SXC-JACS: An E-Journal Management System

Thesis Project submitted to St. Xavier's College (Autonomous), Kolkata in partial fulfilment of the requirements for the award of the degree of

B. Sc. Computer Science (Honours)

Sahon Bhattacharyya
Kalyan Mondal
Somdip Dey

Department of Computer Science
St. Xavier's College (Autonomous)
30, Park Street, Kolkata 700016
West Bengal, India
[April 2012]

ST. XAVIER'S COLLEGE
(Autonomous)
Department of Arts & Science

30, Mother Teresa Sarani
Kolkata - 700 016
Phone : 2287-7278
Fax : 033-2280 1927

CERTIFICATE

Certified that the Thesis Project entitled, "*SXC-JACS: An E-Journal Management System*", submitted by Sahon Bhattacharyya (501), Kalyan Mondal (566) and Somdip Dey (558) in partial fulfilment of the requirements for the award of the degree of B.Sc. Computer Science (Honours) from this institution is an original work of their own. This work has not been submitted for any other degree in this institution or any other institution.

Anal Acharya.
Prof. Anal Acharya 11/04/2012
(Head of Department)
Assistant Professor
Department of Computer Science

11/04/2012
Prof. Dr. Asoke Nath
(Supervisor)
Assistant Professor
Department of Computer Science

Head of the Department,
Department of Computer Science,
St. Xavier's College,
30, Park Street,
Kolkata-700 016

12/04/2012
External Examiner

Table of Contents

1. Introduction

Journals and conferences constitute the primary channels of academic research publication in today's academic world. The bulk of journal publications in the last century have been mainly through paper-based publications. The process of these paper-based publications is tedious, costly and much time-consuming and not to mention, the environmental waste management problems. It initially consisted of mailing out the calls for papers to authors, receiving their submissions, submitting them for review, receiving reviews and after the decisions are made, publishing the copies and distributing them. But with the advent of client-server models of computing and the World Wide Web, it may be found that almost every journal enjoys the benefits of Information Technology in its working. We will differentiate such applications by the way they digitize the processes of call-for-papers, submissions, reviewing and publication. Some use the Web to a lesser degree as in that they merely advertise their calls for papers online but process the submissions, reviewing and publication offline. Other systems digitize more of the afore-mentioned processes. We wish to reserve the designation of an *e-journal* exclusively to those journals that perform all these processes online. Currently all journals are in the process of a transition from completely offline processes to completely online processes. Certain journals, notably those published by Springer, Taylor & Francis, etc, offer both print and electronic versions of their journals. Other professional organizations that follow suit and are moving towards sole digitization include ACM's Digital Library and IEEE's Xplore.

2. Aims & Objectives

In our project, we have developed an e-journal management system targeted at the use of the Department of Computer Science at St. Xavier's College (Autonomous), Kolkata. Our aim here is to provide the software infrastructure for the online management of this journal. This e-journal supports open access and the publications are available free of charge. Authors may register with this system and submit their papers. The Editor may log in to the system and assign reviewers to the papers. The reviewers may access the system and submit their reviews. Once this is done, the Editor and his team of Sub Editors may accept/reject the paper via this system. And the authors are duly notified by email automatically. When all decisions for an issue are made, the Editor may then publish the journal issue online and it is then accessible for the public.

3. System Specifications

For the system to function properly, there are some requirements on both the server system and the client system. They are as listed below:

3.1. Hardware Specifications

For the server system, the requirements are:

Part	Specification	Operations
Processor	Intel Pentium 4 or AMD Athlon 2.4 GHz or Higher	Normal
Motherboard	Equivalent Intel Chipset or AMD Chipset or Higher	Normal
HD Space	Standard	Normal
RAM	512 MB DDR2 or Higher	Normal
Monitor	Standard	Normal
Optical Drive	52x RW CD/DVD ROM	Normal
Mouse & Keyboard	Normal Optical Mouse and 104-key Base Keyboard or Higher	Normal

For the client system, any PC system with normal operating mode is adequate.

3.2. Software Specifications

For the server system, the following are required:

Requirement	Installed Software
HTTP Server	Apache HTTP Server
Database Server	MySQL Community Server 5.1
PHP	PHP 5

For the client system, any standard web browser is sufficient. Examples include Microsoft Internet Explorer (IE 6 +), Google Chrome (all versions), Mozilla Firefox (all versions), etc.

4. System Analysis

4.1. Requirements & Feasibility Study

To determine the requirements of the SXC-JACS, we are at first required to understand what is the basic objective of the web application that will be performing the procedure of managing journals online and how will it be implemented on the Client-Server paradigm. The Server side development has to be done in such a way, which will ensure minimum and secure data flow to the client side for security concerns. The Client side development is mainly done to facilitate easy interaction of the user with the website. The issues such as the updating the Journal issues and the submitted papers for the Journal issues in the database, and updating the papers submitted by the authors (registered users) are vital part of the system. The creation of user accounts and validating the users or providing permissions to guest users (unregistered users or readers) are also many important issues that need to be maintained by the web based application.

Client Server Communication

The SXC:JACS involves adequate communication between the Client and Server for optimum secure data flow & high security factors. The web application will be designed to maintain proper integration between the Client and Server.

Database Management

Here MySQL has been used as the back-end database which will store all required data of the system, simultaneously, maintaining back ups for recovery purpose. Database constraints will be used to maintain high data security.

Security Constraints

For any web-based application, the need of high security is a must. In this case several high security algorithms will be used on the user name and password to maintain high security along with other measures.

User ID and System Management

The administrator will monitor the validation and authentication of user-ids and will also maintain the integrity and security of the system. The administrator is responsible for configuring the system settings, monitor action/error logs, take back ups, define reports, etc.

Journal Management Procedure

Journal Management is the main work of this system and to achieve the goal different techniques have been used to make the system stable and working. Different techniques used to achieve and build the system comprises of: Model-View-Controller Architecture, Role Management, customization using an internal registry, data validation techniques, object oriented design, etc.

