Web Based Conference Management System

Zeno O. Popovici, Remus Brad *Member IEEE*, Eduard A. Stoica

***Abstract* — This paper describes the framework used to design a web-based conference management system, in order to assist conference organizers in the management of participants, papers, reviewing process and the accounting of a conference. It allows the natural processing flow of a paper, from on-line submission to camera-ready document. Providing three different class of access, the PHP-MySQL application can offer various statistics regarding participants and paper status, financial and venue confirmations. A notification system is also provided for easy conference updates.**

***Keywords* — Conference Management, PHP, MYSQL Database**

I. Introduction

THE need for a management system aroused in the early planning stages for the RoEduNet 2006 Conference. The event has grown constantly each year and the complexity of handling this year's amount of predicted data overwhelmed the human resources available, and also raised serious issues in the revision and submission processes.

As available conference systems were researched and mostly were found to be either to expensive or not providing the features we were looking for we decided to build a system customized on our needs.

II. Implementation

The initial requirements of the application were a simple registration system to accommodate users and articles and provide the necessary data for conference reviewers to work on. In the late planning stage of the application other features were introduced: three layer access, administrative options and statistics, revision and notification interfaces, automated notification system.

As the system needed to be operational as soon as possible after planning implementation began for the basic modules: registration and article submission. To assure data reliability and catch possible errors failsafe procedures were included at every database insert that prevent data corruption and send out automated e-mail notifications.

We choose to develop a PHP-MYSQL application because the both systems are OpenSource so they do not generate additional costs and also provide easy programming and maintenance. Also such a system would be 100% web-based accessible anytime, anywhere. An additional relatively new technology called Ajax has been used in the administration pages, to simplify administrative tasks and speed up response times.

A. Proposed Functionality

To summarize, the proposed components of such a system were:

- `User – Registration`
- `User – Login Data Recovery`
- `User – Paper Submission`
- `User – Venue Confirmation`
- `Reviewer – Paper Review`
- `Admin – Register User`
- `Admin – User Management`
- `Admin – Paper Management`
- `Admin – Application Configuration`
- `Admin – Conference Statistics`
- `Admin – SPAM System`
- `System – Login and Access`
- `System – Form Validation (VDaemon)`
- `System – Notifications`
- `System – Error Management`

B. User Roles and Application Life Cycles

There are three kinds of users of the conference management system: Administrators (e.g., Program Committee chairs, Organizing Committee) who administer papers and users, Users (e.g. Authors) who submit papers, and Reviewers assigned by the Administrator to review the submissions. The activities of a user in each of these roles over the lifecycle of this conference are summarized below.

Administrator Lifecycle:

1. An initial admin account with a predefined password exists in the system.
2. Registers as a normal user, authenticates trough the initial account and issues admin rights.
3. Changes his initial password.
4. Registers new users (e.g. invited Reviewers and Conference Committee members) and assigns new Administrator and Reviewer roles.
5. Opens registration and paper submission.
6. Assigns for every paper two Reviewers.
7. Sends SPAM messages to all users of the conference with updates.
8. Disables paper submission after deadline.
9. Enables venue confirmation.
10. Decides on final acceptance or rejection of reviewers and notifies users of the result, e-mail includes Reviewers marks and notes.
11. Registers payment information for users and sends notifications confirming payment, e-mail includes payment information.

Zeno O. Popovici – is with the "Lucian Blaga" University of Sibiu, Romania (phone: 40/744/684804; e-mail: zeno.popovici@ulbsibiu.ro).

Remus Brad – is currently assoc. prof. and director at "Lucian Blaga" University of Sibiu, Romania. (e-mail: remus.brad@ulbsibiu.ro).

Eduard A. Stoica - "Lucian Blaga" University of Sibiu, Romania. (e-mail: eduard.stoica@ulbsibiu.ro).

12. Rejects user that haven’t paid or resubmitted paper in the deadline.
13. Disables Venue confirmation.
14. Deletes unwanted users or papers.

