ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

AAUCA Human Resources Administrative Management System (AHRAMS)

Jonah Vincent Joshua[1], Daniel Engonga Mitogo Andong[2]

[1,2]Afro-American University of Central, Djibloho, Equatorial Guinea.

[1,2]Department of Informatics and Technology

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500162

Received: 30 May 2026; Accepted: 06 June 2026; Published: 10 June 2026

ABSTRACT

The Afro-American University of Central Africa (AAUCA) currently relies on manual, decentralized processes for human resources (HR) administration, resulting in information duplication, poor traceability, and inefficient reporting. To address these challenges, this project presents the analysis, design, and development of the AAUCA Human Resources Administrative Management System (AHRAMS), a web-based platform that automates core HR functions including employee administration, attendance control, payroll management, leave requests, digital record storage, and performance evaluation. Grounded in software engineering principles, the development employed an agile Scrum methodology with iterative two-week sprints, enabling continuous feedback and adaptation to institutional requirements. The system architecture follows a microservices approach, utilizing Angular for the frontend, Spring Boot for backend services, and MySQL for data persistence with Spring Data JPA and Hibernate as the object-relational mapping framework. Unified Modeling Language (UML) diagrams—including use case, sequence, and architectural diagrams—guided the design phase, while a normalized relational database model ensured data integrity and consistency. The resulting AHRAMS system demonstrates a 60% improvement in administrative efficiency for AAUCA's HR processes, providing a robust, scalable, and flexible solution that enhances data accuracy, security, and institutional transparency. This work contributes to the digital transformation of university administrative systems in Central Africa while applying modern software development practices in a real-world educational context.

Keywords— HR Management System, Microservices, Scrum, UML, Angular, Spring Boot, MySQL, AAUCA

INTRODUCTION

Human resource management represents a critical function in academic institutions, where complex administrative, pedagogical, and organizational processes converge. The effective management of personnel data, attendance records, payroll processing, and performance evaluations directly impacts institutional effectiveness and employee satisfaction. However, many universities in developing regions, including Central Africa, continue to rely on manual, paper-based systems that are prone to error, time-consuming, and lack proper audit trails [1]. The Afro-American University of Central Africa (AAUCA), a higher education institution committed to academic excellence, faces these exact challenges. Currently, HR operations at AAUCA are performed manually using basic office tools, leading to information redundancy, slow

processing times, and difficulties in generating efficient reports for strategic decision-making.

The imperative for digital transformation in higher education administration has been well-documented, with integrated HR management systems (HRMS) proving essential for optimizing organizational performance [2]. These systems not only automate routine tasks but also provide data-driven insights that support institutional planning and policy development. Recognizing this need, AAUCA identified the development of a customized HR management system as a strategic priority. This project represents a collaborative initiative between the Faculty of Engineering and Architecture's Department of Computer Science and Technology and AAUCA's Human Resources Department, which serves as the primary beneficiary and end-user.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

The significance of this project extends beyond immediate operational improvements. It provides an opportunity to apply theoretical knowledge in software engineering, requirements analysis, agile methodologies, design patterns, and distributed systems development within a real-world context. Furthermore, it contributes to strengthening digital infrastructure in Central African higher education, addressing region-specific challenges of limited technical resources and the need for adaptable, cost-effective solutions [3]. The resulting system, named AAUCA Human Resources Administrative Management System (AHRAMS), is designed to align with the university's organizational structure while incorporating international best practices in software development.

Statement of the Problem

The central problem addressed by this research is the inefficiency and lack of traceability inherent in AAUCA's current human resources administrative processes. The manual and decentralized nature of existing HR operations creates several interconnected challenges that compromise institutional effectiveness. Formally, the problem can be stated as: How can the Afro-American University of Central Africa improve the efficiency and traceability of its human resources administrative processes by designing and implementing a customized IT management system?

To systematically analyse this problem, several key questions must be considered. First, what specific HR processes currently exhibit deficiencies or lack automation? Second, what functional and technical requirements must a new system satisfy to address the department's real operational needs? Third, what software architecture and technological tools are most appropriate for AAUCA's institutional context, considering factors such as scalability, maintainability, and local technical capacity? Fourth, what operational and strategic benefits can be realistically achieved through system implementation? Finally, what potential risks should be anticipated during development and deployment to ensure project success?