The most vital outcome of the Preliminary Investigation is determining the feasibility of the proposed system. There are various aspects of feasibility study such as:

Economic Feasibility

Economic feasibility is most frequently used for evaluation of the effectiveness of the system. More commonly know as cost/benefit analysis. The procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. This part of feasibility study gives the top management of the economic justification for the new system. This is an important input to the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases. In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving a lot of time.

Technical Feasibility

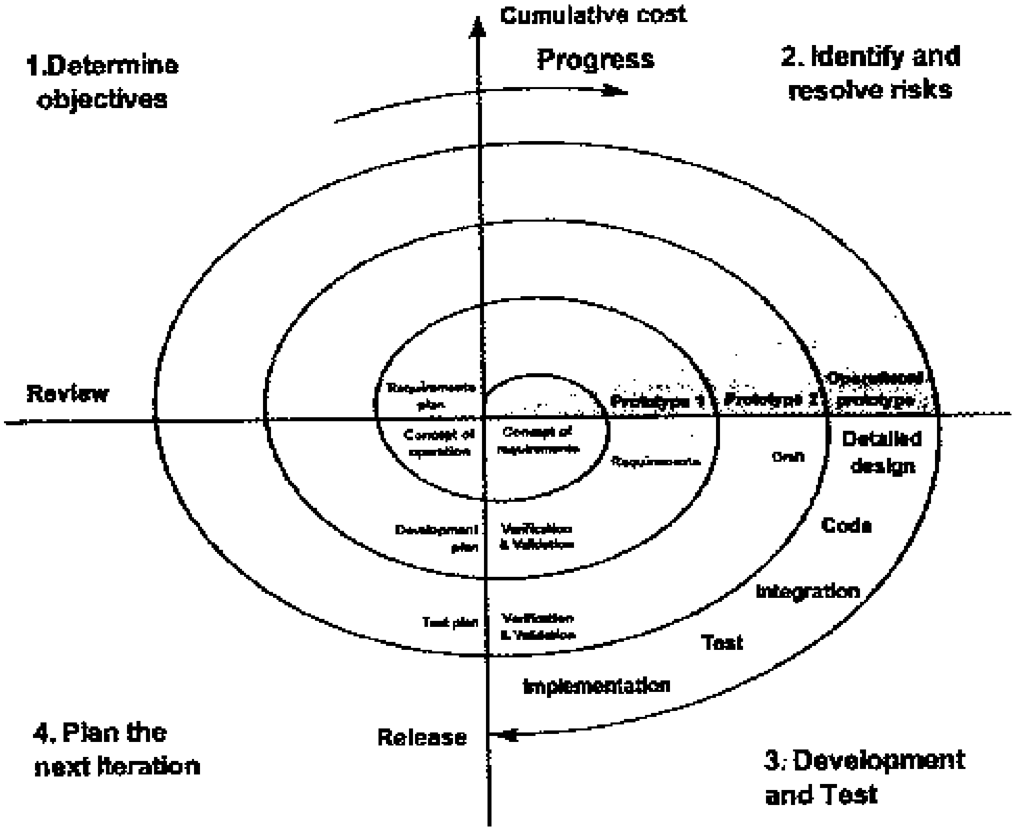
Technical feasibility focuses on the existing manual system of Journal archiving and management, and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are defined. It is also one of the important phases of the system development activities. The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view and management convenience that the project is operationally feasible.

Behavioural Feasibility

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

4.2. Software Development Life Cycle

r project, we have developed our software based on the *spiral model* of software development. But actual SDLC used is not the spiral model *per se*. Subject to various constraints and limiting factors, luding limited time, etc, we found the need to mould this model to our specific requirements. In other ds, the spiral model is the closest formally established model to our development life cycle. The al spiral model is described below:



spiral model combines the idea of iterative development (prototyping) with the systematic, olled aspects of the waterfall model. It allows for incremental releases of the product, or mental refinement through each time around the spiral. The spiral model also explicitly des risk management within software development. Identifying major risks, both technical and gerial, and determining how to lessen the risk helps keep the software development process under ol.

spiral model is based on continuous refinement of key products for requirements definition alysis, system and software design, and implementation (the code). At each iteration around the e, the products are extensions of an earlier product. This model uses many of the same phases as the rfall model, in essentially the same order, separated by planning, risk assessment, and the building totypes and simulations.

