Development of a Workflow Management System for Higher Education: Case study of a Department in a Nigerian University

*[1]Olufikayo A. Adedapo, [1]Jamilu Bello, [1]Mutiat A. Ogunrinde,
[2]Mayowa O. Oyediran, [2]Michael O. Adio and [1]Okikiola A. Oyekola
[1]Department of Mathematical and Computer Sciences, Fountain University, Osogbo, Nigeria
[2]Department of Computer Engineering, Ajayi Crowther University, Oyo, Nigeria
**{adedapo.fikayo|bellfulani26|ogunrinde.mutiat}@fuo.edu.ng|mo.oyediran@acu.edu.ng|
mo.adio@acu.edu.ng| kikidoesflutter@gmail.com**

Received: 27-OCT-2023; Reviewed: 25-NOV-2023; Accepted: 01-DEC-2023
http://doi.org/10.46792/fuoyejet.v8i4.1126

ORIGINAL RESEARCH

Abstract- The growing complexity of academic and administrative processes in educational institutions necessitates an efficient and streamlined approach to managing workflows effectively. The research work presents the development of a cutting-edge Workflow Management System (WMS) designed to cater to the needs of the Department of Mathematical and Computer Sciences. The WMS is built using Flutter, a popular cross-platform framework, and Firebase, a scalable and cloud-based backend service. These challenges underscore the importance of a tailored WMS that can centralize communication, automate tasks, and enhance productivity. This research project encompasses a systematic approach, beginning with requirements gathering through observation and prototyping, followed by the design and development phases. Agile development principles were employed to ensure flexibility and adaptability throughout the project life-cycle. The outcomes of the research work, including any improvements made based on user feedback. This research has significantly improved communication efficiency, task allocation, and time management. Users reported increased collaboration, reduced delays, and better organization of academic and administrative activities and the potential to improve the efficiency and accuracy of collaboration management within the department and enhance the user experience for both the collaboration department and the college community. The developed workflow Management System proved valuable for the department, addressing their needs and streamlining their workflows. The successful integration of Flutter and Firebase showcased the potential of modern technologies to revolutionize traditional processes in educational institutions.

Keywords- workflow management system, Firebase, task management, cloud storage, user authentication

1 INTRODUCTION

Using workflow management systems has become increasingly important in recent years as organizations strive to optimize their processes and improve efficiency. Universities are no exception, as they face the challenge of managing complex workflows related to student admissions, course scheduling, academic advising, research administration, and many other tasks. Existing workflow management systems often need to be tailored to the unique requirements of universities. They may need critical features like integration with existing academic systems and support for complex approval processes. As a result, there is a need for a specialized workflow management system that can meet the unique requirements of universities and improve the overall efficiency of their operations (Ayodeji & Olubiyi, 2020).

The research developed a workflow management system for higher institutions in Nigeria. The system was based on a thorough analysis of the existing workflows within universities, with input from stakeholders, including administrators, faculty members, and students.

*Corresponding Author

Section B- ELECTRICAL / COMPUTER ENGINEERING & RELATED SCIENCES
Can be cited as:
Adedapo O.A., Bello J., Ogunrinde M.A., Oyediran M.O., Adio M.O. and Oyekola O.A. (2023). Development of a Workflow Management System for Higher Education: Case study of a Department in a Nigerian University, FUOYE Journal of Engineering and Technology (FUOYEJET), 8(4), 456-462
http://doi.org/10.46792/fuoyejet.v8i4.1126

The system's creation aimed to support many workflows, from student admissions and course scheduling to research administration and grant proposal management. It will be user-friendly, with a web-based interface accessed from any device (Feldmann, Witzmann, and Maedche, 2018). It will incorporate automated notifications, real-time status updates, and customizable approval processes. By developing a specialized workflow management system for universities, this project aims to improve the efficiency of university operations and enhance the overall experience for students, college members, and staff. Furthermore, incorporating automated notifications and customizable approval processes can lead to faster turnaround times, ensuring that critical tasks, such as approvals for research proposals or course changes, are processed efficiently (Denecke & Kohler, 2020). This, in turn, can positively impact research timeliness and contribute to the overall academic growth of the institutions.

