default), then the default value is 'localhost:3306'. In SQL safe mode, this parameter is ignored and the value 'localhost:3306' is always used.

username

Username. The default value is specified by the mysql.default_user directive. In SQL safe mode, this parameter will be ignored and the user owning the server process will be used.

password

Password. The default value is specified by the mysql.default_password directive. In SQL safe mode, this parameter will be ignored and an empty string will be used as the password.

`new_link`

If a second call to `mysql_connect()` is made with the same arguments, no new connection will be established. Instead, the function returns a link to an already established connection. The `new_link` parameter can cause `mysql_connect()` to open another connection, even if a connection with similar parameters is already open. In SQL safe mode, this parameter is ignored.

HTML5 and CSS3 capabilities are used to implement the client part and work with the application's graphical interface.

This function opens a session for registering a new user in the system and creates a cookie file.

```

if (!$_POST['login'] or !$_POST['email'] or !$_POST['password'] or
!$_POST['name'] or !$_POST['ind'] or $_POST['country']>4)
    CheckRegInfo();
CheckRegInfo($_POST['login'], $_POST['email']);
$_SESSION['REGISTER_INFO'] = "$_POST[login], $_POST[email],
$_POST[password],$_POST[name],$_POST[ind],$_POST[country]";
$_SESSION['REGISTER_CONFIRM'] = GenPass($_POST['email'],
$_POST['login']);
mail($_POST['email'], 'Registration', 'Activation link:
http://zolt.zzz.com.ua/account/activate/code/'.MIX($_SESSION['REGISTER_CONFI
RM']), 'From: zolt.zzz.com.ua');

```

After opening a session, the user's personal data is entered into the database and displayed in the user's personal profile. Using the login and password entered into the database during registration, the user is authorized in the system.

After a user registers in the system, an email is sent to the mailbox specified during registration, which contains a link that, when clicked, activates the user's account.

```

else if ($Module == 'activate' and $Param['code'] and
$_SESSION['REGISTER_INFO'] and $_SESSION['REGISTER_CONFIRM'])
{

```

```

    if (MIX($_SESSION['REGISTER_CONFIRM']) != $Param['code'])
MessageSend(1, 'Activation is not possible.');
```

CheckRegInfo();
\$Exp = explode(',', \$_SESSION['REGISTER_INFO']);
mysqli_query(\$CONNECT, "INSERT INTO `users` VALUES (', '\$Exp[0]',
'\$Exp[1]', '\$Exp[2]', '\$Exp[3]', '\$Exp[4]', NOW(), \$Exp[5], 0, 0");
unset(\$_SESSION['REGISTER_INFO']);
unset(\$_SESSION['REGISTER_CONFIRM']);
MessageSend(3, 'Account verified.', '/login'); }

This function is used to authorize the system and grant the rights of a registered user.

```

else if ($Module == 'login' and $_POST['enter']) {
check_login($_POST['login']);
check_password($_POST['password']);
if (!$_POST['login'] or !$_POST['password']) MessageSend(1, 'Unable to
process the form.');
```

\$Row = mysqli_fetch_assoc(mysqli_query(\$CONNECT, "SELECT `password`
FROM `users` WHERE `login` = '\$_POST[login]'"));
if (\$Row['password'] != \$_POST['password']) MessageSend(1, 'Incorrect
login or password!');

\$Row = mysqli_fetch_assoc(mysqli_query(\$CONNECT, "SELECT `id`, `login`,
`email`, `password`, `name`, `ind`, `regdate`, `country`, `avatar`, `group` FROM
`users` WHERE `login` = '\$_POST[login]'"));

This function is implemented so that the user can log in to the system even if they forget their password.

```

if ($Module == 'restore' and $Param['code'] and
$_SESSION['RESTORE_INFO'] and $_SESSION['RESTORE_CONFIRM']) {
    if (MIX($_SESSION['RESTORE_CONFIRM']) != $Param['code'])
MessageSend(1, 'Restore is not possible.');
```

\$Random = RandomString(15);

```

$Password = $Random;
mysqli_query($CONNECT, "UPDATE `users` SET `password` = '$Password'
WHERE `login` = '$_SESSION[RESTORE_INFO]'");
unset($_SESSION['RESTORE_INFO']);
unset($_SESSION['RESTORE_CONFIRM']);
MessageSend(2, 'Password changed successfully, use new password
<b>'. $Random. '</b>', '/login');

```

In case of forgetting the password, the user is given the opportunity to go to the form in which it is necessary to enter his login, if this login is registered in the system, a letter is sent to the mailbox specified during registration containing a link, when clicking on which the user is given the opportunity to change the password. After the password change procedure, the user can log in to the system using the new password.