Reviewer Lifecycle:

1. Visits the site and registers providing contact information and specific information.
2. Receives an automatic e-mail with login information, username and password.
3. Receives an automatic e-mail when an article was assigned to his user.
4. Logs in and checks the statistics regarding allocated papers.
5. Reviews papers one by one, assigning marks from 1 to 4 for every defined category and entering a short note explaining the marks.
6. Receives SPAM messages with conference updates.

User (Author) Lifecycle:

1. Visits the site and registers providing contact information and specific information.
2. Receives an automatic e-mail with login information, username and password.
3. Enters the paper submission page and submits up to 3 papers, entering the paper’s title, authors, abstract (not exceeding 1000 chars), keywords, selects paper first and second topic and attaches and PDF file containing the paper.
4. Receives an automatic e-mail confirming paper submission, e-mail contains article information.
5. Received and automatic e-mail regarding paper acceptance/rejection, e-mail contains 1st and 2nd reviewers marks and notes.
6. Logs in and checks the status of submitted papers.
7. Resubmits the camera-ready papers.
8. Receives an automatic message confirming paper resubmission, e-mail contains article information.
9. Enters the venue confirmation page and submits venue information.
10. Receives and automatic message confirming venue submission, e-mail contains selected venue information.
11. Receives SPAM messages with conference updates.

Paper Lifecycle

1. Paper is submitted with status “Pending” by the Author.
2. After both Reviewers evaluate the paper, it’s status changes to “Reviewed”.
3. Administrator changes status of the paper to “Rejected” (paper has been accepted) or “Resubmit” (paper has been rejected and camera-ready version is awaited).
4. User (Author) resubmits the camera-ready version and paper status changes to “Finalized”.

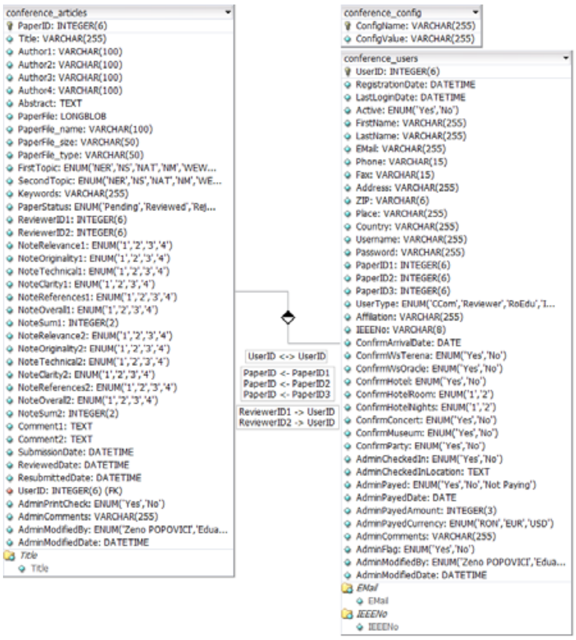
C. Data Model

The Data Model has been implemented around three tables “conference_articles”, “conference_users” and “conference_config”.

Relational elements are:

```
  UserID – UserID;
  PaperID – PaperID1, PaperID2, PaperID3;
  ReviewerID1, ReviewerID2 – UserID;
```