The significance of this research extends beyond just enhancing operational efficiency; it also aims to elevate the overall experience for stakeholders within the university community. The workflow management system creates a more conducive learning, research, and innovation environment by streamlining intricate processes and minimizing administrative obstacles. While there have been advances in workflow management systems in various sectors, the specialized nature of universities demands a unique and tailor-made solution. As such, this research was done to contribute to the growing body of knowledge surrounding workflow management systems and establish a framework that can

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

be adopted by other higher education institutions both within Nigeria and globally (Zhang et al., 2019).

On the other hand, developing a workflow management system for higher institutions in Nigeria is paramount to elevating universities' efficiency, transparency, and overall productivity. By understanding the intricacies of existing workflows and collaborating with key stakeholders, this research created a transformative tool that can positively impact all university operations, ultimately benefiting students, college members, administrators, and staff alike. Through continuous innovation and improvement, this specialized system has the potential to set new standards in the realm of higher education workflow management, empowering universities to thrive in an ever-evolving academic landscape.

2 Related Works

In their work, Zhu *et al.,* 2016 highlight such difficulties and model the workflow scheduling problem, which optimizes both make span and cost as a Multi-objective Optimization Problem (MOP) for the Cloud environments. They propose an Evolutionary Multi-Objective Optimization (EMO) algorithm to solve this workflow scheduling problem on an infrastructure as a service (IaaS) platform. Novel schemes for problem-specific encoding and population initialization, fitness evaluation and genetic operators are proposed in this algorithm. Extensive experiments on real-world workflows and randomly generated workflows show that the schedules produced by our evolutionary algorithm present more stability on most of the workflows with the instance-based IaaS computing and pricing models. The results also show that the algorithm can achieve significantly better solutions than existing state-of-the-art QoS optimization scheduling algorithms in most cases. The conducted experiments are based on the on-demand instance types of Amazon's EC2; however, the proposed algorithm can easily be extended to the resources and pricing models of other IaaS services.

Chen *et al.* (2018) reported model cloud workflow scheduling as a multi-objective optimization problem that optimizes both execution time and execution cost. A novel multi-objective ant colony system based on co-evolutionary multiple populations for multiple objectives framework is proposed, which adopts two colonies to deal with these two objectives, respectively. Moreover, the proposed approach incorporates the following three novel designs to deal with the multi-objective challenges efficiently: 1) a new pheromone update rule based on a set of nondominated solutions from a global archive to guide each colony to search its optimization objective sufficiently; 2) a complementary heuristic strategy to avoid a colony only focusing on its corresponding single optimization objective, cooperating with the pheromone update rule to balance the search of both objectives; and 3) an elite study strategy to improve the solution quality of the global archive to help further approach the global Pareto front. Experimental simulations are conducted on five types of real-world scientific workflows and consider the properties of the Amazon EC2 cloud platform. The experimental results show that the proposed algorithm performs better than state-of-the-art multi-objective and constrained optimization approaches.

Wang *et al.* (2019), in their article, propose a dynamic group learning distributed particle swarm optimization (DGLDPSO) for large-scale optimization and extend it for large-scale cloud workflow scheduling. DGLDPSO is efficient for large-scale optimization due to its following two advantages. First, the entire population is divided into many groups, and these groups are coevolved by using the master-slave multigroup distributed model, forming a distributed PSO (DPSO) to enhance the algorithm diversity. Second, a dynamic group learning (DGL) strategy is adopted for DPSO to balance diversity and convergence. When DGLDPSO is applied to large-scale cloud workflow scheduling, an adaptive renumber strategy (ARS) is further developed to make solutions related to the resource characteristic and make the searching behaviour meaningful rather than aimless. Experiments are conducted on the large-scale benchmark functions set and the large-scale cloud workflow scheduling instances to investigate DGLDPSO's performance further. The comparison results show that DGLDPSO is better than or at least comparable to other state-of-the-art large-scale optimization algorithms and workflow scheduling algorithms.

Reis and De Carvalho (2019) presented a case study and framework for implementing BPM in higher education, highlighting the need for better process management tools in universities. Limitations: Focuses on a single case study and may only be generalizable to some universities or institutions. Pussep and Schulte, (2020) implementation of Business Process Management in Higher Education Institutions: A Case Study from an Administrative Perspective, Explored the implementation of BPM in a higher education institution, identifying critical success factors and challenges faced by the university administration; limitations: Focuses on a single case study, and may not be generalizable to all universities or institutions. Denecke and Kohler (2020) worked on framework for Digital Transformation in Higher Education: Integrating Process Automation and Data analytic: Proposed a comprehensive framework for digital transformation in higher education, addressing the integration of process automation (including workflow management) and data analytics for improved decision-making and efficiency limitations: Lacks empirical validation, and focuses on a conceptual framework rather than specific implementations.