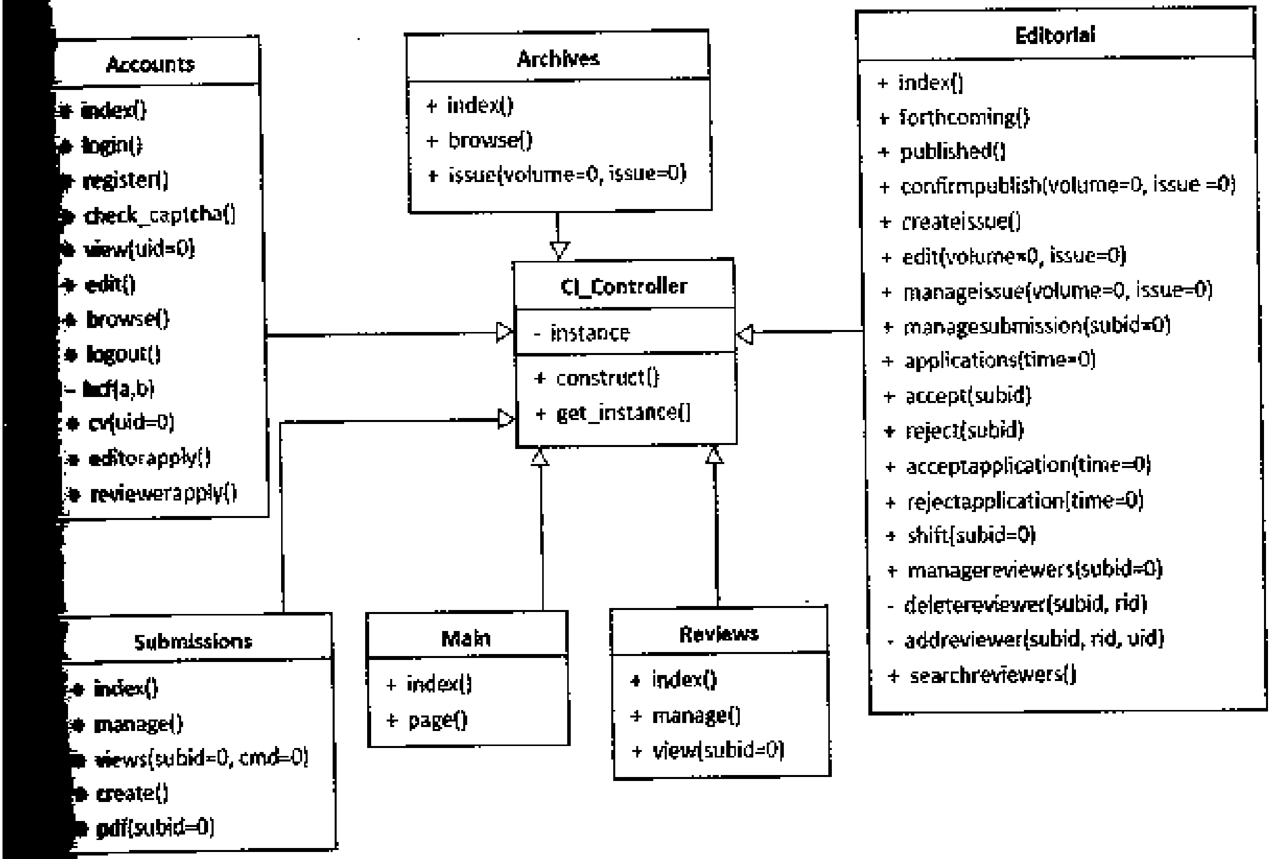
Documents are produced when they are required, and the content reflects the information necessary at that point in the process. All documents will not be created at the beginning of the process, nor all at the end (hopefully). Like the product they define, the documents are works in progress. The idea is to have a continuous stream of products produced and available for user review.

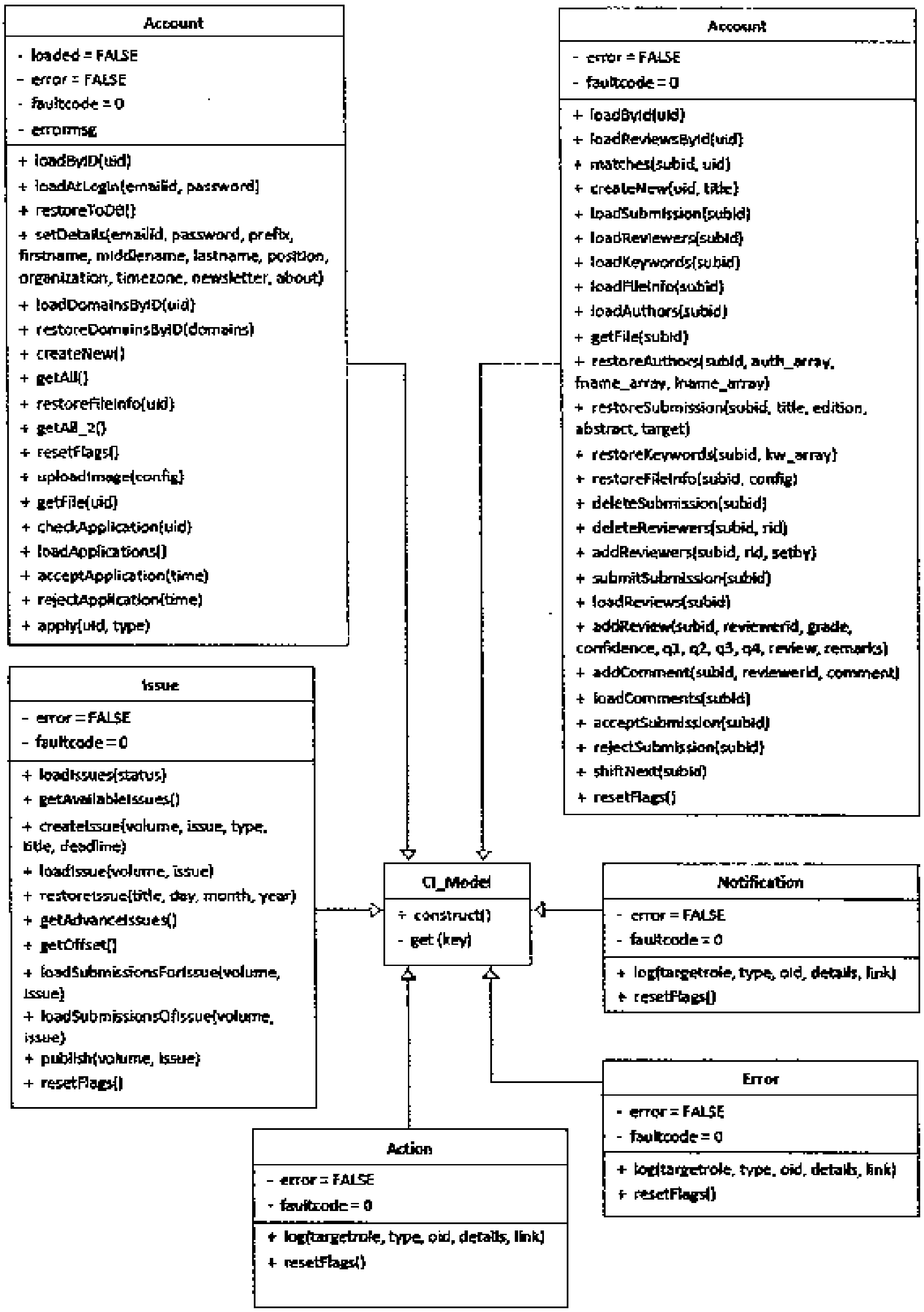
The spiral life cycle model allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design. This method is consistent with approaches that have multiple software builds and releases and allows for making an orderly transition to a maintenance activity. Another positive aspect is that the spiral model forces early user involvement in the system development effort. For projects with heavy user interfacing, such as user application programs or instrument interface applications, such involvement is helpful.

a. Starting at the center, each turn around the spiral goes through several task regions:
b. Determine the objectives, alternatives, and constraints on the new iteration.
c. Evaluate alternatives and identify and resolve risk issues.
d. Develop and verify the product for this iteration.
e. Plan the next iteration.