`UserID` is used to track users in the “conference_articles” table, `PaperID1-3` is used as a secondary method (precaution) to track the user’s papers, `ReviewerID1-2` are used to track Reviewer assignments to papers and take values equal to the `Reviewer UserID`.



```
conference_articles
PaperID: INTEGER(6)
Title: VARCHAR(255)
Author1: VARCHAR(100)
Author2: VARCHAR(100)
Author3: VARCHAR(100)
Author4: VARCHAR(100)
Abstract: TEXT
PaperFile: LONGBLOB
PaperFile_name: VARCHAR(100)
PaperFile_size: VARCHAR(50)
PaperFile_type: VARCHAR(50)
FirstTopic: ENUM('NER','NS','NAT','NM','WEW...
SecondTopic: ENUM('NER','NS','NAT','NM','WE...
Keywords: VARCHAR(255)
PaperStatus: ENUM('Pending','Reviewed','Rej...
ReviewerID1: INTEGER(6)
ReviewerID2: INTEGER(6)
NoteRelevance1: ENUM('1','2','3','4')
NoteOriginality1: ENUM('1','2','3','4')
NoteTechnical1: ENUM('1','2','3','4')
NoteClarity1: ENUM('1','2','3','4')
NoteReferences1: ENUM('1','2','3','4')
NoteOverall1: ENUM('1','2','3','4')
NoteSum1: INTEGER(2)
NoteRelevance2: ENUM('1','2','3','4')
NoteOriginality2: ENUM('1','2','3','4')
NoteTechnical2: ENUM('1','2','3','4')
NoteClarity2: ENUM('1','2','3','4')
NoteReferences2: ENUM('1','2','3','4')
NoteOverall2: ENUM('1','2','3','4')
NoteSum2: INTEGER(2)
Comment1: TEXT
Comment2: TEXT
SubmissionDate: DATETIME
ReviewedDate: DATETIME
ResubmittedDate: DATETIME
UserID: INTEGER(6) (FK)
AdminPrintCheck: ENUM('Yes','No')
AdminComments: VARCHAR(255)
AdminModifiedBy: ENUM('Zeno POPOVICI','Edua...
AdminModifiedDate: DATETIME
Title
  Title
```

```
UserID <-> UserID

PaperID <- PaperID1
PaperID <- PaperID2
PaperID <- PaperID3

ReviewerID1 -> UserID
ReviewerID2 -> UserID
```

```
conference_config
ConfigName: VARCHAR(255)
ConfigValue: VARCHAR(255)
```

```
conference_users
UserID: INTEGER(6)
RegistrationDate: DATETIME
LastLoginDate: DATETIME
Active: ENUM('Yes','No')
FirstName: VARCHAR(255)
LastName: VARCHAR(255)
EMail: VARCHAR(255)
Phone: VARCHAR(15)
Fax: VARCHAR(15)
Address: VARCHAR(255)
ZIP: VARCHAR(6)
Place: VARCHAR(255)
Country: VARCHAR(255)
Username: VARCHAR(255)
Password: VARCHAR(255)
PaperID1: INTEGER(6)
PaperID2: INTEGER(6)
PaperID3: INTEGER(6)
UserType: ENUM('CCom','Reviewer','RoEdu','I...
Affilation: VARCHAR(255)
IEEENo: VARCHAR(8)
ConfirmArrivalDate: DATE
ConfirmWsTerena: ENUM('Yes','No')
ConfirmWsOracle: ENUM('Yes','No')
ConfirmHotel: ENUM('Yes','No')
ConfirmHotelRoom: ENUM('1','2')
ConfirmHotelNights: ENUM('1','2')
ConfirmConcert: ENUM('Yes','No')
ConfirmMuseum: ENUM('Yes','No')
ConfirmParty: ENUM('Yes','No')
AdminCheckedIn: ENUM('Yes','No')
AdminCheckedInLocation: TEXT
AdminPayed: ENUM('Yes','No','Not Paying')
AdminPayedDate: DATE
AdminPayedAmount: INTEGER(3)
AdminPayedCurrency: ENUM('RON','EUR','USD')
AdminComments: VARCHAR(255)
AdminFlag: ENUM('Yes','No')
AdminModifiedBy: ENUM('Zeno POPOVICI','Edua...
AdminModifiedDate: DATETIME
EMail
  EMail
IEEENo
  IEEENo
```

Figure 1. Data Model Relational Diagram.

III. APPLICATION MODULES

A. User – Registration

The user registration module enables the registration of the user, inserts the information provided into the database, generates a random password and notifies the User and Administrator via the notification system with the login data.

