

Development Of Journal Management Web Application, Case Study Of Journal Of Research And Innovations In Engineering, University Of Uyo

Kufre Effiong Okon¹

Department Of Electrical/Electronic and Computer Engineering
University of Uyo, Akwa Ibom State Nigeria

Ozuomba Simeon²

Department Of Electrical/Electronic And Computer Engineering,
University of Uyo, Akwa Ibom State Nigeria
simeonozuomba@uniuyo.edu.ng

Philip M. Asuquo³

Department Of Electrical/Electronic And Computer Engineering,
University of Uyo, Akwa Ibom State Nigeria
philipasuquo@uniuyo.edu.ng

Abstract— In this paper, the development of journal management web application, case study of Journal of Research and Innovations in Engineering (JORIE) is presented. The software development was done using iterative-incremental model. Also, the web application was based on the three tier architecture while the backend was implemented using the model-view-controller (MVC) architecture which separate the system into 3 main logical components, namely; view, controller and model. Functional decomposition of the system consists of two sub-systems, namely: the journal website and the journal management dashboard (administrative backend). The website enables users to view, read online, search and download published articles and also access journal information such as journal policies, author's instructions, editorial board and journal discussion forum. The journal management dashboard enables effective management of the entire editorial workflows which includes manuscript review by the editor-in-chief upon submission, manuscript assignments to editors and subsequently reviewers, peer-review by the reviewers, reviewer recommendation review by the editor, editorial decision (accept/reject) and publication. The system was implemented using ASP.Net Core 3.1 framework, along with VueJs framework, Microsoft Server (MSSQL) database and was hosted on Internet Information Service (IIS) server. The results of the implemented system are presented with screenshots of requisite web pages and accompanying discussion of the results.

Keywords— *Three Tier Architecture, Journal Management, Iterative-Incremental Development Model, Functional Decomposition, Web Application*

1. Introduction

As many academic institutions are seeking for global relevance and how to move up or sustain their global ranking, institutional journal publishing out becomes a ready tool to achieve some of those objectives. Researchers across the globe prefer academic institutional journal publishing outs for disseminating their research outputs. However, such journal publishing outfit will not be able to attract researcher as it lacks global visibility. In that case, well-developed journal management application is needed for the institutional journal.

Accordingly, in this paper, the development of journal management web application for a case study Journal of Research and Innovations in Engineering (JORIE), University of Uyo is presented. The software development methodology, architecture, functional decomposition of the system and other design details of the web application are presented [1,2,3,4]. The journal web application design is based on the features extracted from notable journal management web platforms with track record of global best practices. Also, some notable peculiarities of the case study JORIE publishing platform are also noted and accommodated in the design. The system was implemented and hosted locally and tested with information from the case study JORIE team in University of Uyo.

2.0 Methodology

2.1 The software development methodology

The focus of this study is to develop journal management web application for case study journal of research and innovations in engineering, (JORIE) mounted by the faculty of Engineering in University of Uyo. An incremental but also iterative methodology (Figure 1) was used for the development of the web application..

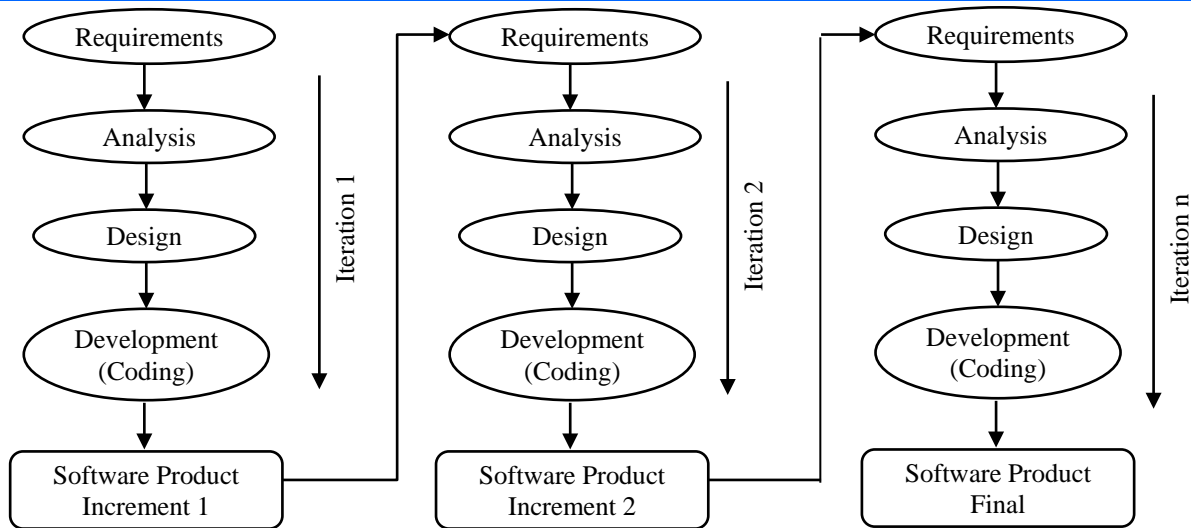


Figure 1 Iterative-Incremental Development Model

2.2 System Design

The system was designed using the three tier architecture, as shown in Figure 2 [5,6,7]. The backend is implemented using the model-view-controller (MVC) architecture which separate the system into 3 main logical components – View, Controller and Model as shown in Figure 3 [8,9,10].

Functional decomposition of the system: The system incorporates two sub-systems thus: the journal website and the journal management dashboard (administrative backend). The website enables users to view, read online, search and download published articles and also access journal information such as journal policies, author’s instructions, editorial board and journal discussion forum. The journal management dashboard enables effective management of the entire editorial workflows which

includes manuscript review by the editor-in-chief upon submission, manuscript assignments to editors and subsequently reviewers, peer-review by the reviewers, reviewer recommendation review by the editor, editorial decision (accept/reject) and publication. The sub-systems are shown in Figure 4. The journal website was decomposed into two modules - the reader module and the forum module as shown in Figure 5. Similarly, the journal dashboard was decomposed into authentication, notification, admin, and editorial modules. The editorial module is further split into three submodules - author, reviewer and editor modules as shown in Figure 6. In the subsequent section, detailed design of the following modules are presented; user authentication module, author module, reviewer module and editor module.

