

GRIEVANCE HANDLING

A Cloud-Based Complaint Management System for Scalable, Secure, and Transparent University Grievance Handling

Osunniyi James Segun, Adedokun Kehinde Hafeez

Department of Computer Science, Ladoke Akintola University of Technology, Oyo State, Nigeria

Published October 2026

Abstract

Manual complaint handling in tertiary institutions is often associated with delayed responses, misplaced records, weak monitoring, and limited transparency. These challenges can reduce administrative efficiency and negatively affect students' and staff members' confidence in institutional grievance processes. This study presents the design and implementation of a cloud-based complaint management system developed to provide a scalable, secure, and traceable mechanism for managing university complaints. The system was developed using Object-Oriented Analysis and Design (OOAD) within an iterative development approach. Its architecture combines a React-based frontend with a Node.js and Express backend, PostgreSQL and MongoDB databases, and cloud services hosted on Amazon Web Services (AWS). PostgreSQL and Prisma ORM support structured transactional data, while MongoDB and Mongoose provide flexible storage for audit and operational logs. Security mechanisms include Role-Based Access Control (RBAC), password hashing, encrypted communication, restricted cloud permissions, audit logging, and web application protection. Automated complaint routing and SMS notifications were incorporated to improve workflow efficiency. Functional, integration, security, user acceptance, and load tests were conducted using simulated data. The evaluation recorded a median API response latency of 85 ms, maximum observed throughput of 750 concurrent requests per second under the configured test conditions, and system uptime of 99.98%. The automated workflow also reduced median complaint acknowledgement time by 85% compared with the historical paper-based process. The findings demonstrate that a cloud-based architecture can improve the scalability, security, traceability, and operational efficiency of university complaint management.

Keywords: cloud computing; complaint management; university administration; scalability; cybersecurity; role-based access control; PostgreSQL; MongoDB; AWS

1 Introduction

Complaint management is an important component of effective university administration because it provides students and staff with a formal channel for reporting academic, administrative, welfare, and service-related concerns. In many tertiary institutions, however, complaint processes continue to depend partly on paper forms, physical submissions, spreadsheets, email exchanges, and informal communication. Such approaches can result in delayed acknowledgement, misplaced records, inconsistent routing, and difficulty monitoring the progress of unresolved complaints [1], [2].

The increasing availability of cloud computing provides an opportunity to redesign these processes. Cloud computing enables convenient, on-demand access to configurable computing resources and supports rapid provisioning and elasticity [3]. These characteristics make cloud platforms suitable

for institutional applications that may experience variable workloads and require continuous availability.

Existing university complaint-management applications have demonstrated the usefulness of web-based submission and tracking [4], [5]. Nevertheless, many systems primarily concentrate on capturing complaints without providing an integrated architecture for scalable deployment, secure data management, automated routing, notification, and comprehensive auditability. Cloud adoption in Nigerian higher education has also been associated with opportunities for improving institutional administration and service delivery [2].

This study therefore designed and implemented a cloud-based complaint management system for university environments. The specific objectives were to: (i) develop an accessible web-based platform for submitting and tracking complaints; (ii) automate complaint acknowledgement and routing; (iii) implement security mechanisms for protecting complaint information; (iv) provide reliable storage for transactional and audit data; and (v) evaluate system performance, availability, and workflow efficiency under simulated workloads.

2 Literature review

2.1 Complaint management in higher education

Complaint management involves the systematic recording, classification, assignment, investigation, resolution, and closure of complaints. Traditional paper-based processes can create administrative bottlenecks because records may have to pass through several offices before reaching the appropriate authority [1]. Web-based complaint systems address some of these limitations by centralising complaint records and making their status easier to monitor [4], [5].

However, a complaint platform must provide more than an online form. Effective systems require controlled access, traceability, timely notification, reliable storage, and mechanisms for assigning complaints to responsible personnel. The absence of these capabilities can limit the effectiveness of otherwise functional complaint portals.

2.2 Cloud computing and scalable architecture

Cloud computing provides a foundation for deploying applications whose resource requirements may change over time [3]. Rapid elasticity, resource pooling, broad network access, and measured service make cloud environments particularly suitable for applications requiring variable capacity and remote accessibility.

For complaint management, cloud deployment can support centralised access, automated backups, flexible resource allocation, and high availability. AWS services provide infrastructure and managed services that can support application hosting, database management, object storage, security controls, and monitoring [6].