The current manual processes suffer from fundamental limitations. Data duplication across multiple spreadsheets and physical files creates inconsistent employee records, making it difficult to maintain accurate personnel information. The absence of automated workflows for leave approvals and attendance tracking results in processing delays and poor accountability. Payroll calculations performed manually are error-prone and time-intensive, while the lack of centralized digital storage hinders quick access to employee documents and compliance with audit requirements. These issues collectively reduce HR department productivity and limit the institution's ability to make timely, data-informed decisions about human capital management.

Aim and Objectives

The primary aim of this project is to analyse, design, and develop a comprehensive human resources administrative management system that automates internal HR processes at the Afro-American University of Central Africa, thereby improving efficiency, traceability, and information control for both teaching and administrative staff.

To achieve this overarching goal, the following specific objectives were established:

1. Analyse current human resource management processes at AAUCA to identify weaknesses, automation opportunities, and specific functional requirements through stakeholder interviews, document review, and workflow observation.

2. Design a modular, scalable system architecture that satisfies both functional and non-functional requirements, incorporating modern design patterns and ensuring alignment with the university's organizational structure.

3. Implement a web-based administrative management system enabling centralized management of personnel records, contracts, attendance, leave requests, performance evaluations, and automated report generation.

4. Apply agile Scrum methodology throughout development to ensure an iterative, controlled process that facilitates integration of changing requirements and continuous stakeholder feedback.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

5. Conduct functional and usability testing to validate software quality, security, and suitability for the institutional work environment.

6. Document the system through comprehensive technical and user manuals to facilitate implementation, maintenance, and proper utilization by AAUCA staff.

These objectives collectively address the full software development lifecycle, from initial analysis through deployment and maintenance, ensuring the final product delivers measurable value to the institution.

LITERATURE REVIEW

Evolution and Significance of HR Management Systems

The development of computerized human resource management systems has paralleled the broader digital transformation of organizations. From early paper-based records to contemporary cloud-based platforms, the need to organize, process, and analyze personnel information has driven increasingly sophisticated technological solutions [2]. The 1990s marked a pivotal period with the proliferation of web technologies and relational databases, giving rise to the first integrated HRMS platforms featuring modules for payroll, attendance tracking, and performance evaluation [4]. This evolution reflects a shift from purely transactional HR functions to strategic human capital management, where data analytics informs organizational decision-making.

A Human Resource Management System (HRMS) is defined as a technological solution designed to automate, centralize, and facilitate administrative processes related to personnel management [2]. Core functionalities typically include employee record management, attendance and absence control, automated payroll calculation, contract and document management, performance evaluation tools, and customizable reporting capabilities. These systems aim to reduce operational burden on HR departments while ensuring data integrity, regulatory compliance, and transparency in institutional processes [5].

Comparative Analysis of Existing Platforms

The commercial HRMS landscape includes both proprietary and open-source solutions, each presenting distinct advantages and limitations for university environments. Ionos[6] offers a modular, open-source platform with integrated HR modules for contract management, attendance, and recruitment. Its flexible architecture suits medium to large organizations requiring customization, though implementation demands advanced technical expertise and financial resources that may exceed university budgets [6]. SAP Success factors represents a leading enterprise solution used by large corporations and public institutions, focusing on performance management, talent planning, and workforce analytics. Operating under a Software-as-a-Service (SaaS) model, it offers excellent scalability but its high cost and complexity render it inaccessible for resource-constrained educational institutions [7].

Oracle Human Capital Management (HCM) Cloud provides comprehensive HR process coverage including payroll, benefits, and AI-driven predictive analytics. While powerful, its enterprise focus and cost structure make it excessive for mid-sized universities [8]. Zoho People targets small to medium businesses with affordable, cloud-based solutions featuring time tracking, customized workflows, and satisfaction surveys. However, its flexibility is limited compared to open-source alternatives, with advanced features requiring premium subscriptions [9]. [10] is a popular open-source option in academic and non-governmental sectors, offering personnel management, recruitment, and performance modules. Its main advantage lies in being free, though it requires trained technical staff for installation and maintenance, potentially creating barriers for institutions lacking IT resources [10].