3 Materials and Method

The research design used in this work is a mixed-methods approach, incorporating both qualitative and quantitative research methods. The qualitative research methods include interviews with stakeholders and focus groups, while the quantitative research methods include surveys and data analysis. The research design aims to gather data on the existing workflows in universities and identify the key features and requirements for a workflow management system.

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

http://doi.org/10.46792/fuoyejet.v8i4.1126 http:// journal .engineering.fuoye.edu.ng/

3.1 Research Questions

This paper will address the following research questions:

- What are the organization's specific needs and goals
- regarding workflow management?
- What features and functionality should the workflow management system include to meet those needs and goals?
- How can the effectiveness of the workflow management system be measured and evaluated?

3.1.1 Functional Requirements

- Communication: The system should enable communication between team members and stakeholders in the workflow process. This may include notifications, messaging, and collaboration tools.
- Workflow automation: The system should automate repetitive and routine tasks to improve efficiency and reduce errors.

3.2 Non-Functional Requirements

- Security: The system should have robust security measures to protect sensitive data and prevent unauthorized access.
- Scalability: The system should be able to handle a growing number of users and workflows as the university department expands.
- Usability: The system should be easy to use and navigate, with clear instructions and an intuitive interface.
- Performance: The system should be fast and responsive, with minimal lag or downtime.
- Accessibility: The system should be accessible to users with disabilities, with features such as screen reader support and keyboard navigation.

3.3 System Architecture

The proposed workflow management system will follow a client-server architecture, wherein the system will be accessible through a web-based interface (client) and hosted on a server. The web-based approach ensures widespread accessibility and compatibility with various devices, allowing students, faculty, and staff to access the system from their laptops, tablets, or smartphones.

System Architecture and Technology Stack

The workflow Management System has been developed using client-server architecture. The server side is responsible for processing requests, managing data, and ensuring secure access, while the client side provides an interface for users to interact with the system. The chosen technology stack includes;

Programming Language: Dart for the backend development due to its robustness and compatibility with various platforms.

Framework: Flutter Framework to facilitate rapid development, dependency injection, and RESTful API creation.

Database: Firebase Cloud Fire Store stores user data, communication records, and system logs.

Frontend: A web-based user interface using Dart programming language for easy access across multiple devices.

Security: Firebase authentication for user authentication and authorization, ensuring secure data transmission and access control.