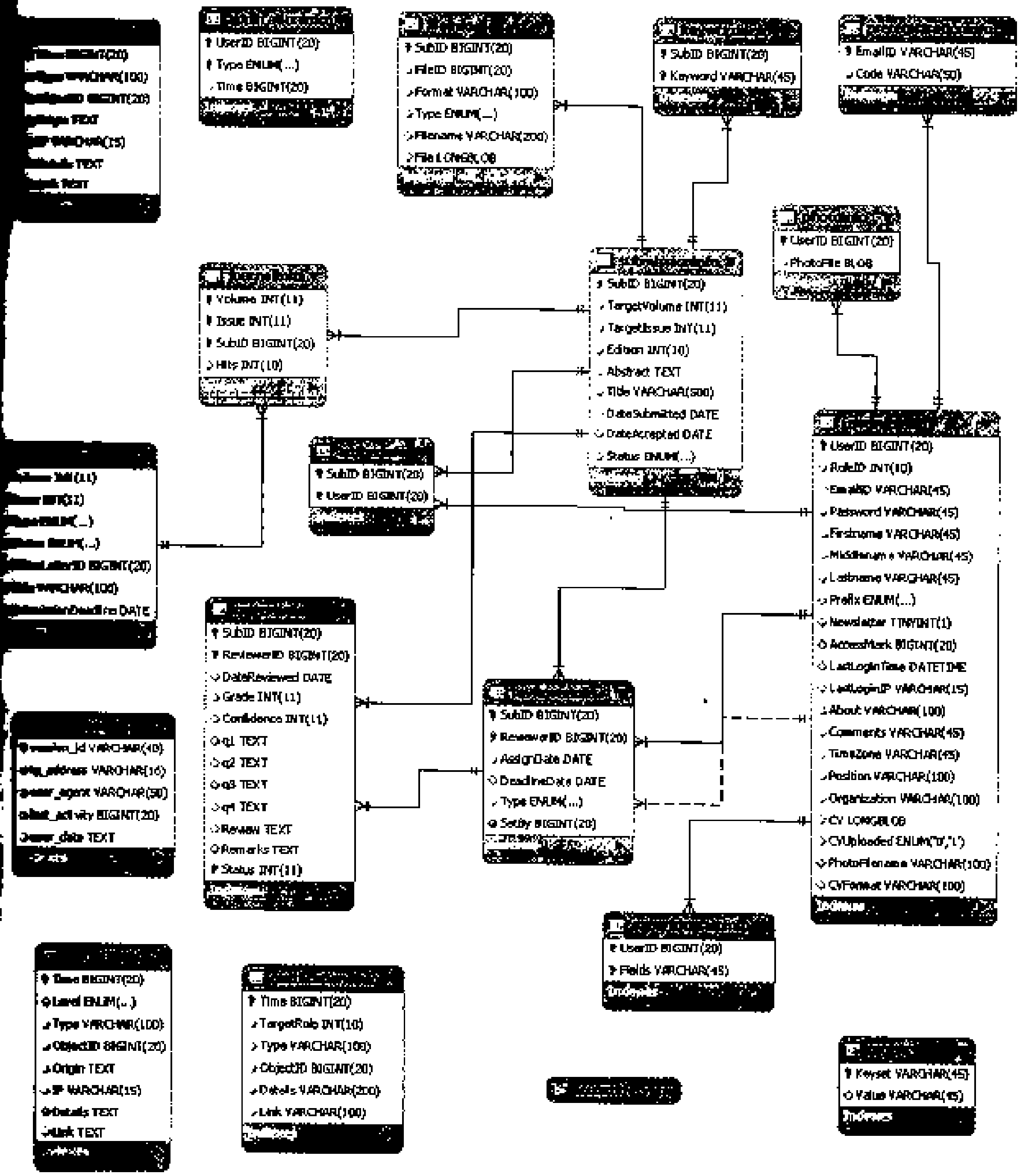
Note that the requirements activity takes place in multiple sections and in multiple iterations, just as planning and risk analysis occur in multiple places. Final design, implementation, integration, and test occur in iteration 4. The spiral can be repeated multiple times for multiple builds. Using this method of development, some functionality can be delivered to the user faster than the waterfall method. The spiral method also helps manage risk and uncertainty by allowing multiple decision points and by explicitly admitting that all of anything cannot be known before the subsequent activity starts.