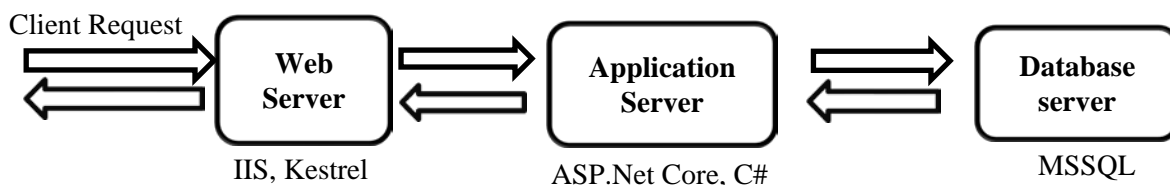


Figure 2: The Three Tier Architecture of the System

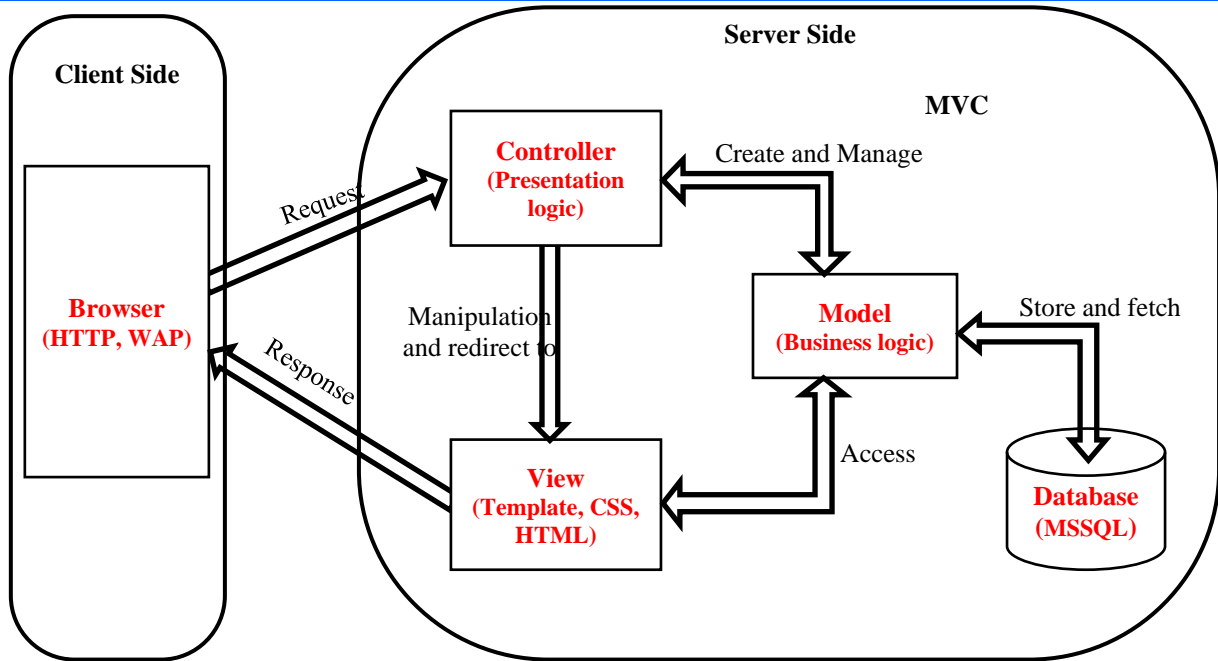


Figure 3 System MVC Architecture

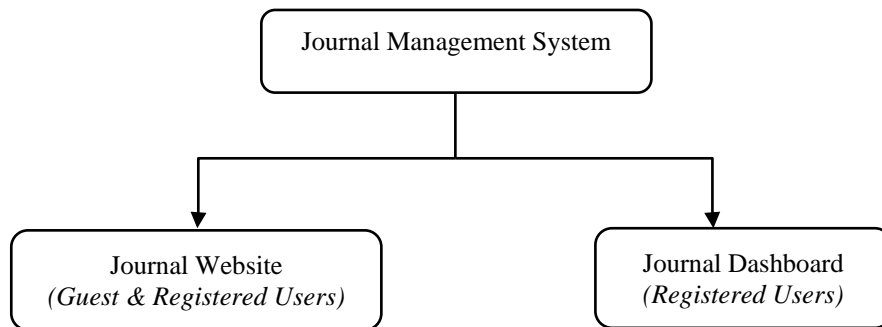


Figure 4: Journal management sub-systems

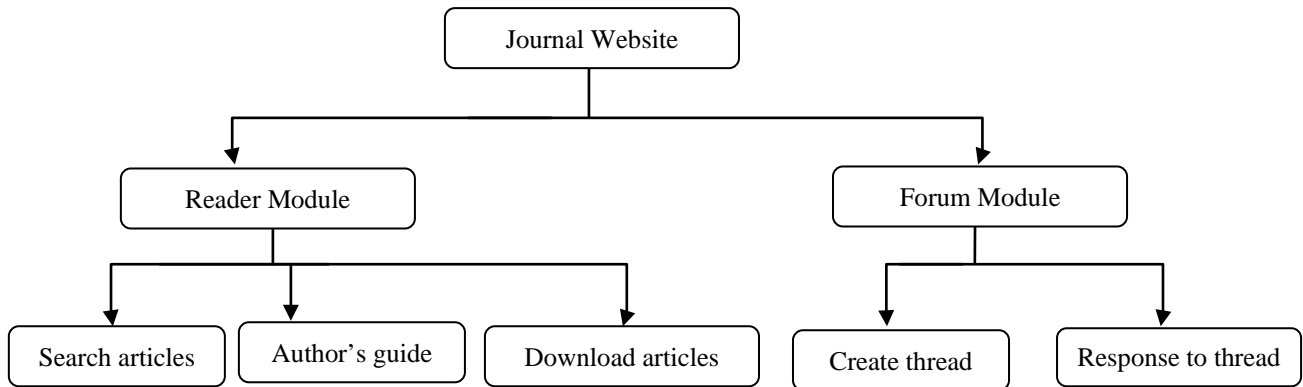
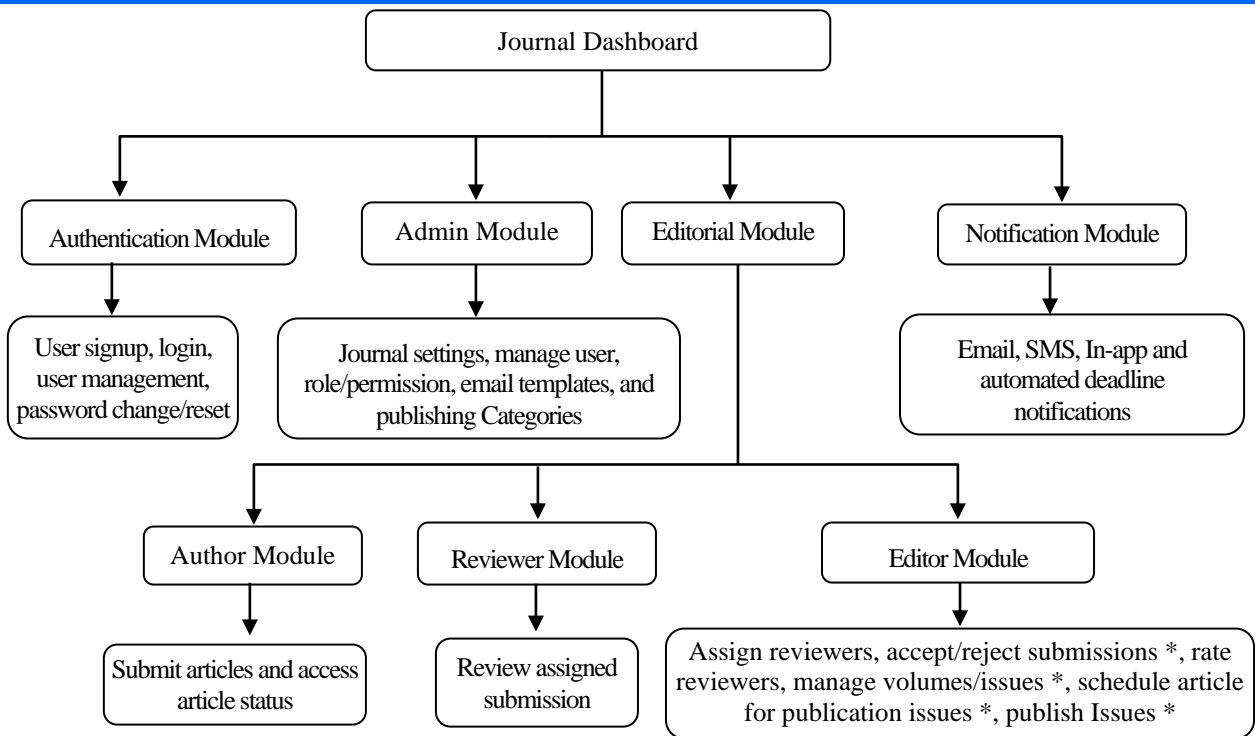


Figure 5 Journal website functional decomposition



* *permission-controlled activities*

Figure 6 Journal dashboard functional decomposition

2.3 User Authentication Module

The user authentication module (Figure 7) handles all authentication related activities on both the journal website and the journal dashboard. Authentication is required to access the discussion forum as well as the journal dashboard. The module is built on top the ASP.Net Core identity API that supports user interface (UI) login functionality and manages users, passwords, profile data, roles, claims, tokens, email confirmation, and more.