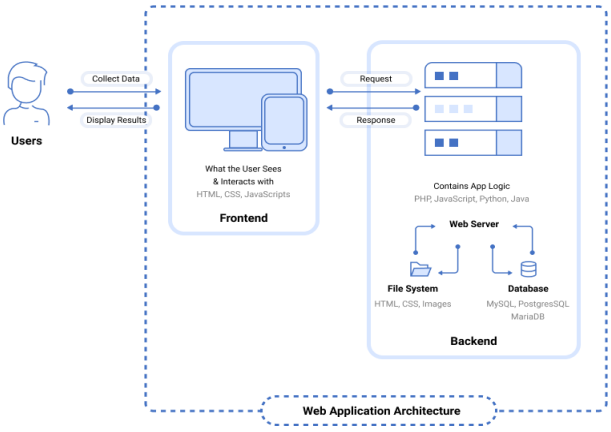

Fig. 1: System Architecture

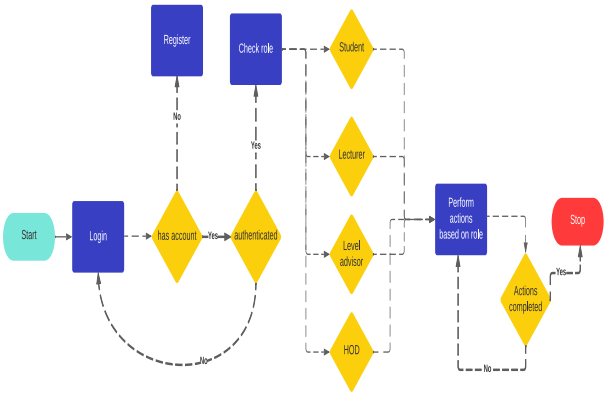

Fig. 2: Flow Chart of the system

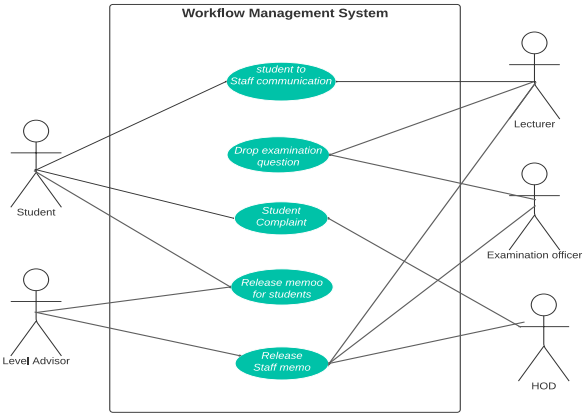

Fig. 3: Use Case diagram

3.4 Development Environment

The development environment for this project includes both hardware and software components. The hardware components include a personal computer with at least 8 GB of RAM and a 64-bit processor. The software components include the Flutter SDK Android Studio. The development environment is also set up to support continuous integration and deployment using tools such as GitHub and Jenkins.

3.5 Testing Process

The testing process for this project includes both manual and automated testing methods. Manual testing is used to

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

http://doi.org/10.46792/fuoyejet.v8i4.1126 http:// journal .engineering.fuoye.edu.ng/

test the user interface and application logic, while automated testing is used to test the system functionality and integration. The testing process also includes user acceptance testing, where stakeholders are invited to test the system and provide feedback on its usability and functionality.

3.6 Evaluation

The research was evaluated based on system functionality, usability, and performance. The evaluation uses objective and subjective measures, including system testing, user feedback, and performance metrics. The evaluation results refine the system and identify areas for future improvement.

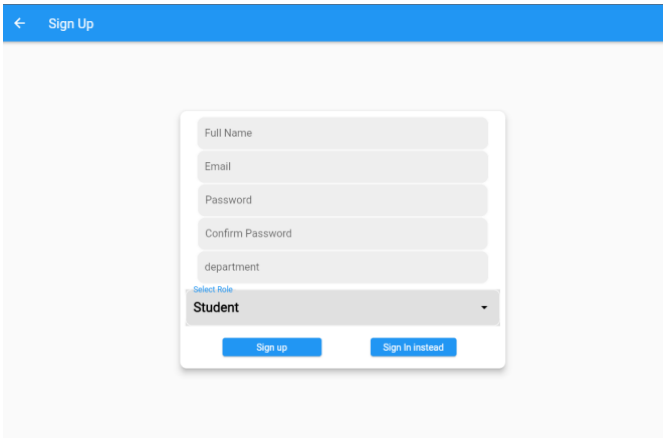

Fig. 4: Registration Screen

4 Results and Discussion

4.1 User Authentication

User authentication was implemented using JSON Web Tokens (JWT). When users log in, the system verifies their credentials and issues a unique token. This token is then used for subsequent API calls to authenticate and authorize user actions. By adopting JWT, the system enhances security by preventing unauthorized access to sensitive resources

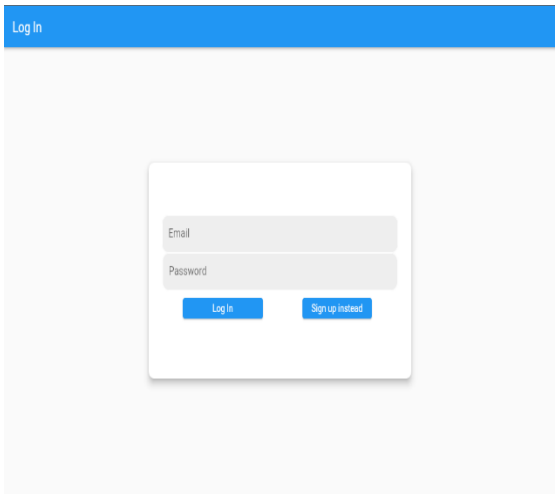

Fig. 5: Log-In Screen

4.2 Lecturers' Ability to Send Exam Questions to Exam Coordinator

Another essential feature of the WMS is facilitating communication between lecturers and the exam coordinator regarding examination materials.

