

SXC-JMS: A WEB -BASED JOURNAL MANAGEMENT SYSTEM

Sahon Bhattacharyya¹, Kalyan Mondal², Shalabh Agarwal³ and Asoke
Nath⁴

Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata, West Bengal, India

¹sahon.dgro@acm.org

²myfirstmail.k@gmail.com

³shalabh@sxccal.edu

⁴asokejoy@gmail.com

ABSTRACT

Due to immense advances in internet technologies in recent times, the entire method of maintenance and publication of journals has changed drastically. People prefer journals to be published online rather than to be published in printed form because of cost and availability. An online journal has multifaceted advantages, such as for example, publication time is instantaneous and the Editor can almost single-handedly manage the entire publication process. The readers can access the journal contents from anywhere and at anytime. By moving towards web-based journal systems, not only are we saving money, we are also making the transition towards the paradigms of green computing. In the United States and many European countries, most of the IT related journals are published online over the World Wide Web. India is also moving very fast in this direction. In this paper the authors have given a brief idea how to develop a web-based Journal Management System with all possible options in functionality and security. In the present research paper, the authors have demonstrated the first phase of the work and it will be implemented very soon in the authors' parent institution.

KEYWORDS

Online Journal, MVC Architecture, OOP PHP, Role Management

1. INTRODUCTION

Printed journals are now being open challenged by online journals. There are different problems associated with printed journals like the storage of papers is a serious problem in Green Computing. With this, the total cost of printed journals is very high in comparison to online journal, and the maintenance of printed journal is also very expensive and it needs physical space to store. On the other hand, in comparison to printed journals, online journals do not need much space and moreover the reader can access the journal from any part of the globe provided he has an internet connection. So due to space, time and cost constraints, online journal is always the better option than the classical printed journal now a days. For this reason, we have developed a software system which we have named SXC-JMS (St. Xavier's College – Journal Management System), which is a web-based Journal Management System and can be used in any Institution

Natarajan Meghanathan, et al. (Eds): ITCS, SIP, JSE-2012, CS & IT 04, pp. 417–427, 2012.

© CS & IT-CSCP 2012

DOI : 10.5121/csit.2012.2138

for submission and management of research papers. This software system provides online support for the management of a new e-journal. It facilitates the process of academic publishing, namely, the submission of a new paper, the process of peer review and various editorial associated functions. SXC-JMS is a web-based software system developed using WAMP environment (Windows + Apache + MySQL + PHP) and designed to run on any server. The software would be operated by two classes of users as follows:

A. *Unregistered Users or Readers*

Since this journal management system has been designed as per open access standards, unregistered users, i.e. users who have not created an account with the web application, can also have limited access to the journal issues and search facilities. Registration additionally provides email newsletters and alerts to the users.

B. *Registered Users*

The users those who have to create first and sometimes may be authenticated also. This category is further sub-divided into four subclasses:

1) *Authors*

This class consists of people who are in the process of making a new submission to a journal issue or have already submitted a paper. These people use the system to make submissions, track the progress of a paper once it is submitted and later submit revisions, if so required.

2) *Reviewers*

This class consists of people, who may or may not be authors, who have been assigned reviewing rights by the editorial team/the *Editor*/the *Administrator*. The people use the system to review papers which have been assigned to them, submit reviews and/or comments and track the progress of those papers.

3) *Sub-Editors*

This class of users consists of the editorial team who work under the *Editor* and use the system to monitor submissions made to various journal issues, assign reviewers to papers based on keyword matches, track reviews submitted, accept/reject papers (if proper rights are granted) and perform other associated functions.

4) *Editor*

This class, although ideally comprising of one single person, may also be customized to include multiple persons. As the name suggests, the Editor uses the system to create new journal issues in addition to those already performed by the above class. The Editor has the sole right to accept/reject finally the papers submitted for a given issue and create/publish an issue. The Editor can also use the system to obtain automatically generated statistics regarding the usage of the system as well as information reflecting the publication trends of the journal or of a particular issue.