This function displays ads from registered users divided into categories.

```

if ($Module == 'main' and !$Param['cat']) {
if ($_SESSION['USER_GROUP'] != 2) $Active = 'WHERE `active` = 1';
$Param1 = 'SELECT `id`, `name`, `added`, `date`, `active` FROM `news`
' . $Active . ' ORDER BY `id` DESC LIMIT 0, 5';
$Param2 = 'SELECT `id`, `name`, `added`, `date`, `active` FROM `news`
' . $Active . ' ORDER BY `id` DESC LIMIT START, 5';
$Param3 = 'SELECT COUNT(`id`) FROM `news`';
$Param4 = '/news/main/page/';
}
else {
if ($_SESSION['USER_GROUP'] != 2) $Active = 'AND `active` = 1';
$Param1 = 'SELECT `id`, `name`, `added`, `date`, `active` FROM `news`
WHERE `cat` = ' . $Param['cat'] . ' ' . $Active . ' ORDER BY `id` DESC LIMIT 0, 5';
$Param2 = 'SELECT `id`, `name`, `added`, `date`, `active` FROM `news`
WHERE `cat` = ' . $Param['cat'] . ' ' . $Active . ' ORDER BY `id` DESC LIMIT START, 5';

```

```

$Param3 = 'SELECT COUNT(`id`) FROM `news` WHERE `cat` =
'.$Param['cat'];
$Param4 = '/news/main/cat/'.$Param['cat'].'/page/';
}

```

This function allows the user to add an ad to a specific category.

```

    if ($_SESSION['USER_GROUP'] == 2) $Active = 1;
else $Active = 0;

if ($_POST['enter'] and $_POST['text'] and $_POST['name'] and
$_POST['cat']) {
    $_POST['name'] = FormChars($_POST['name']);
    $_POST['cat'] += 0;

    mysqli_query($CONNECT, "INSERT INTO `news` VALUES (',
'$ _POST[name]', $_POST[cat], 0, '$ _SESSION[USER_LOGIN]', '$ _POST[text]',
NOW(),$Active,0,")");

    MessageSend(2, 'News added', '/news');

```

After adding, the ad goes through a verification procedure by the administrator; if the ad matches the topic of the selected category, it is placed on the site.

Using this function, the web resource administrator has the ability to edit the news category and make changes to its content.

```

$Param['id'] += 0;

if (!$Param['id']) MessageSend(1, 'News ID not specified', '/news');

$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `cat`,
`name`, `text` FROM `news` WHERE `id` = $Param[id]"));

if (!$Row['name']) MessageSend(1, 'News not found', '/news');

if ($_POST['enter'] and $_POST['text'] and $_POST['name'] and
$_POST['cat']) {
    $_POST['name'] = FormChars($_POST['name']);
    $_POST['text'] = FormChars($_POST['text']);
    $_POST['cat'] += 0;

```

```

mysql_query($CONNECT, "UPDATE `news` SET `name` = '$_POST[name]',
`cat` = $_POST[cat], `text` = '$_POST[text]' WHERE `id` = $Param[id]");
MessageSend(3, 'The news has been edited.',
'/news/material/id/'. $Param[id]);
}

```

Using this function, a user authorized in the system has the ability to add comments in the announcements section.

```

if($_POST['enter'] and $_POST['text']) {
$_POST['text'] = FormChars($_POST['text']);
$ID = ModuleID($Param['module']);
if($ID == 1) $Table = 'news';
else if($ID == 2) $Table = 'loads';
$Row = mysql_fetch_assoc(mysql_query($CONNECT, "SELECT `id` FROM
`$Table` WHERE `id` = $Param[id]"));
if(!$Row['id']) MessageSend(1, 'Material not found. ');
mysql_query($CONNECT, "INSERT INTO `comments` VALUES (',
$Param[id], $ID, '$_SESSION[USER_LOGIN]', $_POST[text]', NOW())");
MessageSend(3, 'Comment added. '); }

```

This function provides the user with the ability to conveniently search among ads on the site.

```

$Count = mysql_fetch_row(mysql_query($CONNECT, "SELECT
COUNT(`id`) FROM `$Module` WHERE `name` LIKE
'%$_SESSION[SEARCH]%'"));
if($Count[0]) {
if(!$Param['page']) {
$Param['page'] = 1;
$Result = mysql_query($CONNECT, "SELECT `id`, `name`, `date` FROM
`$Module` WHERE `name` LIKE '%$_SESSION[SEARCH]%' ORDER BY `id` DESC
LIMIT 0, 5");
} else {

