

# Online Conference Management

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

**Abstract** — This paper describes an application designed to manage submitted conference articles, participants, reviewers, venue and finances of a conference. The application allows complete online processing of a paper from submission to print-ready document. Build on three access levels, it also provides real-time statistics of submitted papers, finances and venue needed for accurate conference planning and also an integrated notification system for easy conference e-mail updates. The software is built in PHP and has a MYSQL backend.

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

## I. INTRODUCTION

EVERY organized event that exceeds a certain complexity level has to be thoroughly planned and organized in order to be successful, large scale conferences are make no exception. The complexity of handling a large amount of data can overwhelm the human resources available and also raise serious issues in the revision and submission processes or in the registration process.

There are many online conference systems out there, but mostly are either too expensive or not providing the customization level needed.

## II. IMPLEMENTATION

The initial requirements of the application are made of a simple registration system to accommodate users and articles and provide the necessary data for conference reviewers to work on. Extra features are introduced in the later in the development process: three layer access, administrative options and statistics, revision and notification interfaces, automated notification system.

One of the requirements was a modular design and therefore, any other software project, implementation begins with the basic modules: registration and article submission. To assure data reliability and catch possible errors failsafe procedures are included at every database insert that prevent data corruption and send out automated e-mail notifications.

The development of PHP application with a MYSQL backend was chosen because both platforms are OpenSource and therefore do not generate additional costs and provide easy programming and maintenance. Also such a system would be 100% web-based accessible anytime,

Zeno O. Popovici - "Lucian Blaga" University of Sibiu, Romania (e-mail: [zeno.popovici@ulbsibiu.ro](mailto:zeno.popovici@ulbsibiu.ro)).

Remus O. Brad - "Lucian Blaga" University of Sibiu, Romania. (e-mail: [remus.brad@ulbsibiu.ro](mailto:remus.brad@ulbsibiu.ro)).

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

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 through the initial account and issues admin rights to his user.
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 articles and notifies users of the result, e-mail includes Reviewers grades and notes.

- Registers payment information for users and sends notifications confirming payment, e-mail includes payment information.
- Rejects users that haven't paid or resubmitted paper in the deadline.
- Disables Venue confirmation.
- Deletes unwanted users or papers.

#### Reviewer Lifecycle:

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

#### User (Author) Lifecycle:

- Visits the site and registers providing contact and other specific information.
- Receives an automatic e-mail with login information, username and password.
- 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's first and second topic and attaches the PDF file containing the paper.
- Receives an automatic e-mail confirming paper submission, e-mail contains article information.
- Receives an automatic e-mail regarding paper acceptance/rejection, e-mail contains 1<sup>st</sup> and 2<sup>nd</sup> reviewers grading and notes.
- Logs in and checks the status of submitted papers.
- Resubmits the camera-ready papers.
- Receives an automatic message confirming paper resubmission, e-mail contains article information.
- Enters the venue confirmation page and submits venue information.
- Receives and automatic message confirming venue submission, e-mail contains confirmed venue data.
- Receives SPAM messages with conference updates.

#### Paper Lifecycle

- Paper is submitted with status "Pending" by the Author.
- After both Reviewers evaluate the paper, its status changes to "Reviewed".
- Administrator changes status of the paper to "Rejected" (paper has been rejected) or "Resubmit" (paper has been approved and camera-ready version is awaited).
- 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.

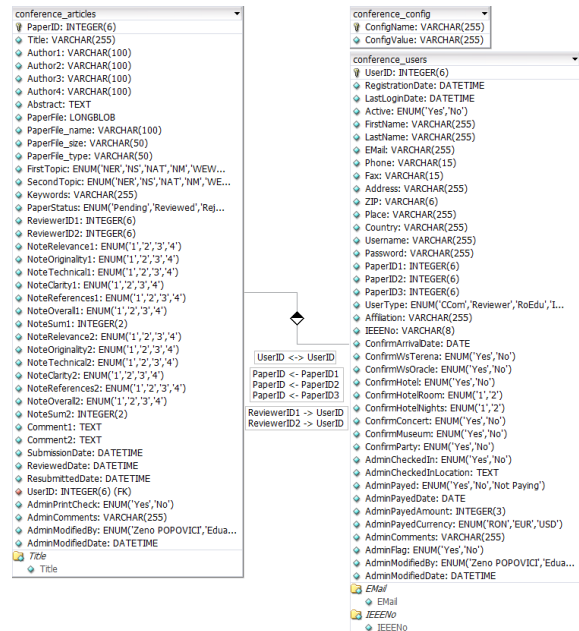


Figure 1. Data Model Relational Diagram.

### 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 confirmed venue data.

Figure 2. Venue Confirmation Screen

### 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. The Reviewer downloads and evaluates the attached PDF file, assigns grades 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.

Figure 3. Paper Review Screen

### 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. for a Student Paper Contest).

Figure 4. 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 site-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.)

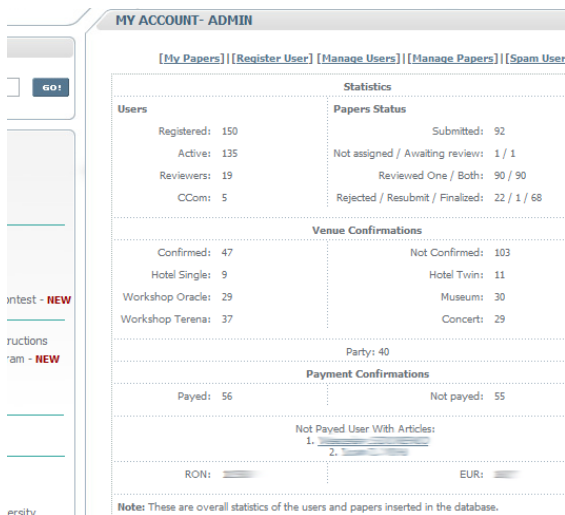


Figure 5. Statistics Module Overview

### K. Admin – SPAM

The SPAM module enables the administrator to send out notifications to different user groups (e.g. Administrators, Reviewers, Users that haven't paid, 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.

Figure 6. SPAM Module Overview

### 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 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). MYSQLQ 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 XMLHttpRequest Object Initialisation Script:

```
function xmlhttpgetInit() {
// Initialize XMLHttpRequest ActiveX Object
var xmlhttp=false;
try {
xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
try {
xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
} catch (E) {
xmlhttp = false;
}
}
// Initialize XMLHttpRequest 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="+firstname+"&lastna
me="+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 because of the modular design. User feedback was very good, and underlined the ergonomics of the application. Some users have even shown interest in purchasing it over other similar applications.

- The modular system allowed on-the-go development and debugging.
- Application can be further developed adding new modules like article bidding, online collaboration systems or online payment systems, and extending application configurability.

- Security of the application has proven very effective; all attempts to compromise conference data were unsuccessful.

## 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](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>