B. User – Login Data Recovery

This module enables the user to recover its forgotten login information by providing his e-mail address, ZIP code and country. If all three are correct the module notifies the user via the notification system with the login data.

C. User – Paper Submission

The paper submission module enables the user to submit up to 3 papers, validates the entries via the validation system, inserts article data into the database if validation is passed, links article with the user via UserID and PaperID1-3 and notifies it via the notification system with the paper data. Abstract length is restricted to 1000 Characters; a warning is issued the user if limit is broken.

D. User – Venue Confirmation

This module enables the user to submit/view its venue information. If validation is passed, the information is inserted into the database and the user is notified via the notification system with the selected venue data.

E. Reviewer – Paper Review

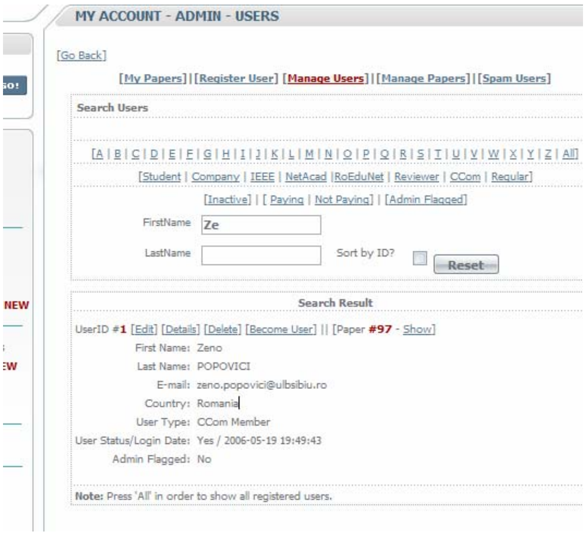
The paper review module provides the review interface for the Reviewer. This user type can review assigned papers one by one by. The Reviewer downloads download and evaluate the attached PDF file, assigns marks from 1 to 4 for all the defined categories (e.g. Originality, Technicality, Clarity, etc.) and provides a short recommendation for the paper author. If the form passes validation, article data is updated in the database and a notification is sent via the notification system to the Administrators.

F. Admin – Register User

This module is basically the same with the user registration module, except it does not require validation for the entered data and provides insertion of all fields. Notifications are sent to both Administrator and User.

G. Admin – User Management

The user management module provides the Administrator with a complete set of tools for managing the user database (Search, User Groups, Edit, Notify, Add Payment Data, Delete, etc.). This interface contains Ajax technology for easier and faster access. This interface also does not require validation. An additional feature in this module is an Admin-Notify flag which marks the users with special comments (e.g. Student Paper Contest).


MY ACCOUNT - ADMIN - USERS
[Go Back]
[My Papers]|[Register User] [Manage Users]|[Manage Papers]|[Spam Users]
[A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | All]
[Student | Company | IEEE | NetAcad |RoEduNet | Reviewer | CCom | Regular]
[Inactive] | [Paying | Not Paying] | [Admin Flagged]
Reset
NEW

Search Users

FirstName Ze

LastName

Sort by ID? □

Search Result

UserID #1 [Edit] [Details] [Delete] [Become User] || [Paper #97 - Show]

First Name:	Zeno
Last Name:	POPOVICI
E-mail:	zeno.popovici@ulbsibiu.ro
Country:	Romania
User Type:	CCom Member
User Status/Login Date:	Yes / 2006-05-19 19:49:43
Admin Flagged:	No

Note: Press 'All' in order to show all registered users.