```

```
$Start = ($Param['page'] - 1) * 5;
```

```
$Result = mysqli_query($CONNECT, str_replace('START', $Start, "SELECT  
'id', 'name', 'date' FROM '$Module' WHERE 'name' LIKE  
"%$_SESSION[SEARCH]%" ORDER BY 'id' DESC LIMIT START, 5")); }
```

This function provides the ability to edit the user's personal data specified during registration.

```
if ($_POST['name'] != $_SESSION['USER_NAME']) {  
    mysqli_query($CONNECT, "UPDATE `users` SET `name` = '$_POST[name]'  
WHERE `id` = $_SESSION[USER_ID]);  
    $_SESSION['USER_NAME'] = $_POST['name'];  
} if ($_POST['ind'] != $_SESSION['USER_IND']) {  
    mysqli_query($CONNECT, "UPDATE `users` SET `ind` = '$_POST[ind]'  
WHERE `id` = $_SESSION[USER_ID]);  
    $_SESSION['USER_IND'] = $_POST['ind'];}  
if (UserCountry($_POST['country']) != $_SESSION['USER_COUNTRY']) {  
    mysqli_query($CONNECT, "UPDATE `users` SET `country` =  
$_POST[country] WHERE `id` = $_SESSION[USER_ID]);  
    $_SESSION['USER_COUNTRY'] = UserCountry($_POST['country']);}
```

This function allows the user to rate the ad.

```
$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `rateusers`  
FROM '$Module' WHERE `id` = $Param[id]"));  
if (!isset($Row['rateusers'])) MessageSend(1, 'Material not found.', '/');  
if (in_array($_SESSION['USER_ID'], explode(',', $Row['rateusers'])))  
MessageSend(2, 'You have already rated this material.');  
mysqli_query($CONNECT, "UPDATE '$Module' SET `rate` = `rate` + 1,  
'rateusers' = CONCAT(rateusers, ',$_SESSION[USER_ID]') WHERE `id` =  
$Param[id]);  
MessageSend(3, 'Your rating has been accepted. Thank you.');
```

This function is necessary for exchanging personal messages between users registered in the system.

```

$Param['id'] += 0;
$Info = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `recv`,
`send` FROM `dialog` WHERE `id` = $Param[id]"));
if (!in_array($_SESSION['USER_ID'], $Info)) MessageSend(1, 'Dialog not
found.', '/');
if ($Info['recv'] == $_SESSION['USER_ID']) mysqli_query($CONNECT,
"UPDATE `dialog` SET `status` = 1 WHERE `id` = $Param[id]");
if ($Info['send'] == $_SESSION['USER_ID']) $Info['send'] = $Info['recv'];
$User = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `login`
FROM `users` WHERE `id` = $Info[send]"));

```

Also, in addition to sending a personal message using this function, the user dialog displays the name of the recipient and sender of the message.

Conclusions

The qualification work is a high-quality software product that meets all the requirements set for it. All planned functionality has been implemented. The software product is capable of performing all the tasks set for it and is fully ready for implementation.

As a result of the work, the following was implemented:

- The relevance of the problem being solved has been proven.
- The subject area analyzed.
- Analogues analyzed.
- A general approach to implementing the system has been formulated, and its components have been identified.
- Programming languages and tools that can be used to solve the task are analyzed.
- A database schema has been developed and all database tables and fields have been explained.
- The application scheme has been developed.
- An explanation of the stages of development and justification of means for developing specific tasks is presented.

– High-quality software has been developed that satisfies all the criteria set for it, and is also capable of solving all the problems presented in the subject area.

References

1. Vandyk, J. K., & Westgate, M. (2010). *Pro Drupal 7 Development: Third Edition*. Apress.
2. Holzner, S. (2007). *PHP in examples* (Trans. from English). Binom-Press.
3. Ullman, L. (2001). *Fundamentals of PHP Programming* (Trans. from English). DMK Press.
4. Mazurkevich, A., & Elov, D. (2003). *MB PHP: A programmer's desk book*. New Knowledge.
5. Thomson, L., & Welling, L. (2003). *Web Application Development on PHP and MySQL* (2nd ed., rev.; Trans. from English). DiaSoftYUP.
6. Gutmans, E., Bakken, S., & Retans, D. (2006). *PHP 5. Professional programming* (Trans. from English). Simbol-Plus.
7. Dubakov, M. (2002). *Web mastering*. BHV.
8. Official website of the PHP programming language. (n.d.). Retrieved from <http://www.php.net/>
9. SQLite vs MySQL vs PostgreSQL: comparison of database management systems
10. Resource access mode: <http://devacademy.ru/posts/sqlite-vs-mysql-vs-postgresql/>.
11. Groff, J. R., Weinberg, P. N., & Oppel, E. J. (2009). *SQL: The Complete Reference*.
12. Graber, M., & Lori. (2010). *Understanding SQL*.