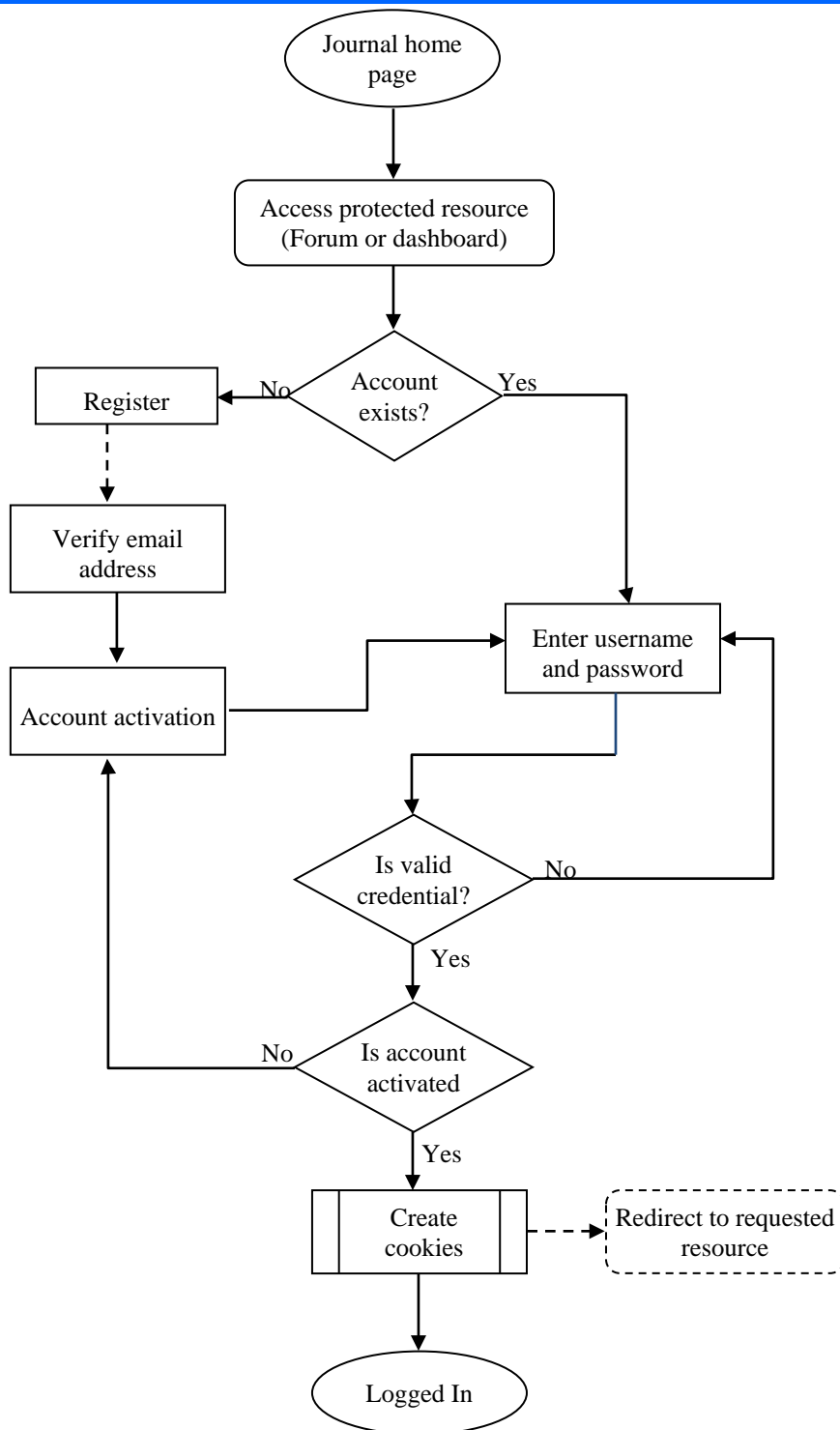


Figure 7 User authentication flowchart

2.4 Author Module

In this module, when users register on the system, they automatically assume an author role, and a verification email is sent to their email address so as to make sure the email account is working. Next, authors will have access to their personal and a lightweight dashboard. The author dashboard offers several functions; the most important one being paper submission process.

In this process the authors are able to fill the required fields and upload their article to the system for immediate submission or save for later editing and submission to the editor-in-chief. The submission workflow is depicted in Figure . Once the submission process is completed, the notification system will notify the authors. acknowledging their submission also, a secured deep link is generated and send to the author. With the deep-link, the authors can access their submission status without having to go through the login procedures. The deep-link is designed in such a way that the user is prevented from accessing any other

resource aside the one the deep-link was generated for; this is a security measure to reduce the risk of account hijacking

or disclosure of vital information when unintended user by any means has access to the deep-link.

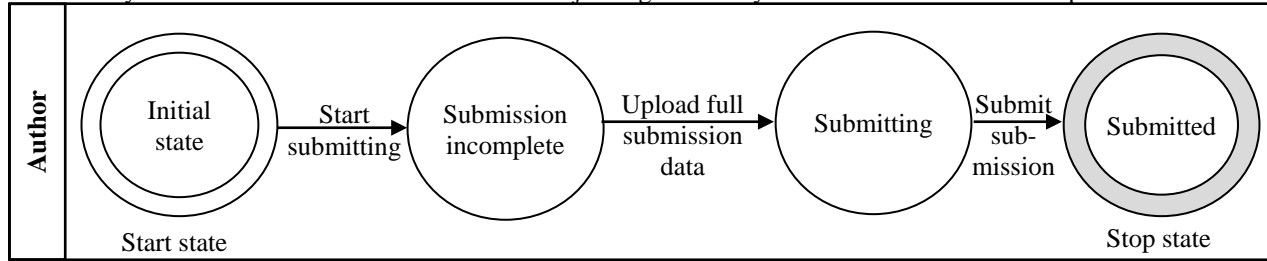


Figure 8 Submission process workflow

2.5 Reviewer Module

This module enables reviewing and giving comments/recommendation to an assigned manuscript. In order to become a reviewer, a normal registration as authors is needed, then the user is required to apply as a reviewer, upon confirmation by the editor-in-chief, the reviewer is then presented with their private dashboard. After receiving a review request, the reviewer needs to decide to accept or decline the request within the dateline. This can be done

through the deep-link generated when the manuscript was assigned or through the reviewer dashboard. Upon accepting the review request, the reviewer is presented with another interface for recording his/her comments and option for recommendation. The process is designed in such a way that the reviewer may choose to save his/her review to be completed at later date. The flowchart for the reviewer module is shown in Figure .

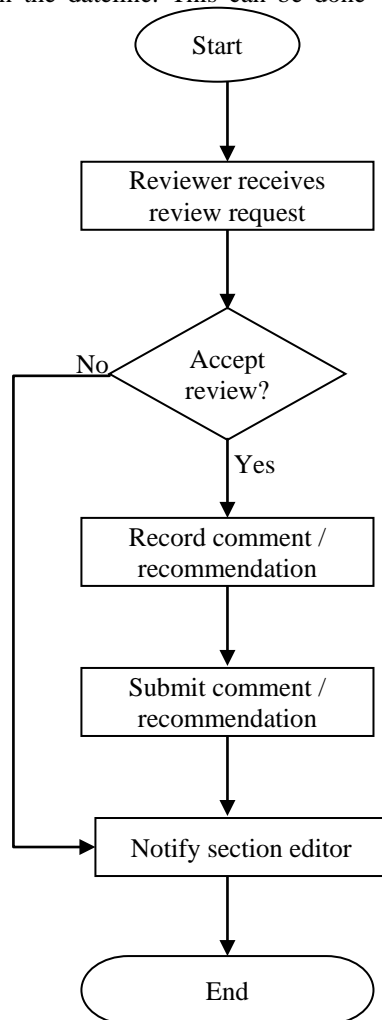


Figure 9 Reviewer process flowchart

2.6 Editor Module

This module is specifically designed for editors and the editor-in-chief to manage the journal. Editors need to register on the system similarly to reviewers, and they have to be approved by the editor-in-chief. Much the same as

other two modules, editors have their own dashboard but with a wider range of functionalities. The functionalities are a bit restricted depending on the role of the logged in user either editor-in-chief or section editor. The design of this module allows only the editor-in-chief to receives new

submission notifications and also user applications to join the editorial team either as an editor or as a reviewer while the section editors receive notification when author submit revisions of their original submission. On new submission received, the editor-in-chief performs preliminary review of the submission then assign/reassign it to an editor if found to be in accordance with the submission guidelines or outright rejection. The section editor also reviews the submission before assigning it to two reviewers for the peer review process. On receiving notification of the completed peer-review, the section editor reviews the reviewers' comment / recommendation. In an event where the per-review is inconclusive or revision recommended by the reviewers, the section editor then send the manuscript back to the author along with the reviewers' comment. Finally, the section sends editorial recommendation (accept/reject) to the editor-in-chief. Depending on the discretion of the editor-in-chief, section editors may be given privilege to give final verdict on the submission; this is designed intentionally to lessen the workload on the editor-in-chief.