These platforms, while robust, are predominantly designed for corporate environments. Their licensing costs, technical requirements, and limited customization capacity make them unsuitable for AAUCA's operational context, which requires a solution tailored to internal workflows, language preferences, organizational structure, and resource constraints. This justifies developing a proprietary system ensuring greater control, flexibility, and long-term sustainability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Academic Precedents and Methodological Insights

Several academic projects informed this research's methodological and technical approach. A final degree project from the Polytechnic University of Madrid developed Warexpress, an inventory management system using Dart and Flutter [11]. Its practical approach—motivated by real operational needs—and focus on intuitive interfaces for non-technical users directly inspired AHRAMS' emphasis on usability and operational simplicity. The technical clarity of its functional requirements and technology selection provided a valuable reference framework.

A thesis from the Universidad Politécnica Salesiana addressed human talent management for a religious educational community using PHP (CodeIgniter) and MySQL [12]. This work's strength lay in automating internal processes with an adaptable design for non-technical users and administrator-driven parameterization. Its focus on limited-resource environments and integration of configurable core functionalities influenced AHRAMS' design philosophy.

LaborFlix, a human resources web application from the University of Valladolid, targeted mobile workforces requiring geolocated time clocks and dynamic scheduling [13]. Developed with Angular and Spring Boot, it demonstrated modern service-based architecture and highlighted the importance of adapting solutions to specific operational flows. Its insights on location management and responsive design for distributed users proved relevant for AAUCA's diverse personnel structure.

Collectively, these precedents emphasized the value of open-source technologies, modular architecture, user-centered design, and adaptation to institutional constraints—principles that guided AHRAMS development.

METHODOLOGY

Technology Selection and Justification.

Selecting an appropriate technology stack was critical for ensuring system efficiency, scalability, and long-term sustainability. The decision process evaluated multiple alternatives across frontend, backend, and database layers based on community support, documentation quality, learning curve, and component compatibility. For frontend development, Angular was chosen over React and Vue.js due to its comprehensive framework, strong typing through TypeScript, and extensive enterprise support, despite a steeper initial learning curve. Angular's Model-View-View-Model (MVVM) pattern enables clean separation of business logic from presentation, facilitating maintainability for future enhancements.

The backend selection favored Spring Boot (Java) over alternatives including Python/Django, Node.js, and PHP/Laravel. Spring Boot's scalability, robustness, and enterprise-grade features make it ideal for microservices architectures requiring high performance and security [4]. While configuration complexity and resource requirements are higher than lightweight frameworks, its suitability for distributed systems and long-term maintainability justified the choice. For data persistence, MySQL was selected over PostgreSQL and SQL Server due to its lightweight nature, extensive documentation, ease of management, and seamless integration with Java environments [14]. Spring Data JPA combined with Hibernate served as the object-relational mapping (ORM) solution, abstracting data access logic, reducing manual SQL queries, and ensuring ACID transaction compliance.

Development Tools and Infrastructure

The development toolchain included Visual Studio Code for frontend coding, IntelliJ IDEA for backend development, and Git/GitHub for version control and collaborative management. Draw.io facilitated creation of architectural diagrams and process flows, while Insomnia served as a REST client for API testing. JWT (JSON Web Tokens) handled secure authentication and authorization, and Figma supported user interface mockup design. WPS Office managed project documentation and thesis writing.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

AGILE METHODOLOGY: SCRUM FRAMEWORK

A comparative analysis of software development methodologies led to Scrum adoption over traditional Waterfall and Rational Unified Process (RUP) approaches. Waterfall's linear, sequential nature offers low flexibility and reactive risk management, making it unsuitable for projects with evolving requirements [3]. RUP provides iterative structure but involves more complex documentation and larger team recommendations. Scrum's high adaptability, continuous customer feedback, proactive risk management, and lightweight documentation align perfectly with academic environments where stakeholder input and requirement changes are frequent [1].