4.3. Class Diagrams




Account
- loaded = FALSE
- error = FALSE
- faultcode = 0
- errormsg
+ loadByID(uid)
+ loadAtLogin(emailid, password)
+ restoreToDB()
+ setDetails(emailid, password, prefix, firstname, middlename, lastname, position, organization, timezone, newsletter, about)
+ loadDomainsByID(uid)
+ restoreDomainsByID(domains)
+ createNew()
+ getAll()
+ restorefileInfo(uid)
+ getAll_2()
+ resetFlags()
+ uploadImage(config)
+ getFile(uid)
+ checkApplication(uid)
+ loadApplications()
+ acceptApplication(time)
+ rejectApplication(time)
+ apply(uid, type)
Account
- error = FALSE
- faultcode = 0
+ loadById(uid)
+ loadReviewsById(uid)
+ matches(subid, uid)
+ createNew(uid, title)
+ loadSubmission(subid)
+ loadReviewers(subid)
+ loadKeywords(subid)
+ loadFileInfo(subid)
+ loadAuthors(subid)
+ getFile(subid)
+ restoreAuthors(subid, auth_array, fname_array, lname_array)
+ restoreSubmission(subid, title, edition, abstract, target)
+ restoreKeywords(subid, kw_array)
+ restoreFileInfo(subid, config)
+ deleteSubmission(subid)
+ deleteReviewers(subid, rid)
+ addReviewers(subid, rid, setby)
+ submitSubmission(subid)
+ loadReviews(subid)
+ addReview(subid, reviewerid, grade, confidence, q1, q2, q3, q4, review, remarks)
+ addComment(subid, reviewerid, comment)
+ loadComments(subid)
+ acceptSubmission(subid)
+ rejectSubmission(subid)
+ shiftNext(subid)
+ resetFlags()
Issue
- error = FALSE
- faultcode = 0
+ loadIssues(status)
+ getAvailableIssues()
+ createIssue(volume, issue, type, title, deadline)
+ loadIssue(volume, issue)
+ restoreIssue(title, day, month, year)
+ getAdvanceIssues()
+ getOffset()
+ loadSubmissionsForIssue(volume, issue)
+ loadSubmissionsOfIssue(volume, issue)
+ publish(volume, issue)
+ resetFlags()
CI_Model
+ construct()
- get (key)
Notification
- error = FALSE
- faultcode = 0
+ log(targetrole, type, oid, details, link)
+ resetFlags()
Error
- error = FALSE
- faultcode = 0
+ log(targetrole, type, oid, details, link)
+ resetFlags()
Action
- error = FALSE
- faultcode = 0
+ log(targetrole, type, oid, details, link)
+ resetFlags()

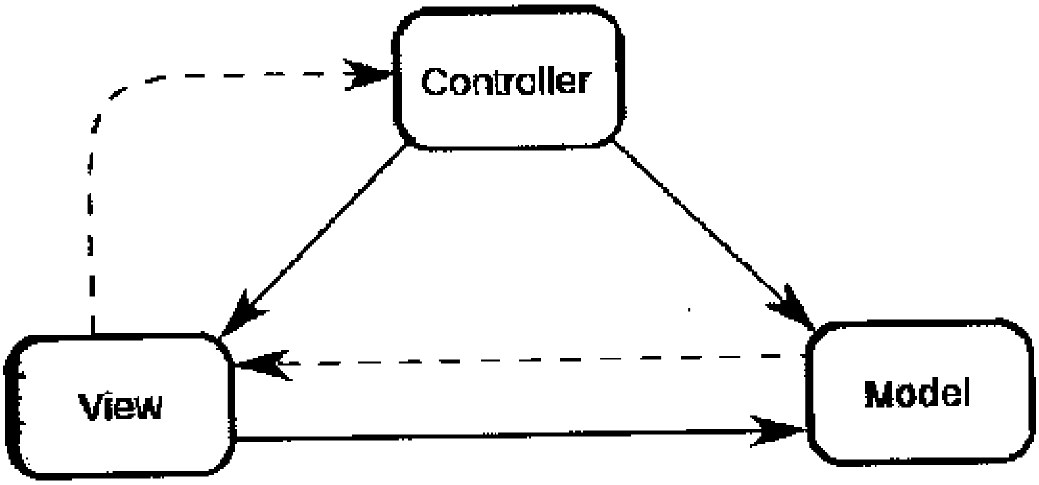
4.4. Database Design



5. System Design

5.1. Design Overview

our project, we have employed the Model-View-Controller architecture. In software eering, Model–View Controller (MVC) is an architectural pattern that splits interactions between and applications into three roles: the Model(business logic), the View (user interface), and Controller (user input). This separation of concerns facilitates the independent development, testing, maintenance of each role.



Model manages the behavior and data of the application domain, responds to requests for mation about its state (usually from the view), and responds to instructions to change state (usually the controller). In event-driven systems, the model notifies observers (usually views) when the mation changes so that they can react.

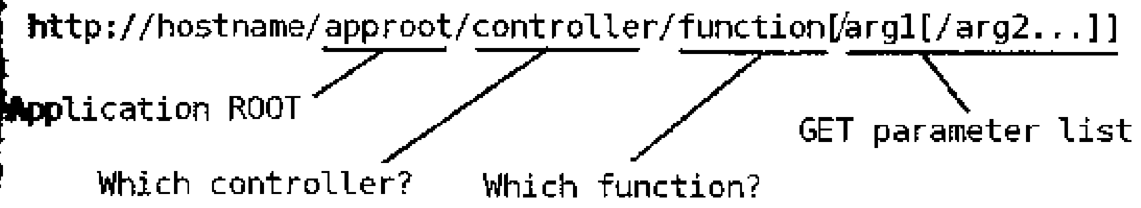
View renders the model into a form suitable for interaction, typically a user interface element. le views can exist for a single model for different purposes. A view port typically has a one to one spondence with a display surface and knows how to render to it.

Controller receives user input and initiates a response by making calls on model objects. A ler accepts input from the user and instructs the model and a view port to perform actions based input.

cation based on the MVC pattern may be a collection of model–view–controller triads, each sible for a different user interface element. The Swing GUI system, for example, models almost rface components as individual MVC systems. MVC is often seen in web applications where the s the HTML or XHTML generated by the application. The controller receives GET or POST input cides what to do with it, handing over to domain objects (i.e. the model) that contain the business and know how to carry out specific tasks such as processing a new subscription, and which hand ed to (X)HTML-generating components such as templating engines, XML pipelines, Ajax cks, etc. The model is not necessarily a database. In MVC, the *Model* is both the data and the ess logic needed to manipulate the data in the application. Many applications use a persistent ge mechanism such as a database to store data. MVC does not specifically mention the data access because it is understood to be underneath or encapsulated by the model. Models are not data access ts; however, in very simple applications that have little domain logic there is no real distinction to de. Active record is an accepted design pattern that merges domain logic and data access code; a l which knows how to persist itself.