5) Administrator

This class comprises of the person/s in charge of maintaining the integrity and security of the system. The Administrator is responsible for configuring the system settings, monitor action/error logs, take backups, define reports, etc.

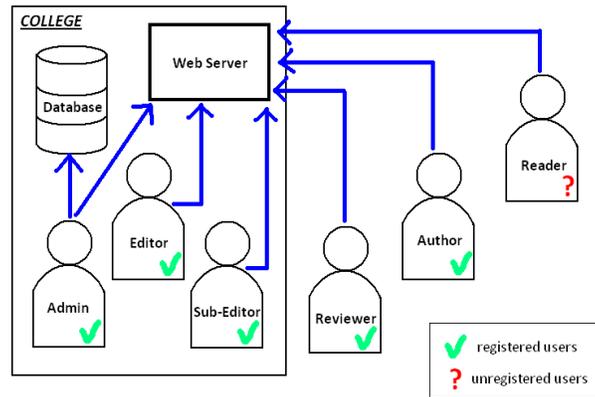


Figure 1. Registered and Unregistered Users

At present, our web application uses a variety of role management techniques. For example, the administrator can define who does what, for instance, what the Editor can do exactly, etc. and the Editor in turn can define the Sub-Editors who can work on which papers and what rights they might be granted for any given issue. The extremely lightweight and fast MVC architecture provided by *CodeIgniter* framework [Copyright (c) 2008 - 2011, EllisLab, Inc.], maintenance of a Registry modelled after the Windows Registry stored as a database table of Key/Value pairs which can customize almost all functionalities of the software. The use of Object Oriented Programming paradigms largely facilitates easy maintenance of code, server-side validation to prevent HTML form manipulation, etc.

The present software is technically more sophisticated than the commercial products which are available on the net. The use of MVC architecture along with CodeIgniter's lightweight performance has resulted in more efficient code and faster page loads. The present S/W takes only 0.8912 seconds to perform R/W operations as many as 7 databases on *localhost* and 1.0031 secs on a different machine on a LAN.

2. FUNCTIONALITY OF SXC-JMS

The system allows any user to access the website and search for papers, keywords and/or authors in various journal issues in the SXC-JMS database. For more features, the user has to log in to his account. If he does not have an account, he can opt to register for one. Once he has logged in, he may manage all his submissions, create a new submission or delete a submission, provided it has not been submitted for review already.

A reviewer will see a list of his reviews such as the papers that have been assigned to him, whether that has already reviewed or if the review is still pending, etc. If a person is a reviewer *and* an author, he will see both these tables.

The Sub-Editor and Editor modules exhibit strong cohesion and coupling because the functions of the Editor are defined as being those of the Sub-Editor plus the right to create/edit/delete an issue, provided it already does not exist/already exists/is not already published respectively.

Additionally, the Editor may also assign rights to the Sub-Editors to define which Sub-Editors may work on which issue/submission. This is an advanced feature though and in all probability, all Sub-Editors will be having equal rights. The Editor is required to create the forthcoming issues in advance since submission can be made only to an available forthcoming issue.

Now we will discuss the various modules and their functions which we have incorporated into the SXC-JMS system:

1) Login/Registration Module :

It allows a user

- To log in to his existing account OR
- To request a password-reset if he has forgotten his password OR
- To register for a new account

2) Submission Module :

It allows a user the following:

- To view all submissions made to the system
- Make a new submission
- Edit an existing submission, if not submitted for review already,
- Delete an existing submission, if not submitted for review already,
- Submit a draft submission
- View reviews made on reviewed submissions
- Add comments on the reviews
- View/Edit his profile
- Browse other profiles

3) Reviewer Module :

It allows a user to

- View all reviews assigned by the editorial office
- View a submission under review and comments made on it
- Add comment/remarks to a submission under review

4) Sub-Editor/Editor Module :

It allows a user to

- Create/edit/delete a new or forthcoming issue (Editor privileges required!)
- Manage the submissions made towards an available issue
- Add/remove reviewers for a given submission (Editor privileges required for removing reviewers!)
- View reviews made by reviewers
- Accept/reject submissions (Editor privileges required!)
- Shift a submission to the next available issue
- Remove a review submitted (Editor privileges required!)
- Send notification to all users of the system.
- Grant rights and privileges to Sub-Editors (Editor privileges required!)
- Generate usage and statistics reports. (Editor privileges required!)