The design also makes it possible and easier for the section editor to cancel review request sent to any of the reviewers and most importantly functionality for reviewer rating and thanking the reviewer for their effort.

Once a manuscript is accepted or rejected by the editor-in-chief or the section editor, notification is send to the author. In the case of accepted submission, the author is requested to upload the final copy for publication, and once this is done, it can be schedule for publication in either a back issue or a further issue, this action is controlled by permission which can be given to the section editor or remain exclusive to the editor-in-chief. Publication of issues is also handled by the editor-in-chief or any other privileged user, here table of contents which is generated automatically as manuscripts are scheduled for publication can be edited as required and the final copies submitted by the authors which are word documents are automatically converted to pdf by the system for easier downloading by the readers. The publication process is depicted in Figure 10.

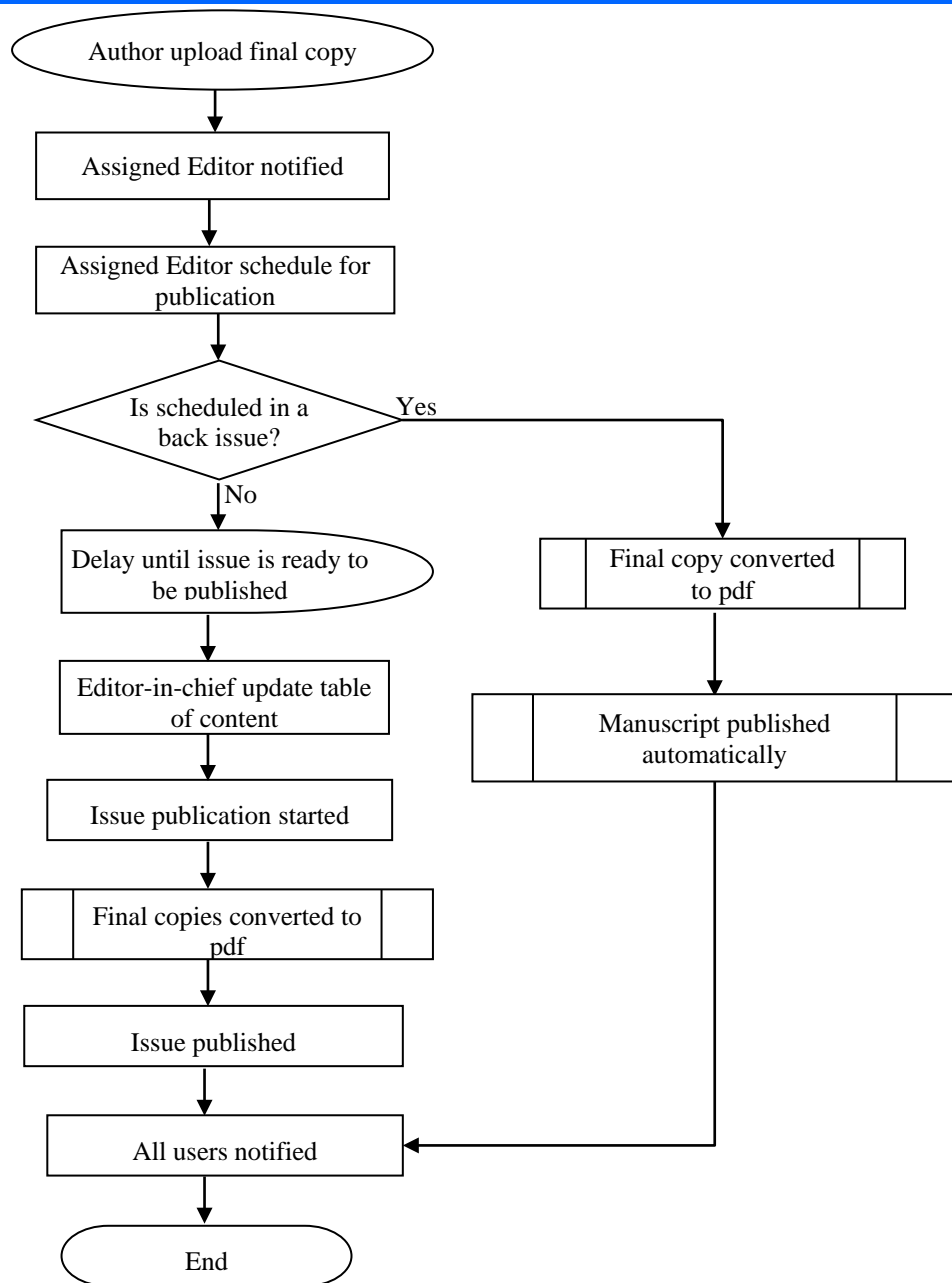


Figure 10 Manuscript publication flow diagram

3.0 Result of the Implementation of the Journal System

The results of the system are represented with screenshots according to the two subsystems – journal website and the journal dashboard in subsequent sections.

3.1 Results on the Journal Website

The website serves as the entry point to the journal management system. The website homepage is shown in Figure 11. The website enables users (both registered and unregistered alike) to view, read online, search and download published articles of interest.

JORIE | ISSN: 123-456-789

Enter keywords, authors etc...