2.3 Security and data protection

Complaint records may contain personally identifiable and institutionally sensitive information. Consequently, confidentiality, integrity, and availability must be considered throughout system design. Role-Based Access Control restricts users according to their responsibilities, while password hashing and encrypted communication reduce the risks associated with credential and data exposure.

The Nigeria Data Protection Act (NDPA) 2023 further establishes requirements for responsible processing and protection of personal data in Nigeria [7]. Cloud-based institutional systems should therefore incorporate security controls into application, database, and infrastructure layers rather than treating security as an afterthought.

2.4 Research gap

Existing complaint-management systems provide useful mechanisms for electronic submission and tracking, but there remains a need for integrated solutions that combine cloud deployment, scalable processing, secure access control, heterogeneous data storage, automated routing, notifications, and auditability. The system developed in this study addresses these requirements through a layered cloud-based architecture and a polyglot persistence strategy.

3 Methodology

3.1 Research design and development approach

The study adopted a Design Science-oriented system development approach supported by Object-Oriented Analysis and Design (OOAD). Requirements were identified through review of existing complaint-handling procedures, observation, structured interviews, and questionnaire-based requirements gathering. The development process followed an iterative approach involving requirements analysis, system design, implementation, testing, and refinement.

The system supports the complete complaint lifecycle: submission, acknowledgement, assignment, investigation, resolution, and closure. Students and staff can submit complaints through a web interface, while authorised administrators can review, assign, update, and close complaints.

3.2 System architecture

The system uses a layered cloud-based architecture consisting of presentation, application, and infrastructure/data layers. The presentation layer was implemented with React and Vite, with Tailwind CSS used for interface styling. Zustand provides client-side state management, while Axios supports communication with backend APIs.

The application layer uses Node.js and Express.js to implement RESTful services. Complaint-processing logic, authentication, routing, notification, and administrative functions are exposed through controlled application interfaces.

The data layer adopts a polyglot database strategy. PostgreSQL, accessed through Prisma ORM, stores structured transactional information such as users, complaints, categories, assignments, and complaint status. MongoDB, accessed through Mongoose, stores flexible audit and operational log data. Amazon S3 provides object storage for complaint-related file attachments.

3.3 Cloud deployment and technologies

The system was deployed using AWS cloud infrastructure and Docker-based containerisation. Amazon RDS supports the PostgreSQL database, while cloud object storage is used for attachments. NGINX is incorporated as a reverse proxy within the deployment architecture. Continuous integration and deployment practices support repeatable application deployment and maintenance.

Automated routing assigns complaints to relevant administrative units, while Twilio webhooks support SMS notifications. These mechanisms reduce dependence on manual forwarding and improve the timeliness of communication.

3.4 Security implementation

Security was incorporated at multiple levels. JWT-based authentication and RBAC were used to restrict access according to user roles. Passwords were protected through cryptographic hashing, while HTTPS/TLS was used to protect data transmitted between clients and application services. AWS security controls, restricted permissions, web application protection, and audit logging were incorporated to strengthen infrastructure and application security.

These measures support the protection of confidentiality, integrity, and availability while providing an auditable record of relevant system activities.

3.5 System evaluation

The completed system was evaluated through functional, integration, security, user acceptance, and load testing. Synthetic data were used during performance evaluation to avoid exposing real personal information. Key measures included response latency, throughput, system availability, complaint acknowledgement time, and successful completion of core complaint-management functions.

4 Results and discussion

4.1 System performance

The evaluation demonstrated that the system could support substantial concurrent demand while maintaining relatively low response latency.

Table 1. System Performance Results.

Metric	Observed Result
Median API response latency	85 ms
Maximum observed throughput	750 concurrent requests/second
System uptime	99.98%
Reduction in median acknowledgement time	85%

The median API response latency of 85 ms indicates that the application provided responsive service under the evaluated workload. A maximum observed throughput of 750 concurrent requests per second demonstrates the capacity of the implemented architecture under the configured test conditions. The recorded uptime of 99.98% further indicates a high level of service availability.

4.2 Complaint workflow efficiency

A major improvement resulted from replacing manual routing and acknowledgement procedures with automated processing. Following submission, complaints can be recorded, acknowledged, categorised, and routed to the appropriate administrative unit without requiring repeated physical movement of documents.

The evaluation recorded an 85% reduction in median acknowledgement time compared with the historical paper-based process. This result indicates that workflow automation can significantly reduce the delay between complaint submission and institutional acknowledgement.