5) Administrator Module

It allows a user to

- Survey all action-logs, error-logs and notification-logs.
- Backup all or specified portions of the database
- Manage/customize Registry – add/modify/delete keys/values
- Issue raw SQL to aid in management tasks
- Manage roles of various users

6) *Search Module (accessible to all)* :

It allows a user to

- Search the database by text, title, authors, keywords and/or abstracts, etc.

3. TECHNIQUES USED IN SXC-JMS

Our system has been developed for cross-compatibility with all major browsers MS Internet Explorer, Google Chrome, Mozilla Firefox and Opera. This will help all kind of users to use our S/W anywhere anytime as SXC-JMS is compatible with almost all web browsers.

A. *Model-View-Controller Architecture*

The present system makes the use of MVC architecture. The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The MVC framework is a lightweight, highly testable presentation framework that provides the biggest single advantage of the clear-cut separation between the Model, View, and Controller layers. It helps promote good design from the start. With MVC, there is complete control over what is rendered.

B. *Role Management (using Prime Numbers)*

In SXC-JMS we have not used a database rights table. Instead of that we have kept a special INT field along with the User data table. An advantage of this method is that we need not maintain strictly defined privilege groups for the system. Instead, we can create as many overlapping groups as we require. We shall try to elucidate this method below:

First, for each set of rights we have assigned a unique prime number in the registry. This we shall refer to as the role identifier. We have a role identifier for each action group that can be performed within the system. Currently, our system has 5 Role Identifiers

Now, both users and modules are assigned roles. The role of a user is dynamic and determines the rights of the user – what the user can do within the system. The role of a module, on the other hand, is static and determines who can use that module – what rights are required to access that module. (The roles of modules are hard-coded during development.)

To get the role of a User, we select those Role Identifiers that are applicable, that is, those actions which a particular User can perform and multiply the prime numbers. This gives us the specific role for a specific user.

Now, when an action request is received by a module of the system, the module first takes its own role and compares it with the role of the user who is requesting it. It computes the HCF of these two roles. If this is 1, the NULL Role Identifier, this means that access cannot be granted. If the HCF is greater than 1, that means that there is at least one Role Identifier in common, which implies that access can be granted.

We shall demonstrate with an example:

The Role Identifiers are as follows:

- (i) Reader=2
- (ii) Author =3
- (iii) Reviewer=5
- (iv) Sub-Editor =7
- (v) Editor=11
- (vi) Administrator=17
- (vii) System=19

Let there be four sample users of the system, User A, User B, User C and User D.

Let User A be just an author. Therefore,

$$\text{RoleOf(User A)} = 3$$

Let User B be an author *and* a reviewer. Therefore,

$$\text{RoleOf(User B)} = 3 * 5 = 15$$

Let User C be a staff in the editorial office. Therefore,

$$\text{RoleOf(User C)} = 7$$

And finally, let User D be the Editor, who, say, is also an author. Therefore,

$$\text{RoleOf(User D)} = 3 * 7 * 11 = 231$$

Now, we consider the roles of three actions of the system:

$$\text{RoleOf(W = right to submit a new article)} = 3$$

(since only authors may submit articles to the journal)

$$\text{RoleOf(X = right to create a new issue)} = 11$$

(since only the Editor may do this)

$$\text{RoleOf(Y = right to review an article)} = 5$$

(since only reviewers may do this)

$$\text{RoleOf(Z = right to assign reviewers)} = 7 * 11 = 77$$

(since both the Editor and the Sub-Editors may do this)

Now, let us say User A tried to perform Action W.