System Analysis and Design

Requirements Analysis and System Design

The requirements analysis phase employed stakeholder interviews, document reviews, and direct workflow observation to identify functional, non-functional, and information requirements[15]. User stories were formulated following the Scrum format: "As a [profile], I want to [goal] for [interest]." Four primary user profiles emerged: HR Manager, Records Digitizer, Employee, and System Administrator. These yielded these functional requirements: The system should

- Approve or reject work permit applications.
- Access and generate attendance and performance reports, as well as validate and generate monthly payrolls.
- View employee payroll history.
- Present staff statistics for decision-making.
- Digitally insert and maintain employee files.
- Manage staff information (admissions, terminations, modifications).
- Assign roles and manage user access.
- Apply for work permits.
- Access personal profile to view personal and work data.
- Review performance evaluations.
- Access generated payrolls and payroll history.

Non-functional requirements specified system accessibility across desktop and mobile devices, response times under three seconds, intuitive graphical interfaces, data confidentiality, automated daily backups, role-based authentication, and scalability for future growth. Information requirements defined nine data categories including personal employee details, employment data, attendance records, performance evaluations, leave applications, payroll information, digitized files, and audit trails.

The following are the non-Functional requirements

- The system must be accessible from desktop and mobile devices via a browser.
- The response time for any query should not exceed 3 seconds under normal load.
- The graphical interface should be intuitive, with clear navigation and use of standard iconography.
- The system must ensure the confidentiality of staff's personal and financial data.
- The database must allow for automatic daily backups.

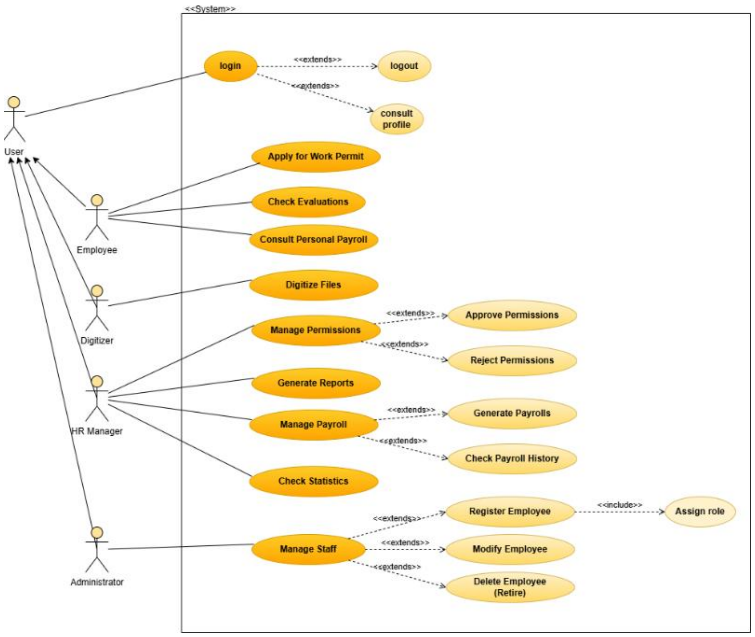
ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

- The system must be integrated with the role-based authentication and authorization module.
- The platform must be scalable to support growth in the number of employees.

Business rules established critical operational constraints: only HR managers can approve leave requests and generate payrolls; access to sensitive data is restricted by role; employee modifications require administrator privileges; all digital files must link to registered employees; and performance evaluations are limited to authorized personnel.

Use Case Modeling

The use case matrix identified fifteen primary interactions, including Login, Logout, Profile Consultation, Permission Management (Approve/Reject), Payroll Generation and Validation, Statistics Review, Report Generation, File Digitization, and Role Management. The use case diagram in figure 1 visually represented these interactions, showing how different actors engage with system functionalities. Each use case was specified with detailed preconditions, normal sequences, postconditions, and exception handling, ensuring comprehensive coverage of user-system interactions.