Lecturers can upload exam questions through the system's interface, specifying the course and relevant details. These questions are securely transmitted to the exam coordinator. The coordinator can then review, organize, and prepare the examination based on the received inputs from various lecturers.

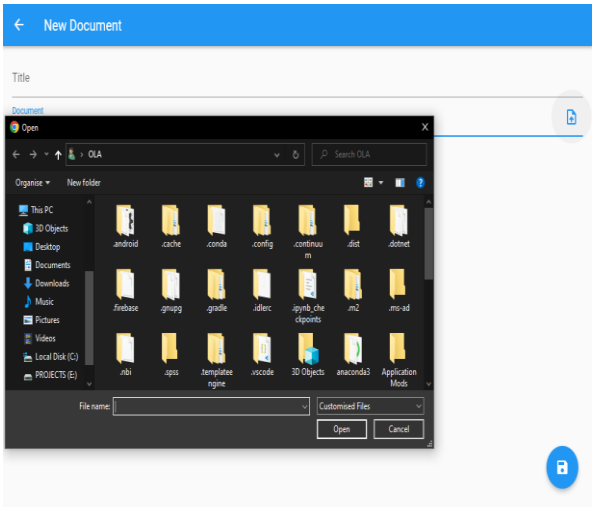

Fig. 6: Exam Uploading Screen

4.3 Implementation

The developed workflow management system using the Flutter framework and Firebase during this work. It was an exciting yet challenging endeavour. One of the implementation problems they encountered was designing the proper data structure and database schema. Initially, it was not easy to anticipate all the requirements, and we frequently changed the schema as the project progressed.

However, they overcame this challenge by starting with a well-thought-out initial design and preparing to iterate and adapt as needed. It used Firebase's NoSQL database capabilities to change the schema without significant disruptions. Another challenge we faced was ensuring real-time data synchronisation across multiple users and devices. Achieving this was complex, and we had to handle potential data inconsistencies and conflicts. Firebase's real-time database capabilities were leveraged on to address this and implemented proper conflict resolution strategies, such as timestamp-based conflict resolution and Firestore transactions. Handling user authentication and security was crucial to protect sensitive data and ensure only authorized users could access the system. It utilized Firebase Authentication for user login and implemented access control mechanisms, using Firebase security rules to define and enforce access restrictions based on user roles and permissions. As the workflow management system grew and more users joined, we encountered performance bottlenecks and scalability issues.

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

http://doi.org/10.46792/fuoyejet.v8i4.1126 http:// journal .engineering.fuoye.edu.ng/

To tackle this, the paper optimized my database queries, used Firebase indexing, and implemented caching mechanisms to improve performance. This research also designed the system to handle scale by leveraging Firebase's cloud infrastructure. Offline support was another essential aspect we had to consider since users needed to access and modify data while offline and sync changes once, they reconnected. This research work used Firebase's offline capabilities to store data locally and sync changes automatically when the device was online again. It also implemented a mechanism to handle conflicts arising from offline edits. Designing a user-friendly and intuitive interface for the workflow management system was challenging, especially with various features and interactions. The paperwork planned the user interface design carefully with usability in mind. Usability testing and gathered feedback from potential users were conducted to refine the interface. Design patterns and guidelines for workflow-related applications were done. Integration and compatibility issues arose while working with Firebase, Flutter, and other third-party packages.

It regularly updated the dependencies to address this and paid attention to breaking changes. It relied on official documentation and developer community forums to resolve integration-related issues. Testing and debugging were critical to ensuring a smooth workflow management system. Comprehensive unit tests and conducted systematic testing at each development stage were written. Debugging tools provided by Flutter and Firebase to diagnose issues quickly were used. Managing the project and dealing with time constraints were challenging tasks. The work is broken into manageable tasks and created a timeline with milestones. Essential features were prioritized first and implemented additional features when time permitted. Regularly communicated with my project advisor and sought help when encountering significant roadblocks. Documenting my code, database schema, and design decisions was essential for future reference and collaboration. Adopted a consistent coding style and followed best practices to improve code maintainability. Throughout the research, challenges as learning opportunities and remained persistent in finding solutions were embraced. Guidance was sorted from online communities, and relevant documentation to overcome implementation problems. Ultimately, the research is done, and the workflow management system using Flutter and Firebase proved valuable and rewarding.