Register Login

JORIE Home About Editorial Board For Authors Policies Achives Forum Contact

Quick Links

- Submit to JORIE
- Instruction for authors
- Current Issue
- Browse list of issues
- Most Downloaded Articles
- Most Viewed Articles
- Publication Fee
- Reviewers
- Peer Review Process

Latest Issues

Volume 1 Issue 1
(October 2021)

Call for papers

Submissions Welcome for JORIE

Journal of Research and Innovations in Engineering (JORIE)

Journal of Research and innovations in Engineering (JORIE) is a publication of the Faculty of Engineering, University of Uyo, Akwa Ibom State, Nigeria. The journal is published twice a year and accepts original articles of scientific research work. Technical notes and reviews in engineering technology and applied sciences. The journal has an internationally recognized editorial board with extensive academic qualifications. The journal has a high standard with an extensive international coverage and acceptance.

ISSN : 123-456-789

Publisher : Faculty of Engineering, University of Uyo

Journal Abbreviation: JORIE

Top Ranked Articles

Research Article
Development of an MVC based Journal management system case study: Journal of Research and Innovations in Engineering
Samuel okon
Pages: 23-27 | Views: 26 | Downloads: 7

Volume 1 Issue 1

Copyright © 2021 JORIE - Journal of Research and Innovations in Engineering

About Contact Privacy

Figure 11: Journal website homepage

User must register on the system and login in order to have access to the discussion forums on the website and the dashboards. The registration process is same for all user groups and requires firstname, lastname, email, username and password as compulsory fields as shown in Figure . After successful registration, an email confirmation is sent to the registered email address where is user is required to

click to confirm and activate his/her account before user can login to the system. After login, the user can apply to become a reviewer or an editor. Figure shows the login page, it requires either the username or the email (designed intentionally to offer more flexibility) and password to login.

JORIE

Register a new membership

Firstname

Middlename

Lastname

Email

Username

Password

Retype Password

Register

[I already have a membership](#)

Figure 12: User registration form

Figure 13: User login form

3.2 The results on the Journal Dashboard

The dashboard is the administrative end of the journal management system that handles user profile management, editorial workflows – submission, peer-review and publication, administrative functions such as user management, role and permission management, email templates, volumes/issues, article types/categories, journal information such as contact, journal policies, author's information, review guidelines and other settings.

3.2.1 Results for the Authors Module

After successful registration, the user automatically assumes the author's role and has access to the author dashboard – a lightweight dashboard providing the basic statistics and functionality for article submission and tracking. Figure shows the author dashboard. Authors as well as other registered users have the provision of changing their password, accessing their profile and notifications generated by the system. Authors can view their previous submissions and statuses at a glance by navigating to the manuscript page as shown in Figure .

Figure 14: Author dashboard

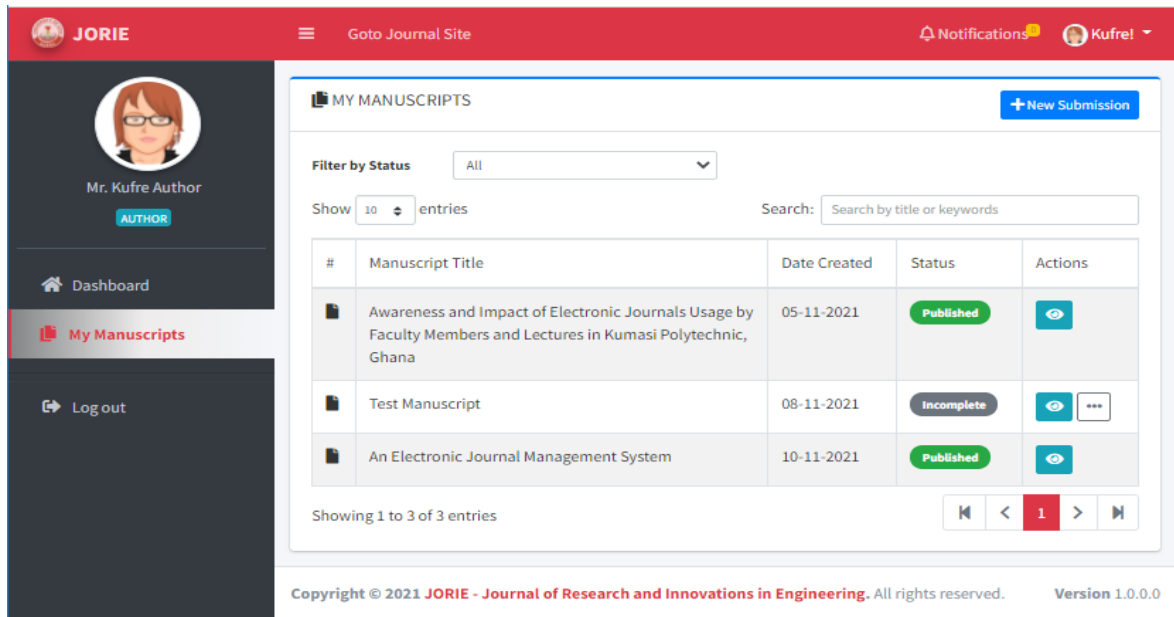


Figure 15: Author's manuscript listing page

3.2.2 Result for the Editor Module

In order to become an editor, an author will need to submit an application through the system. This option is only available on the author dashboard which is the default dashboard after registration. Figure shows the editor application form. The application form contains extra

information necessary to enable the editor-in-chief approve or reject such application. Once the application is submitted, the system automatically notifies the editor-in-chief through who then act on the application and the status communicated back to the applicant. If approves, the author is moved automatically to the editor role.