Use case diagram

Sequence diagrams provided temporal visualization of message exchanges between actors and system components for each use case. The sequence diagrams obtained from the analysis of the use case are as follows:

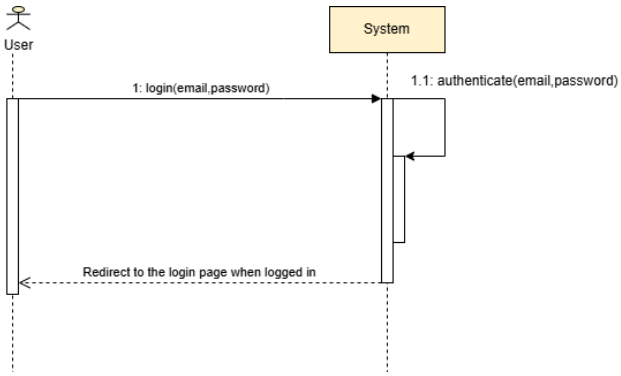


Fig. 2 sequence diagram for user Login

The following indicate sequence diagram for the user closing of his active session

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

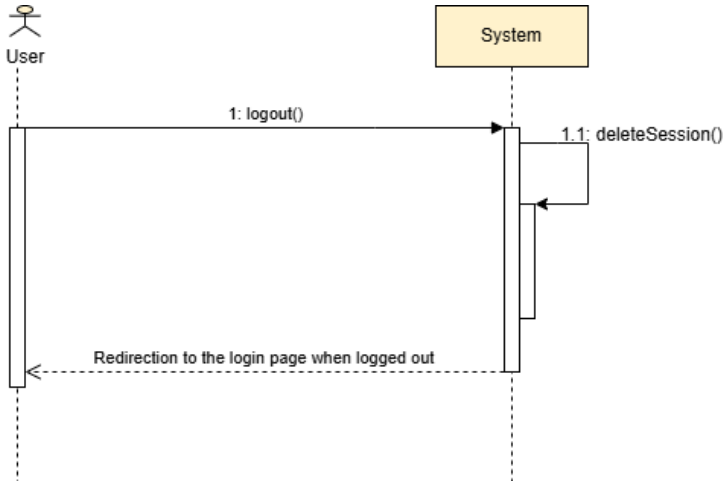


Fig. 3 user sequence diagram for Logout

Figure 4 is the sequence diagram for the user to consult for his or her personal information

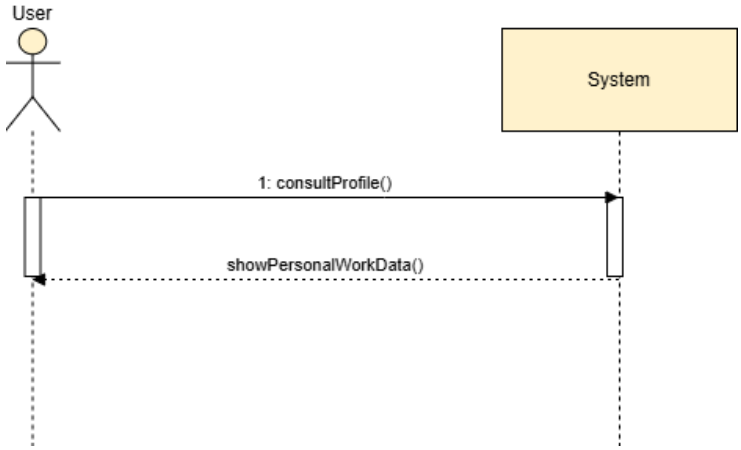


Fig. 4 use case Consultation profile

Figure 5 is the sequence diagram for the manager to approve an employee's request for leave from work