We compute $\text{HCF(RoleOf(A), RoleOf(W))} = 3 > 1$ (Access GRANTED)

Now, let us say User A tried to perform Action Z.

We compute $\text{HCF(RoleOf(A), RoleOf(Z))} = 1$ (Access DENIED)

Thus, we see that proper rights can be managed very efficiently.

Customization using an internal Registry

The system maintains an internal registry, organized as a dictionary of Key/Value pairs which store almost all customizable and configurable data of the system. This may be modified by the Administrator. The advantage of using such an approach is that major behaviours of the system can be altered instantly without making any changes to the internal programming. Sample key-values would be – *abstract.word.limit*, *username.length.limit*, *journal.title*, etc., etc. The registry contains not only the business logic of the system but also the UI configurations. The view of the Web UI, skins and views for example, may also be changed but setting the relevant keys in the Registry.

Data Validation Techniques

SXC-JMS uses server-side scripting with the help of the framework rather than doing it in the client-side. This is because our requirements specified that security should be the most important criterion and had a higher priority than access time. When an HTML form is submitted with invalid data, the controller first checks all data against specified validation rules. If this check fails, the view is invoked again with the error message and the correct fields repopulated. The controller proceeds only if the validation check succeeds. This method strongly shields the system from malicious attacks.

Object Oriented Design

While designing controllers and the models we have used the object-oriented features of PHP. The OOP PHP is similar to the way that using XHTML and CSS allows separation of structure and look from content; except OOP PHP separates functionality from implementation. One of the main advantages of OO programming is its ease of modification; objects can easily be modified and added to a system there by reducing maintenance costs. OO programming is considered to be better at modeling the real world than is procedural programming. It allows for more complicated and flexible interactions.

Template Engine (provided by base framework)

To facilitate rapid web designing, we have used the template engine provided by the base framework. Figure 2 given below shows the functions of this template engine. Basically, when a controller receives a request, it loads the model and passes the relevant data to the template-views, which are then parsed by the framework.

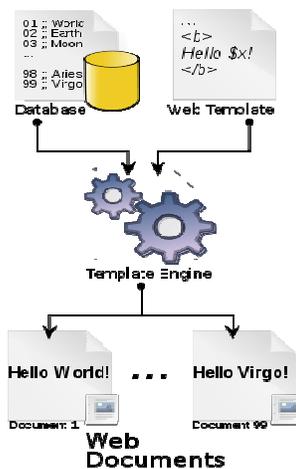


Figure 2. How does a Template Engine work?

3. RESULTS AND DISCUSSIONS

We have provided below some of the screenshots of different functions at different stages in the system as developed until now. The browser used for the screenshots is Google Chrome (15.0.874.121) on Windows 7 operating system.

Journal of Advanced Computing Sciences
St. Xavier's College (Autonomous), Kolkata, WB, India
ISSN: 2375964749

Home Call for Papers Current Issue Archives Search Contact

Register New account

* Enter Email ID: shahaj@ecacs.in (Mandatory Field)

* Enter Password: (8-16 characters)

Username

First Name: (e.g. John, James, etc.)

* Enter Middle Name: (If Any)

* Enter Last Name: Surname

* Enter Profession: (e.g. Professor, Grad Student, etc.)

Enter Organization: St. Xavier's College, Kolkata

Enter Phone Number: No

* Email Newsletter: Yes No

About Project JAACS • Developers • Copyrights & License • Troubleshooting

Figure 3. Registration Screen

Journal of Advanced Computing Sciences
St. Xavier's College (Autonomous), Kolkata, WB, India
ISSN: 2375964749

Home Call for Papers Current Issue Archives Search Contact

Manage Submissions

| Sub ID | Title | Status | Info | Edit | Paper |
|-----------|--------------------|-----------|------|------|-------|
| 13213192A | QR Code Encryption | In-Review | | | |
| 132136293 | Game Theory | ACCEPTED | | | |

Figure 4. Manage Submission Screen

Journal of Advanced Computing Sciences
St. Xavier's College (Autonomous), Kolkata, WB, India
ISSN: 2375964749