5.2. System Architecture

plement MVC in our system, we have made use of the CodeIgniter PHP framework. This open-e framework allows us to create different controllers, models and views in PHP and automatically ates them into one seamless framework. The URLs are re-written with the help of Apache's CCESS file and mapped to the correct controller. The requests are classified as follows:



GET parameter list is optional. If the defined function expects a parameter but the URL doesn't ide it, the CI engine automatically substitutes default values in their places. The afore-mentioned CCESS file also disallows the clients to access any resource other than the index.php file. The s controllers are described below.

5.3. Controller Descriptions

Accounts Controller

accounts controller handles all user account related requests. All changes that pertain to an user's unt are automatically redirected to here. The different functions are as follows:

1. Accounts::login() – the login page which accepts the email and password via POST.
2. Accounts::register() – the registrations page for new users of the system

3. Accounts::view(uid) – views the profile of the user signified by uid
4. Accounts::edit() – allows the logged in user to edit his profile details
5. Accounts::browse() – provides a listing of the registered users
6. Accounts::cv(uid) – downloads the CV of the user signified by uid
7. Accounts::reviewerapply() – makes a reviewer application for the logged in user
8. Accounts::editorapply() – makes an editor application for the logged in user
9. Accounts::activate(code) – activates a pre-registered account from emailed code

Submissions Controller

controller handles all submission related requests. The functions are as follows:

1. Submissions::manage() – overview of the logged in user's submissions
2. Submissions::view(subid) – views details of a particular submission subid
3. Submissions::edit(subid,cmd) – edits the cmd section of a submission subid
4. Submissions::create() – creates a new submission
5. Submissions::pdf(subid) – downloads a submission

Reviews Controller

controller handles all review and reviewer-related requests. The functions are as follows:

1. Reviews::manage() – overview of all reviews assigned to the logged in user
2. Reviews::view(subid) – views a submission subid

Editorial Controller

controller handles all editorial and sub-editorial requests. The functions are as follows:

1. Editorial::forthcoming() – manages all forthcoming issues
2. Editorial::published() – manages all published issues
3. Editorial::confirmpublished(volume, issue) – confirm before publishing online
4. Editorial::createissue() – creates a new issue
5. Editorial::edit(volume, issue) – edits the details of a forthcoming issue
6. Editorial::manageissue(volume,issue) – manages submissions towards an issue
7. Editorial::managesubmission(subid) – manages a submission subid
8. Editorial::applications() – views outstanding applications
9. Editorial::accept() – confirm before acceptance of a submission
10. Editorial::reject() – confirm before rejection of a submission
11. Editorial::acceptapplication() – confirm before acceptance of an application
12. Editorial::rejectapplication() – confirm before rejection of an application
13. Editorial::managereviewers() – allows assignment of reviewers to submissions
14. Editorial::deletereviewer() – unassigns a reviewer from a submission
15. Editorial::searchreviewer() – searches potential reviewers

Archives Controller

handles all archive related requests. The functions are as follows:

1. Archives::browse() – browse through all volumes.
2. Archives::issue(volume, issue) – view articles in an issue

Main Controller

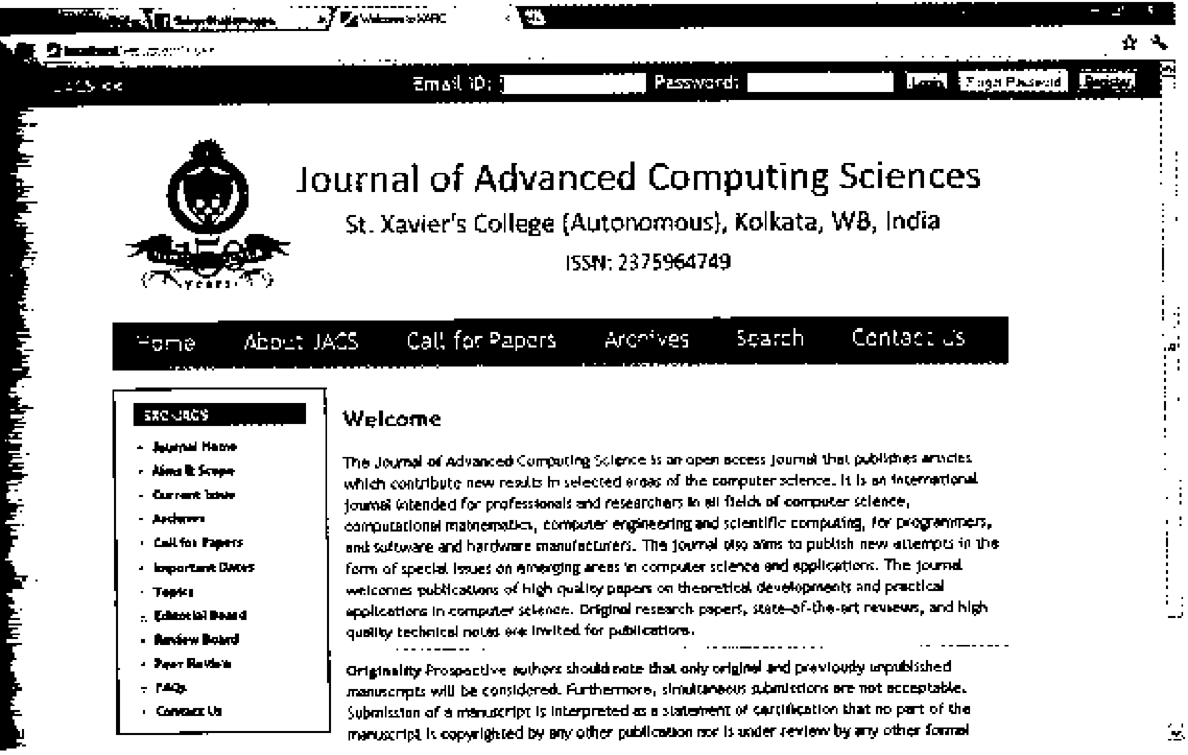
is the home controller of sorts. The functions are as follows:

1. Main::index() – HOME page
2. Main::page(name) – retrieves a static informational page
3. Main::search() – the search module

5.4. User Interface Design

user interface has been developed from scratch by hand by the developers. The UI is based on the gs design framework on a 12-column grid. The top bar is fixed and contains the login and tion details. The top section of the page consists of a banner. The navigational menu consists of hyperlinks (Home, About JACS, Call for Papers, Archives, Search, Contact Us). The context menu the left changes depending on logged in status. If we're not logged in, it displays only the basic links. erwise, it displays other links based on the user's privileges. The right side consists of the main body e the main dynamic content is displayed. The footer also contains some static links (not depicted). UI has been designed with HTML, CSS and minimal JavaScript and has been tested successfully cross-compatibility across all major browsers (including the quirky IE 6).