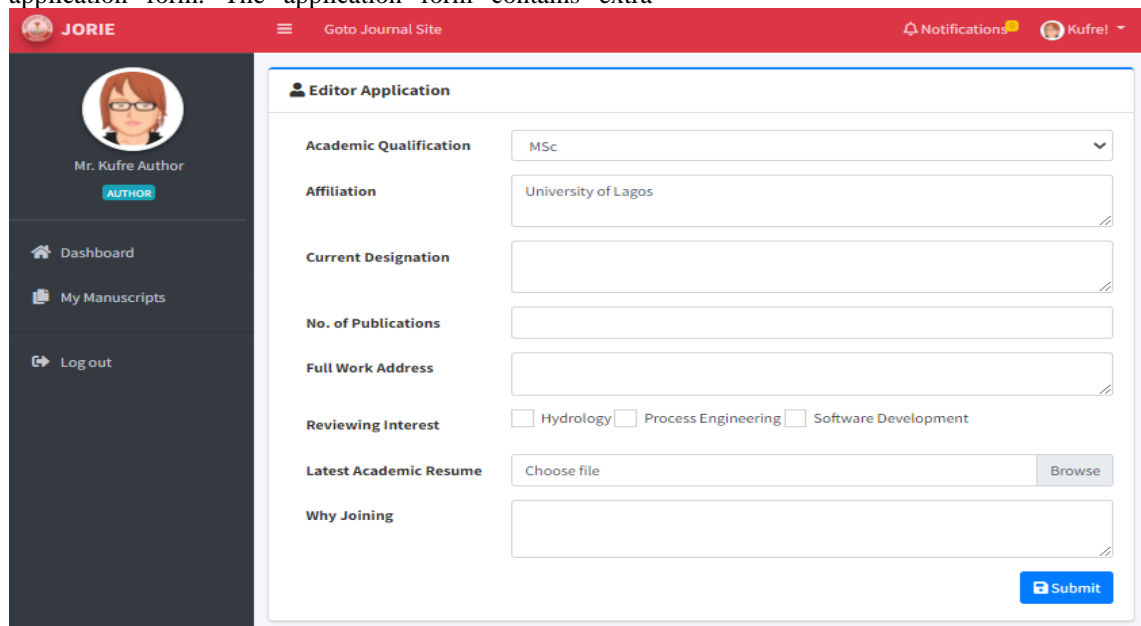


Figure 16 Editor application form

The editor dashboard is shown in Figure . The menu items are controlled by permissions hence editor-in-

chief will have more access which translate to more menu items than section editors.

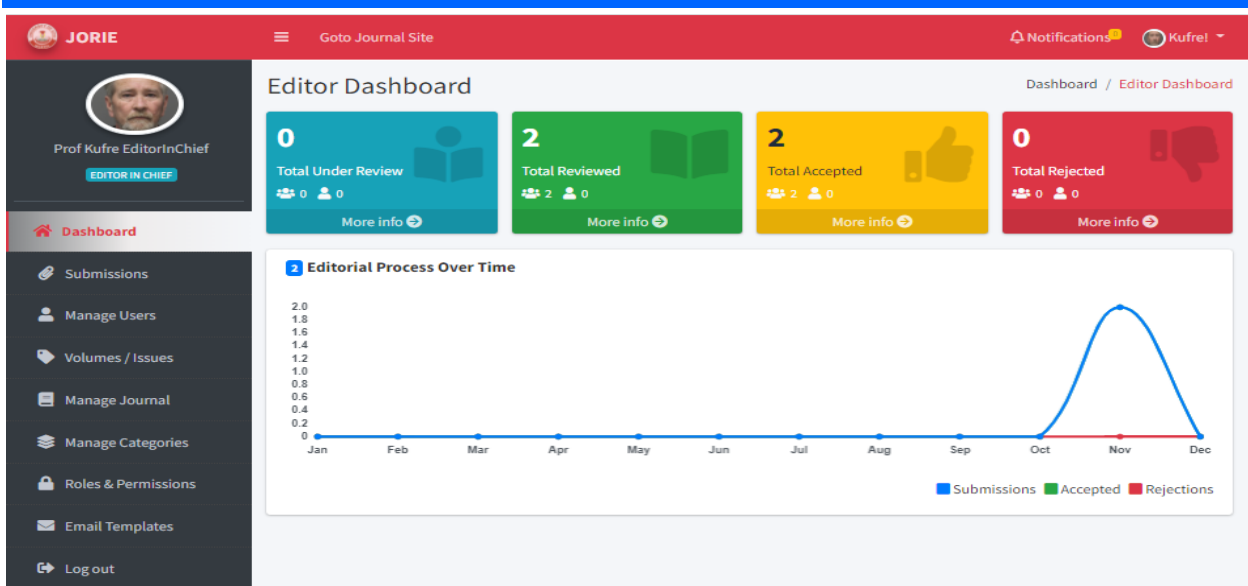


Figure 17: Editor dashboard

Once an author submit manuscript successfully, the editor-in-chief is notified through email, SMS and in-app notification informing of the new submission. The

editor-in-chief then reviews it through the editor submission page. The submission has four main sections – my queue, unassigned, all active and archives as shown in Figure .

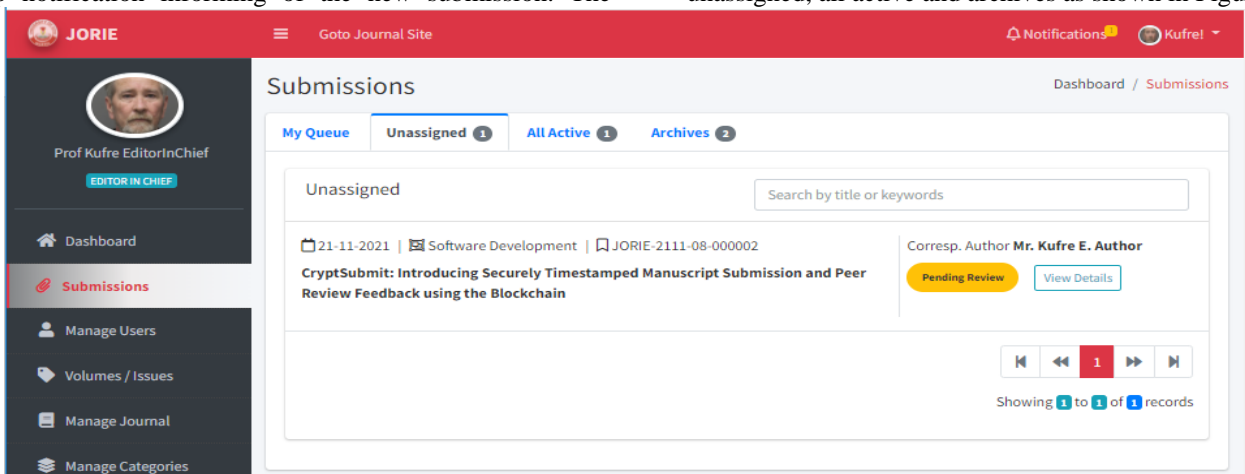


Figure 18: Editor submission page

Figure 19: Section editor assignment form

Submissions assigned to the current editor is listed under 'My Queue', 'Unassigned' is accessible only by the editor-in-chief, it is where new submissions are queued for initial review and subsequent assignment to the section editors. 'All Active' list submissions currently undergoing