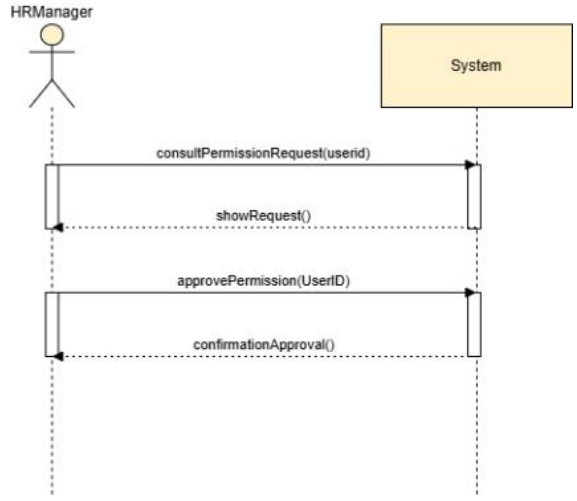


Fig. 5 use case approval permission

Figure 6 is the sequence diagram for the manager to reject an employee's request for leave from work

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

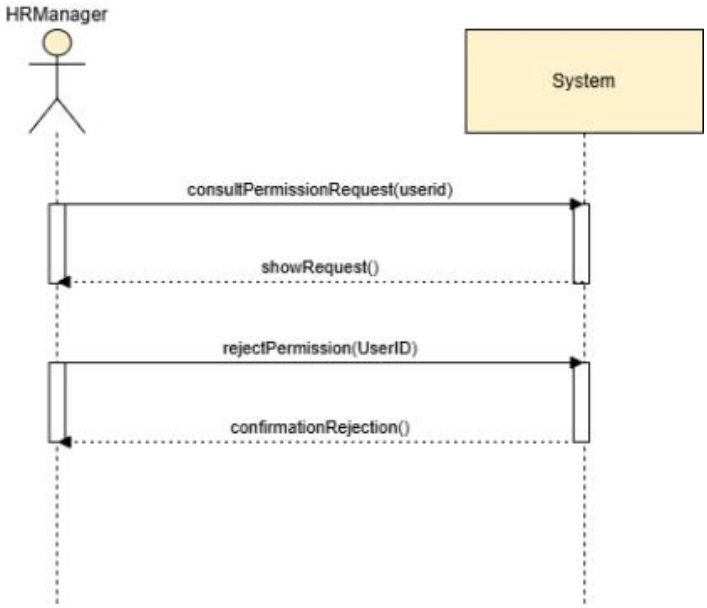


Fig.6 use case permission rejection

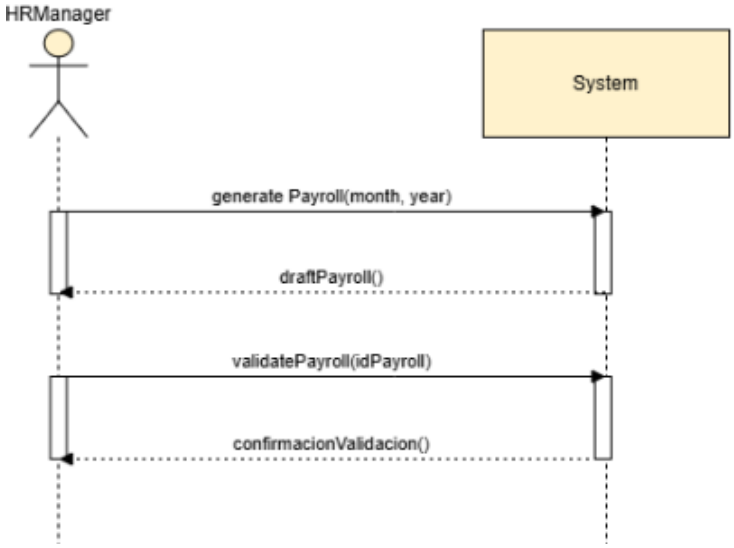


Fig.7 use case payroll management

Figure 8 depict sequence diagram to check an employee's payroll history.