4.3 Security and data management

The combination of RBAC, password hashing, encrypted communication, restricted cloud permissions, and audit logging provides multiple layers of protection. The use of PostgreSQL for structured transactional information and MongoDB for flexible audit and operational data also separates different data-management requirements.

The architecture therefore avoids relying on a single database model for all system functions. Structured records benefit from relational consistency, while operational and audit information can be stored with greater flexibility.

4.4 Discussion

The findings support the suitability of cloud computing for complaint-management applications requiring remote access, scalable resources, and continuous availability. The use of cloud services also provides an architectural foundation for future expansion as institutional demand increases.

The results are particularly relevant to tertiary institutions where complaint volumes can vary according to academic calendars, registration periods, examinations, accommodation processes, and other institutional activities. Automated routing and notification reduce administrative intervention, while audit trails improve traceability.

The system nevertheless represents a focused institutional complaint-management solution rather than a complete university information system. Deep integration with student information systems and learning-management systems was outside the scope of the implementation. Similarly, advanced artificial intelligence and natural language processing capabilities were not included. These limitations provide opportunities for future development.

5 Conclusion

This study presented the design and implementation of a cloud-based complaint management system for improving the scalability, security, transparency, and efficiency of university grievance handling. The solution combines a React frontend, Node.js and Express backend, PostgreSQL and MongoDB databases, AWS cloud services, automated complaint routing, SMS notification, and layered security controls.

Evaluation results showed a median API response latency of 85 ms, maximum observed throughput of 750 concurrent requests per second, system uptime of 99.98%, and an 85% reduction in median complaint acknowledgement time compared with the historical manual process. These results demonstrate that the implemented architecture can provide a responsive and reliable alternative to paper-based complaint management.

Future enhancements should investigate natural language processing for automated complaint classification and sentiment analysis, interoperability with institutional information systems, predictive analytics using accumulated audit data, and development of a dedicated mobile application. Such extensions could further improve automation, accessibility, and institutional decision-making.

References

- [1] Adebayo AO. Development of a web-based complaint tracking system for higher institutions. *Nigerian Journal of Computing*. 2020;17(2):45–58.
- [2] Adeleke IA, Muraina IO, Adegbuyi KK. Adoption of cloud computing technology for effective university administration in Nigeria. *Current Journal of Applied Science and Technology*. 2020;39(40):1–8. <https://doi.org/10.9734/cjast/2020/v39i4031106>
- [3] Mell P, Grance T. *The NIST definition of cloud computing*. NIST Special Publication 800-145. National Institute of Standards and Technology; 2011. <https://doi.org/10.6028/NIST.SP.800-145>
- [4] Anusiuba O, et al. *Design and implementation of a tertiary institution web-based student complaint management system*. International Journal of Software & Hardware Research in Engineering. 2021;9(8). <https://doi.org/10.26821/IJSHRE.9.8.2021.9802>
- [5] Fabiya AA, Olorunlomeye AB, Oluwatosin AO, Ugwu JI, Ramoni TA. Web-based campus complaint management system (WCCMS). *International Journal of Computer Trends and Technology*. 2022;70(10):28–36. <https://doi.org/10.14445/22312803/IJCTT-V70I10P105>
- [6] Amazon Web Services. *AWS Well-Architected Framework*. Amazon Web Services; 2024.
- [7] Federal Republic of Nigeria. Nigeria Data Protection Act, 2023. Official Gazette No. 119, vol. 110. Nigeria Data Protection Commission; 2023.
- [8] International Organization for Standardization. *ISO/IEC 27001:2022 Information security management systems — Requirements*. ISO; 2022.
- [9] Kindervag J. *Build security into your network's DNA: The zero trust network architecture*. Forrester Research; 2021.
- [10] Sharma S, Gupta R, Kumar A. Re-evaluating the CIA triad for cloud-native distributed systems. *Cybersecurity and Infrastructure Journal*. 2023;5(2):34–50.
- [11] Mohamed HM, Mohamed AH, Hashi SO, Hassan HN, Barre KA, Ahmed BA. Automated university complaint management system by leveraging machine learning and natural language processing for enhanced efficiency and accuracy. *Forum for Linguistic Studies*. 2025;7(2):354–373. <https://doi.org/10.30564/fls.v7i2.8162>
- [12] Davis FD. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*. 1989;13(3):319–340. <https://doi.org/10.2307/249008>