review by the section editors and reviewers. 'Archives' list all submissions published or delined. If the manuscript passes through the submission criteria, it is then assigned to a section editor to oversee the editorial process as shown in Figure .

3.2.3 Result for the Reviewer Module

Just like the editor user, authors must submit application through the system to be elevated to the reviewer role as shown in Figure . Once this has been approved by the editor-in-chief, then the reviewer is

notified and can access the reviewer dashboard shown in Figure . From the dashboard, the reviewer can navigate to the submission page shown in Figure .

Figure 20: Reviewer application form

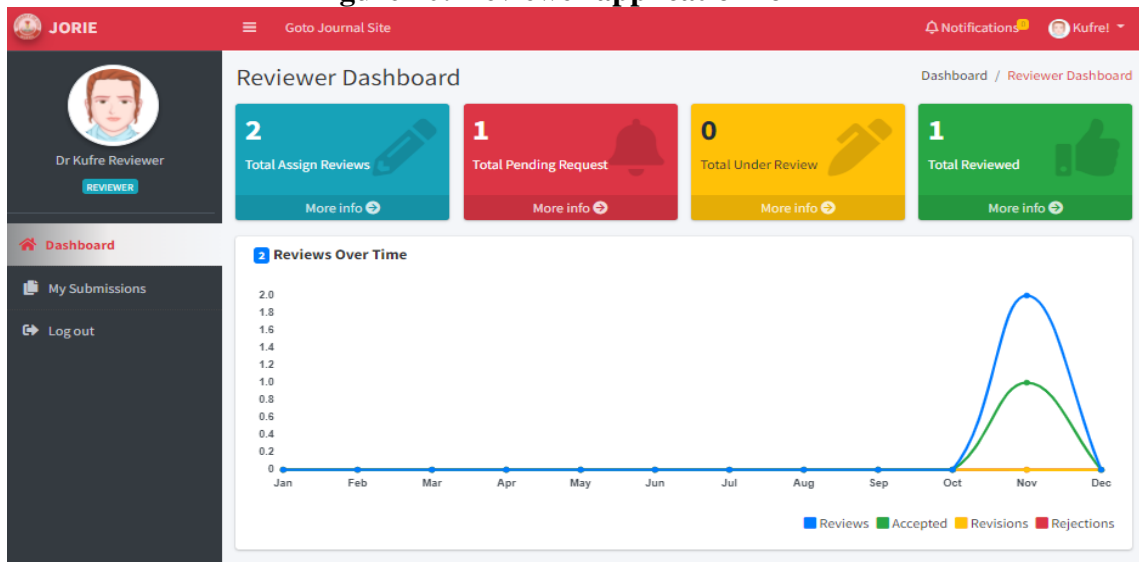


Figure 21: Reviewer dashboard

Figure 22: Reviewer submission page

4. Conclusion

The work presented in this study is the development of journal management web application for case study journal of research and innovations in engineering, (JORIE) mounted by the faculty of Engineering in University of Uyo. An incremental but also iterative methodology (Figure) was used for the development of the web application. The web-based journal management system was designed for JORIE is based on the three tier architecture, and the backend is implemented using the model-view-controller (MVC) architecture which separate the system into 3 main logical components, namely; view, controller and model. Functional decomposition of the system is presented and detailed design of some of the key modules are resented and they include user authentication module, author module, reviewer module and editor module. The results of the implemented system are presented with screenshot of requisite web pages and accompanying discussion of the results.

References

1. Kahraman, H. T., Sagiroglu, S., & Colak, I. (2010, October). Development of adaptive and intelligent web-based educational systems. In *2010 4th international conference on application of information and communication technologies* (pp. 1-5). IEEE.
2. Krylovskiy, A., Jahn, M., & Patti, E. (2015, August). Designing a smart city internet of things platform with microservice architecture. In *2015 3rd international conference on future internet of things and cloud* (pp. 25-30). IEEE.
3. Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3), 8-13.
4. Eppinger, S. D., & Browning, T. R. (2012). *Design structure matrix methods and applications*. MIT press.
5. Grozev, N., & Buyya, R. (2014). Multi-cloud provisioning and load distribution for three-tier applications. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 9(3), 1-21.
6. Loiret, F., Seinturier, L., Duchien, L., & Servat, D. (2010). A Three-Tier Approach for Composition of Real-Time Embedded Software Stacks. In *Component-Based Software Engineering: 13th International Symposium, CBSE 2010, Prague, Czech Republic, June 23-25, 2010. Proceedings 13* (pp. 37-54). Springer Berlin Heidelberg.
7. Woods, E. (2016). Software architecture in a changing world. *IEEE Software*, 33(6), 94-97.
8. Jailia, M., Kumar, A., Agarwal, M., & Sinha, I. (2016, November). Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework. In *2016 International Conference on ICT in Business Industry & Government (ICTBIG)* (pp. 1-5). IEEE.
9. Singh, S., & Iyer, J. (2016). Comparative study of MVC (model view controller) architecture with respect to struts framework and PHP. *International Journal of Computer Science Engineering (IJCSE)*, 5(3), 142-150.
10. Sarker, I. H., & Apu, K. (2014). Mvc architecture driven design and implementation of java framework for developing desktop application. *International Journal of Hybrid Information Technology*, 7(5), 317-322.