Home Call for Papers Current Issue Archives Search Contact

Manage Submissions for Volume 1/ Issue 1

Back to Forthcoming Issues

| Sub ID | Title | Submitted | Status | Info | Paper |
|-----------|---------------------------|------------|-----------|------|-------|
| 132136293 | Game Theory | 15-11-2011 | ACCEPTED | | |
| 132136445 | A New Way to Cryptography | 15-11-2011 | SUBMITTED | | |

Figure 5. Manage Issue Screen

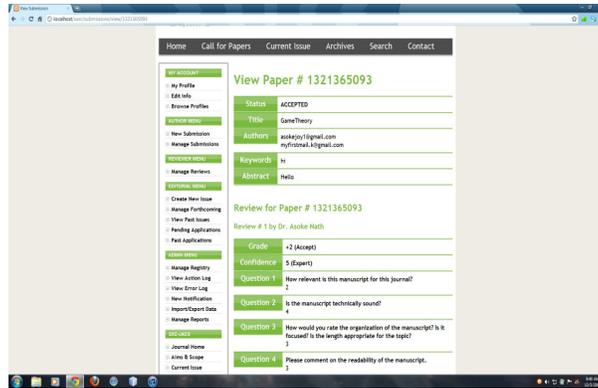


Figure 6. View Reviews Screen

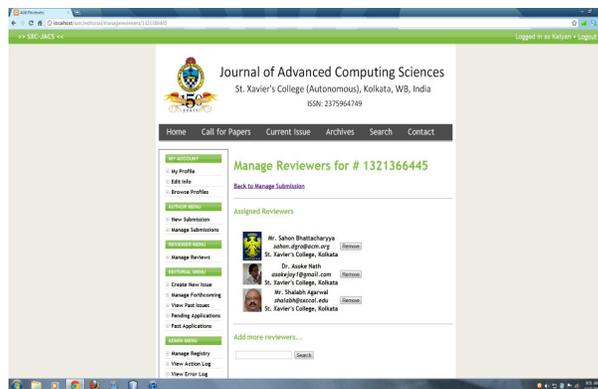


Figure 7. Add Reviewers Screen

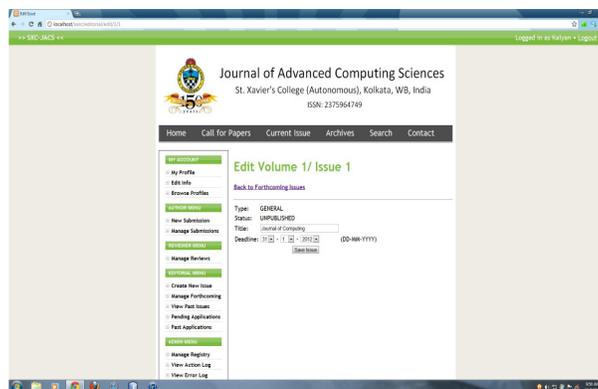


Figure 6. Edit Issue Screen

4. CONCLUSIONS

A SXC-JMS is a Web-based Journal Management System. The software is user-friendly and fully secured. This system can be used to submit and review papers and to prepare various reports. The entire S/W will run automatically with the peripheral interaction of the Administrator. Any user can access our S/W. Any user can enter as author or reviewer by entering his valid e-mail id that he used during registration. One reviewer can see the review comments of other reviewer but not their names, as the system supports the concept of blind refereeing. The editor has the highest priority. He can see the all reviewers report and authors submissions and make necessary decisions. One author cannot see the papers of another author until and unless it has been accepted for publication. In all these, the system has very efficiently fulfilled its requirements. In future, the present work can be further extended to make it more secured and more user friendly:

Firstly, we can provide a mechanism for allowing automatic combination of the accepted papers for a given issue, addition of a cover page, generation and addition of a contents page, paper and author indices, etc, so as to allow the provision of publishing hard copies of the proceedings., if need arise.