4.4 Testing

It was essential to validate and verify that all functional requirements were met, and that the software performed as intended. Running test data through the system could ensure it worked correctly and identify any vulnerability to be addressed. In the testing phase, the research focused on various system components to ensure their reliability and accuracy.

4.5 Test Suite (for Unit Testing, Integration Testing, and System Testing)

Table 1. Test-case 001 (User Login)

LOGIN TEST	RESULT
Test Case Summary	Ensuring users can log in successfully using their roles and carry out tasks based on their roles
Prerequisite to testing	- The system must be up and running. - Users must be registered with their roles. - Uninterrupted internet connection
Test procedure	- Open the website in a browser - Navigate to the login page - Login
Test Data	- Email - Password
Expected result	User should be able to login successfully, and based on the roles access the system.
Actual Result	The user logged in successfully and the interface seen was based on the roles
Status	Passed

Table 2. Test Case 002 Logout Function

LOGOUT FUNCTION	RESULT
Test Case Summary	Ensuring user can Logout
Prerequisite	- Website is Running - Uninterrupted internet connection - Users are already logged in
Test Procedure	- Login - Click log out
Test Data	Logout
Expected Result	-User should be able to logout
Actual Result	User was able to logout
Status	Passed

Table 3. Test Case 003 Register Function

REGISTER FUNCTION	RESULT
Test Case Summary	Ensuring new users can register
Prerequisite	- Website is Running - Uninterrupted internet connection - User information has not been used before
Test Procedure	- Register - Click sign up - Enter all necessary information and select roles
Test Data	Register
Expected Result	-User should be able to register
Actual Result	User was able to register
Status	Passed

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

http://doi.org/10.46792/fuoyejet.v8i4.1126 http:// journal .engineering.fuoye.edu.ng/

Table 4. Test Case 004 Sending Memo Function

RELEASE MEMO FUNCTION	RESULT
Test Case Summary	Ensuring HOD and lecturers can send out Memo
Prerequisite	- Website is Running - Uninterrupted internet connection - User should be logged in - User role should be HOD or lecturer
Test Procedure	- Log in - Click send Memo - Enter all necessary information and select send
Test Data	Memo
Expected Result	-Memo should be sent out
Actual Result	Memo was sent out
Status	Passed

Table 5. Test Case 004 Viewing Memo Function

VIEW MEMO FUNCTION	RESULT
Test Case Summary	Ensuring users can view sent out Memo
Prerequisite	- Website is Running - Uninterrupted internet connection - User should be logged in - Memo should have been sent out
Test Procedure	- Log in - Click view memo
Test Data	Memo
Expected Result	-Memo should be viewed
Actual Result	Memo was viewed
Status	Passed

Table 6. Test Case 006 Sending Complaint Function

RELEASE COMPLAINS FUNCTION	RESULT
Test Case Summary	Ensuring student can send out complain
Prerequisite	- Website is Running - Uninterrupted internet connection - User should be logged in - User role should be student
Test Procedure	- Log in - Click send complain - Enter all necessary information and select send
Test Data	Complain
Expected Result	- Complain should be sent out
Actual Result	Complain was sent out
Status	Passed

Table 7. Test Case 004 Viewing Complaint Function

VIEW MEMO FUNCTION	RESULT
Test Case Summary	Ensuring HOD can view sent out Complain
Prerequisite	- Website is Running - Uninterrupted internet connection - User should be logged in - Complains should have been sent out
Test Procedure	- Log in - Click view Complain
Test Data	Complain
Expected Result	- Complain should be viewed by the HOD alone
Actual Result	Complain was viewed
Status	Passed

5 CONCLUSION

In conclusion, we presented the workflow Management System (WMS) implementation details for the Department of Mathematical and Computer Sciences. The system incorporates seamless communication between entities, robust security using user access control and authentication, and streamlined communication between lecturers and the exam coordinator. By deploying this WMS, the department can improve its workflow efficiency, enhance communication, and maintain data security, benefiting staff and students. Developing a workflow management system tailored for universities using Flutter and Firebase has demonstrated its potential to revolutionize university operations. By providing a user-friendly interface, automation capabilities, and efficient management of workflows, the system offers a valuable solution to enhance productivity and overall efficiency. The proposed system was developed to optimize universities' operations and meet their challenges in managing complex workflows.