Figure 2. User Management Module Overview

H. Admin – Paper Management

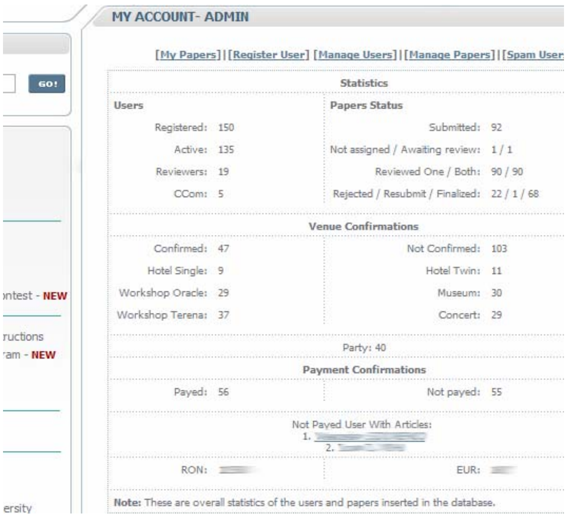
Same as the user management, this modules provides tools for paper management (Search, Edit, Reviewer assignment, Delete, etc.). Ajax has also implemented for this module. Validation is not required for successful update. An additional feature in this module is a Camera-Ready flag.

I. Admin – Application Configuration

This module manages the basic system configuration variables like enabling/disabling paper submission or registration modules, etc.

J. Admin – Conference Statistics

The statistics system generates module wide statistics regarding registered users and papers, payments, venue confirmations, etc. This gives the Administrator an overview of the conference (e.g. total available funds, confirmed concert participants, no. of reservations, etc.)


MY ACCOUNT- ADMIN
[My Papers]|[Register User] [Manage Users]|[Manage Papers]|[Spam User
GO!
ntest - NEW
ructions
ram - NEW
ersity

Statistics

Users		Papers Status	
Registered:	150	Submitted:	92
Active:	135	Not assigned / Awaiting review:	1 / 1
Reviewers:	19	Reviewed One / Both:	90 / 90
CCom:	5	Rejected / Resubmit / Finalized:	22 / 1 / 68

Venue Confirmations

Confirmed:	47	Not Confirmed:	103
Hotel Single:	9	Hotel Twin:	11
Workshop Oracle:	29	Museum:	30
Workshop Terena:	37	Concert:	29

Party: 40

Payment Confirmations

Payed:	56	Not payed:	55

Not Payed User With Articles:
1. [redacted]
2. [redacted]

RON:	[redacted]	EUR:	[redacted]

Note: These are overall statistics of the users and papers inserted in the database.

Figure 3. Statistics Module Overview

K. Admin – SPAM

The SPAM module enables the administrator to send out notifications to different user groups (e.g. e-mail Administrators, Reviewers, Users that haven’t paid, Users with pending articles, etc.). The Administrator can see the e-mail addresses before sending and a standard conference signature is automatically attached. All mails are sent from the official conference e-mail address and mails are sent via BCC so user privacy is kept and a reply address displayed.

L. System – Login and Access

The login and access module manages user authentication from login to logout throughout the site. Credentials are encrypted and verified via Cookies and PHP Session data at every page before any data is sent. If the user is inactive more than 2 hours the module logs out the user automatically. The module clearly differentiates between the three categories of users and gives the appropriate permissions accordingly.

M. System – Form Validation (VDaemon)

This module is not developed by the authors; it is a purchased, highly configurable shareware module that does Server-Side validations for all user forms. If errors are found, the specific fields are colored in red and notifications messages are displayed.

N. System – Notifications

This module is linked with all modules and takes care of all automatic notifications sent out by the other modules. Implementation is easy by using predefined functions.

O. System – Error Management

The error management module supervises all database inserts (only places where errors may occur) and takes care of code sanitizing. If an insert or update error occurs, the administrator is notified via the notification system with a full error description. Meanwhile, the User is reassured that the error has been recorded and assistance will be provided as soon as possible.

IV. Appendix

A. PHP

PHP is an open-source, reflective programming language and it's used mainly for developing dynamic web content and server-side applications. PHP allows interaction with a large number of relational database management systems, and runs on most major operating systems including and can interact with many major web servers.