Secondly, we could alter the whole system to manage multiple journals. Only one installation of the S/W could be used to publish independent journals from different departments in the college. This reduces overload on the server and simplifies the whole process.

Thirdly, the database has been designed in such a way that in future we can add conference or workshop management support. This means in addition to the management of journals, this system can also manage conferences and workshops without any major development or modification.

ACKNOWLEDGEMENTS

The authors are grateful to the entire Department of Computer Science for being given the opportunity to work on this Web-based Journal management system. S.B. is grateful to Mr. Sanjoy Chatterjee (Software Engineer at Oracle Corporation) for his guidance on this paper. S.B. and K.M. are grateful to Somdip Dey, a fellow 3rd year student of Computer Science, for his contribution in preparing documentation and manual of this web application. S.B. and K.M. are also grateful to the 2nd year Computer Science (Honours) students, especially Mourjo Sen for his help with the diagrams and the formatting. A.N. and S.A. sincerely express their gratitude to the Principal, Rev. Fr. Dr. Felix Raj and Vice-Principal (Arts & Science), Rev. Fr. Jimmy Keepuram, of St. Xavier's College (Autonomous), Kolkata, for their constant inspiration to carry out such research projects. Also, A.N. is grateful to all current 3rd year Computer Science students (2009-2012) for their constant encouragement and support.

REFERENCES

- [1] CodeIgniter , PHP framework: <http://www.codeigniter.com/>
- [2] PHP Manual: <http://www.php.net/manual/en/book.php>
- [3] MVC Architecture (Model View Controller): <http://oreilly.com/php/archive/mvc-intro.html>
- [4] Model View Controller Pattern: <http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/mvc.html>
- [5] Slideshare, Web Application Architect : <http://www.slideshare.net/spriebsch/web-application-architecture-2509476>

- [6] The Architecture of Web Application: http://www.ibm.com/developerworks/ibm/library/it-boochoch_web/
- [7] T. Dinh-Trong and M. Bieman . The FreeBSD Project: A Replication Case Study of Open Source Development. IEEE Trans. Software Engineering, 31(6):481--494, June 2005.
- [8] Temple University, Software Management and Compliance Guideline: http://www.temple.edu/cs/policies/software_compliance.pdf
- [9] Software Quality and Standards by Dr. J.A.Bednar, Dr. D Robertson: http://www.inf.ed.ac.uk/teaching/courses/seoc2/2004_2005/slides/quality.pdf

Authors

Sahon Bhattacharyya is currently a final-year undergraduate student at the Department of Computer Science at St. Xavier's College (Autonomous), Kolkata. His research interests include web engineering, web development using J2EE and PHP, cloud computing, cognitive science, cognitive psychology, philosophy of mind, ethics, etc. He has been working in the field of web development for the past five years.



Kalyan Mondal is currently a final-year undergraduate student at the Department of Computer Science at St. Xavier's College (Autonomous), Kolkata. His research interests include web engineering, web development using PHP and J2EE, cloud computing, graphics designing, ethical hacking, etc. He has been working in the fields of web development for past four years.



Dr. Asoke Nath is Assistant Professor at St. Xavier's College (Autonomous), Kolkata. He is presently working in the field of Cryptography and Steganography. Some of the new symmetric key algorithms he has worked on include MSA, DJSA, DJSSA, etc. His other works include data hiding in ASCII files etc. He and his colleagues are currently working on amalgamating symmetric and asymmetric key cryptosystems. He is a member of IEEE, CSI and is on the editorial boards of various journals. Dr. Nath has presented many research papers at and attended numerous International Conferences.



Shalabh Agarwal is an Assistant Professor in Computer Science, faculty member in the departments of Journalism, Business Administration and Biotechnology and the Systems Director of the Computer Centre at St. Xavier's College (Autonomous), Kolkata. His research interests include Software Engineering, E-Governance, Cyber Law, Green Computing and Pervasive Computing. He has been involved in the development and administration of many computerized systems at St. Xavier's for the last two decades. He has also published numerous papers at international conferences.