Future work can build upon this foundation, implementing the system in real-world university settings and addressing the recommendations outlined to refine further and expand its functionality. The developed workflow management system significantly contributes to the advancement of university process management, positively impacting students, faculty, and staff.

REFERENCES

Ayodeji, A. O., & Olubiyi, M. O. (2020). Design and implementation of a workflow management system for universities. International Journal of Engineering Research & Technology, 9(7), 253- 259.

Feldmann, R., Witzmann, D., & Maedche, A. (2018). workflow management systems: A research perspective. Business & Information Systems Engineering, 60(6), 457-470.

Beyer, M., Binder, S., & Müller, C. (2019). Improving healthcare processes with a workflow management system: A case study in a hospital. Procedia CIRP, 84, 122-127.

Mendoza, J., Rodríguez, J. A., & Ramírez, J. (2020). Optimizing manufacturing processes with a workflow management system: A case study in a manufacturing plant. Procedia Manufacturing, 47, 657-662.

Denecke, K., & Kohler, M. (2020). A Framework for Digital Transformation in Higher Education: Integrating Process Automation and Data Analytics. Journal of Higher Education Policy and Management, 42(3), 263-280.

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)

http://doi.org/10.46792/fuoyejet.v8i4.1126 http:// journal .engineering.fuoye.edu.ng/

Masud, M., & Huang, L. (2013). Towards a University e-Administration: A Cloud-Based Approach. Procedia Computer Science, 17, 558-566.
Alomari, S., & Thomas, T. (2016). Evaluating the Efficiency of Workflow Management Systems for Higher Education Institutions. International Journal of Educational Management, 30(5), 738-754.
Pussep, A., & Schulte, S. (2020). Implementation of Business Process Management in Higher Education Institutions: A Case Study from an Administrative Perspective. International Journal of Information Systems in the Service Sector, 12(2), 1-23.
Li, Y., Liu, J., Wang, Y., & Xie, B. (2021). A performance evaluation framework for workflow management systems. Journal of Ambient Intelligence and Humanized Computing, 12(4), 4231-4244.
Zhang, Y., Zhang, C., Li, Y., & Yuan, Y. (2019). A simulation-based approach for workflow optimization. IEEE Access, 7, 122875-122884.
Van Der Aalst, W. M. (2016). Process mining: discovery, conformance and enhancement of business processes. Springer.
Dijkman, R., Dumas, M., & Ouyang, C. (2020). Workflow management: A survey of issues and approaches. ACM Computing Surveys, 53(2), 1-37.
Ghose, A., & Mahajan, R. (2018). Workflow management systems: A review of recent advances and trends. Decision Support Systems, 114, 1-10.
Sarker, S., & Lee, W. (2020). Research trends and future directions in workflow management systems. Information Systems Frontiers, 22(6), 1335-1355.
Li, H., Chen, Y., & Wu, Y. (2021). Cloud-based workflow management systems: Challenges and opportunities. Future Generation Computer Systems, 116, 259-269.
Zhao, Y., Song, L., Zhang, W., Chen, X., & Chen, Q. (2020). Blockchain-based workflow management. IEEE Transactions on Services Computing, 13(2), 290-303.
Zhang, J., Wang, J., Qin, Y., & Tang, H. (2019). Integration of IoT and workflow management system. Future Generation Computer Systems, 97, 107-115.
"Firebase." Firebase. Accessed 6 April 2023, https://Fire base.google.com/ "Web Application." Techopedia. Accessed 6 April2023,https://www.techopedia.com/definition/2386/web-application
"Flutter for Web." Flutter. Accessed 6 April 2023, https://flutter.dev/web
"Agile Methodology." Agile Alliance. Accessed 6April2023,https://www.agilealliance.org/agile101/
"Data Management." Techopedia. Accessed 6 April2023,https//www.techopedia.com/ definition/25245/data-management
"Security." Techopedia. Accessed 6 April 2023, https://www.techopedia.com/definition/26232/security
"Scalability." Techopedia. Accessed 6 April 2023,https://www.techopedia.com/definition/31 1/scalability
"Cloud Computing." NIST. Accessed 6 April 2023, https://www.nist.gov/topics/cloud computing.

© 2023 The Author(s). Published by Faculty of Engineering, Federal University Oye-Ekiti.
This is an open access article under the CC BY NC license. (https://creativecommons.org/licenses/by-nc/4.0/)