B. MYSQL

MySQL is a robust, multithreaded, multi-user, SQL Database Management System (DBMS). MYSLQ is the most popular OpenSource SQL server, is very reliable, has an easy programming language and has little need for maintenance. MYSQL is the perfect "match" for PHP as special PHP functions are designed to interact with MYSQL.

C. Ajax

Ajax (Asynchronous JavaScript and XML), is a relatively new Web development technique for creating interactive web applications. Ajax is used so that web pages feel more responsive by exchanging small amounts of data with a server instead of reloading the page each time a user makes a change. This technology is meant to increase the web page's interactivity, speed, and usability. Ajax is not a technology in itself, but a term that refers to the use of a group of technologies together (Javascript, XMLHttpRequest Object and XML).

Javascript XMLHttp Object Initialisation Script:

```
function xmlhttpgetInit() {
// Initialize XMLHTTP ActiveX Object
var xmlhttp=false;
try {
  xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
  } catch (e) {
try {
  xmlhttp = new
ActiveXObject("Microsoft.XMLHTTP");
  } catch (E) {
  xmlhttp = false;
  }
}
// Initialize XMLHTTP Request
if (!xmlhttp && typeof
XMLHttpRequest!='undefined') {
  xmlhttp = new XMLHttpRequest();
}
}
```

JavaScript Ajax Search Function:

```
function doSearchName(firstname,lastname) {
// Search by FirstName, LastName from $_GET[];
input
// Display ... Searching ... for slow connections
durring search
  displayBox.innerHTML = "<strong>Searching
...</strong>";
// Open XMLHTTP Connection via GET method
  xmlhttp.open("GET","/control-
panel/admin/users/search.php?firstname="+firstnam
e+"&lastname="+lastname,true);
// If connection state changes, act
  xmlhttp.onreadystatechange=function() {
// If readystate OK, reset DIV content and list
search result
    if (xmlhttp.readyState==4) {
      displayBox.innerHTML = null;
      displayBox.innerHTML =
xmlhttp.responseText;
    }
  }
// Reset the connection
  xmlhttp.send(null);
}
```

V. Conclusion

The application has been proven to be very flexible, modular and secure:

- The modular system allowed development of-the-go.
- Application can be further developed adding new modules like article bidding or online collaboration systems and extending configurability.
- Security is implemented at every page level, so that conference data cannot be compromised at any time.

References

[1] Bakken S.S, Aulbach A, Schmid E, Winstead J, Wilson L.T., Lerdorf R, Zmievski A, Ahto J, *PHP Manual*, http://www.php.net/manual/en/

[2] Ben Forta: *"SQL in lectii de 10 minute"*, Editura teora

[3] Luke Welling, Laura Thomson: *"Dezvoltarea aplicatiilor web cu php si mysql"*, Editia a doua, Editura Teora, 2005

[4] Jon Stephens & Chad Russel: *"Beginning MySQL Database Design and Optimization"*

[5] HTML - http://www.w3.org/MarkUp/

[6] Charles Aulds: *"Administrarea serverului Apache"*, Editura Teora

[7] Gilmore W.J, *Beginning MySQL Tutorial*, http://www.devshed.com/Server_Side/MySQL/Intro/page1.html

[8] O. Reilly Media, Inc., *"MySQL Cookbook"* – http://www.oreilly.com

[9] O. Reilly Media, Inc., *"The PHP Cookbook"* – http://www.oreilly.com

[10] Paul Du Bois, *MYSQL*, Editura TEORA, Bucuresti, 2001

[11] Hakon Wium Lie & Bert Bos: “*Cascading Style Sheets – designing for the Web*” “written by the creators of CSS”, Editura Addison-Wesley, 2005

[12] Eric A. Meyer: *“ Cascading Style Sheets 2.0 Programmer's Reference”*, Editura Osborne/McGraw-Hill, 2001.

[13] Marc Johnson, *“Javascript Manual of Style”*, 1998

[14] *MySQL Reference Manual*, http://www.mysql.com/documentation/index.html

[19] *Wikipedia*, http://www.wikipedia.org