Screenshot of the homepage (main::index)



Journal of Advanced Computing Sciences

St. Xavier's College (Autonomous), Kolkata, WB, India

ISSN: 2375964749

Welcome

The Journal of Advanced Computing Science is an open access journal that publishes articles which contribute new results in selected areas of the computer science. It is an international journal intended for professionals and researchers in all fields of computer science, computational mathematics, computer engineering and scientific computing, for programmers, and software and hardware manufacturers. The journal also aims to publish new attempts in the form of special issues on emerging areas in computer science and applications. The journal welcomes publications of high quality papers on theoretical developments and practical applications in computer science. Original research papers, state-of-the-art reviews, and high quality technical notes are invited for publications.

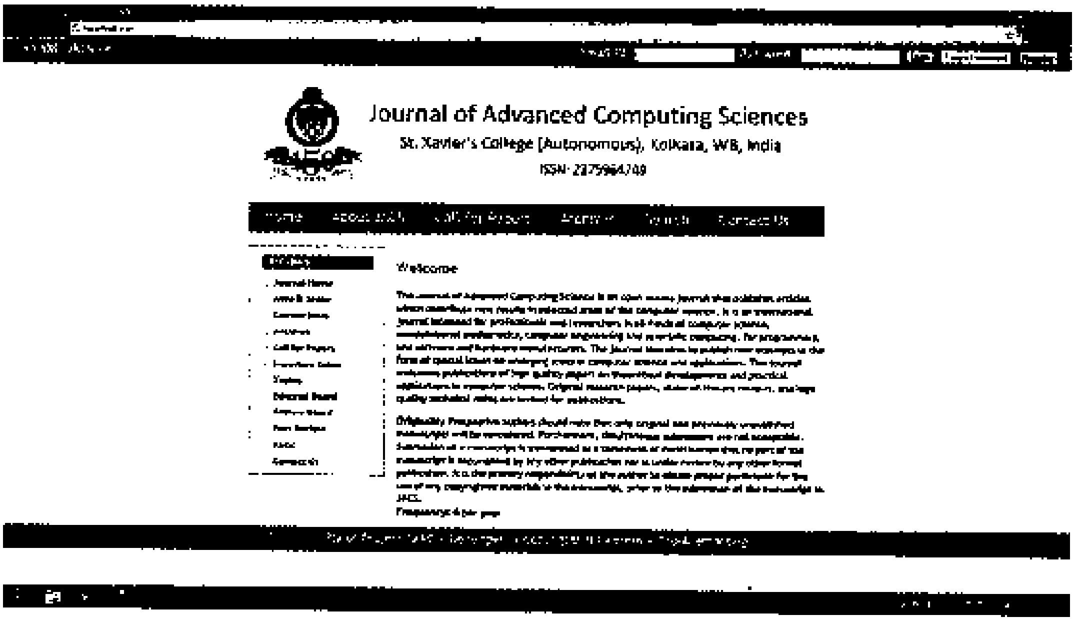
Originality Prospective authors should note that only original and previously unpublished manuscripts will be considered. Furthermore, simultaneous submissions are not acceptable. Submission of a manuscript is interpreted as a statement of certification that no part of the manuscript is copyrighted by any other publication nor is under review by any other formal

System Implementation

6.1. System Entry: index.php

```
default_timezone_set('Asia/Kolkata');  // change to your country/city !

 5.3 will complain without this
(ini_get('date.timezone') == '')

 date_default_timezone_set('GMT');


 define('ENVIRONMENT', 'development');

 if (defined('ENVIRONMENT'))
 {
        switch (ENVIRONMENT)
        {
                case 'development':
                        error_reporting(E_ALL);
                break;

                case 'testing':
                case 'production':
                        error_reporting(0);
                break;

                default:
                        exit('The application environment is not set correctly.');
        }
 }

 $system_path = 'system';
 $application_folder = 'system';

 if (defined('STDIN'))
 {
        chdir(dirname(__FILE__));
 }

 if (realpath($system_path) !== FALSE)
 {
        $system_path = realpath($system_path).'/';
 }

 $system_path = rtrim($system_path, '/').'/';
 if ( ! is_dir($system_path))
 {
        exit("Your system folder path does not appear to be set correctly. Please open the
 file and correct this: ".pathinfo(__FILE__, PATHINFO_BASENAME));
 }

 define('SELF', pathinfo(__FILE__, PATHINFO_BASENAME));
 define('EXT', '.php');
 define('BASEPATH', str_replace("\\", "/", $system_path));
 define('FCPATH', str_replace(SELF, '', __FILE__));
 define('SYSDIR', trim(strrchr(trim(BASEPATH, '/'), '/'), '/'));

 if (is_dir($application_folder))
 {
        define('APPPATH', $application_folder.'/');
 }
 else
 {
        if ( ! is_dir(BASEPATH.$application_folder.'/'))
        {
                exit("Your application folder path does not appear to be set correctly. Please
 following file and correct this: ".SELF);
        }

        define('APPPATH', BASEPATH.$application_folder.'/');
```

6.6. Miscellaneous



Journal of Advanced Computing Sciences

St. Xavier's College [Autonomous], Kolkata, WB, India

ISSN: 2375964749

Home | About JACS | Call for Papers | Archive | Search | Contact Us

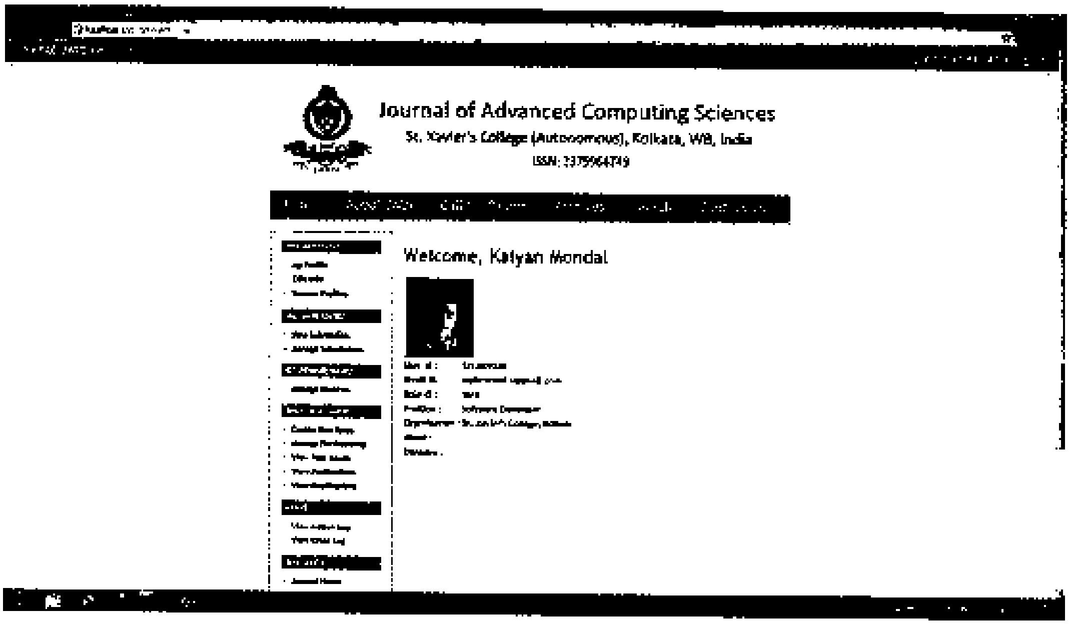
Welcome

[illegible]

[illegible]

Frequency: 4 per year

Figure: Home page



Journal of Advanced Computing Sciences

St. Xavier's College [Autonomous], Kolkata, WB, India

ISSN: 2375964749

Welcome, Kalyan Mondal

[illegible]

Figure: View Profile

. Conclusion

nclusion, we reflect that the world is moving fast towards the complete digitization of such ls. With all the advantages of e-journals, it becomes an extremely profitable option to switch over ournals. E-journals save the expense and the time required for the tedious publications of printed ls. In this paper, we have outlined some of the generalized processes that an e-journal gement system should perform and the functionalities they should have. With even more advances ormation technology it can be predicted that such tasks will become more automated. With the t of the paradigms of cloud computing, this application of information technology in e-journal gement systems can be expected to illuminate new vistas of innovation.

8. Acknowledgements

thank the entire faculty of the Department of Computer Science for their support and uragement, especially our Head of the Department, Prof. Anal Acharya. We thank our supervisor Dr. Asoke Nath for his kind supervision and warm support. We thank Prof. Shalabh Agarwal for dvice and guidance on technical issues of this project. We thank both Prof. Nath and Prof. Agarwal heir inspiration to complete this extended work and to publish two papers on the project at national conferences. We thank our peers and our juniors for their encouragement and support, cially Mr. Mourjo Sen, for his help on the editing of our research papers on this project. Last but not east, we thank the entire lab staff, especially Mr. T. Karmakar, Mr. S. Chandra and Mr. A. akar, who endured our prolonged hours of tiring lab work for months on end. Without the support l the above people, this project would not have been possible. And therefore, we remain deeply ted to them.

9. References & Bibliography

oger S. Pressman. "Software Engineering, A practitioner's Approach". Tata McGraw Hill Higher ducation. 2011.

ames A. Senn. "Analysis and Design of Information System". Tata M G Hill Higher Education. 2011.

teve Holzner. "Complete Reference PHP 5.2". Dreamtech. 2010.

teve Holzner. "HTML Black Book". Dreamtech. 2009.

teve Holzner. "Web Technologies Black Book". Dreamtech. 2010.

ox. "Reference – PHP, MySQL, Apache Server 2.0". O'Reilley. 2010.

obert Sheldon. "SQL: A Beginner's Guide". McGraw Hill Professional. 2011.

aul Wilton and John Colby. "Beginning SQL (Programmer to Programmer)". Wreck. 2005.

. J. Date. "Database System". Narosa Book Distributors Pvt Ltd. 2002.

orth. "Relational Database Management System Vol. -4". McGraw Hill. 2009.

ary E. S. Loomis. "Data Management and File Structure". Englewood Cliffs, N.J. :[s.n.]. 2007.

oger S. Pressman. "Software Engineering: A Practitioner's Approach". McGraw Hill. 2009.

n Sommerville. "Software Engineering, 8e, 2005". Pearson. 2005.

ocalhost ▸ xavnetso_xaro

tionloginfo

mn	Type	Null	Default	Comments
	bigint(20)	No		
	varchar(100)	No		
ctID	bigint(20)	No		
n	text	No		
	varchar(15)	Yes	*NULL*	
[illegible]	text	No		
	text	Yes	*NULL*	

plicationinfo

mn	Type	Null	Default	Comments
rID	bigint(20)	No		
	enum('reviewer', 'editor')	No		
	bigint(20)	Yes	*NULL*	

thorinfo

umn	Type	Null	Default	Comments
D	bigint(20)	No		
rID	bigint(20)	No		

sessionsinfo

umn	Type	Null	Default	Comments
sion_id	varchar(40)	No	0	
ddress	varchar(16)	No	0	
r_agent	varchar(50)	No		
_activity	bigint(20)	No	0	
r_data	text	Yes	*NULL*	

torloginfo

umn	Type	Null	Default	Comments