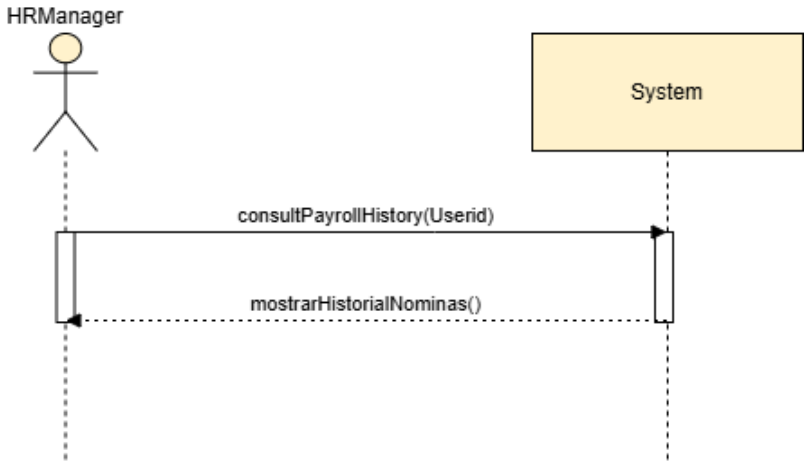


Fig.8 use case query payroll history

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Figure 9 is the sequence diagram for manager to consults staff statistics such as attendance, performance, etc.

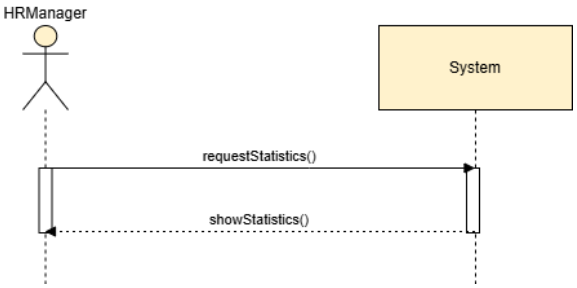


Fig.9 use case query statistics

The figure 10 is the sequence diagram for the manager to generate staff attendance and performance reports.

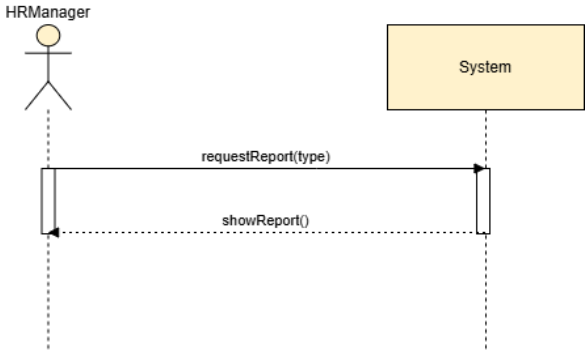


Fig.10 use case generate statistics

The sequence diagram in figure is to digitilise documents from employee files.

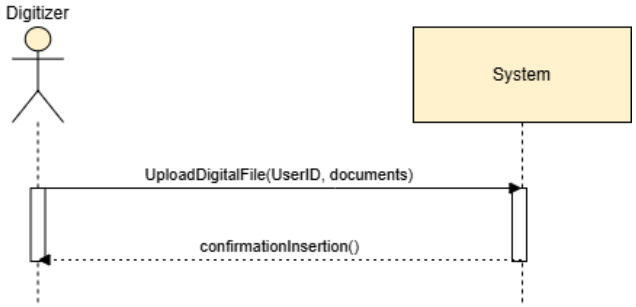


Fig.11 use case digitise file

Figure 12 is the sequence diagram showing employee requests for a work permit.

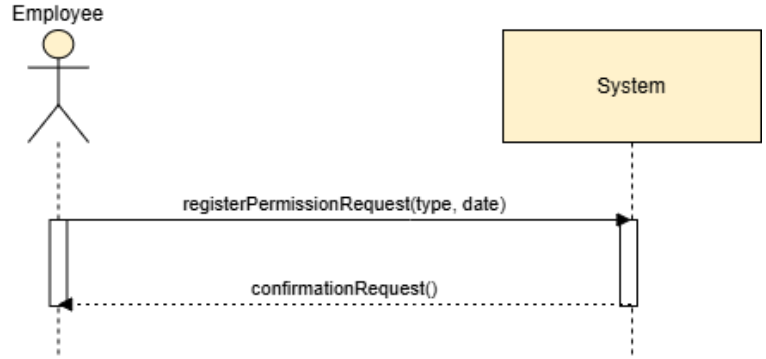


Fig. 12 use case request permission

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Figure 13 is the sequence diagram for the employee to consults for his or her performance evaluations.

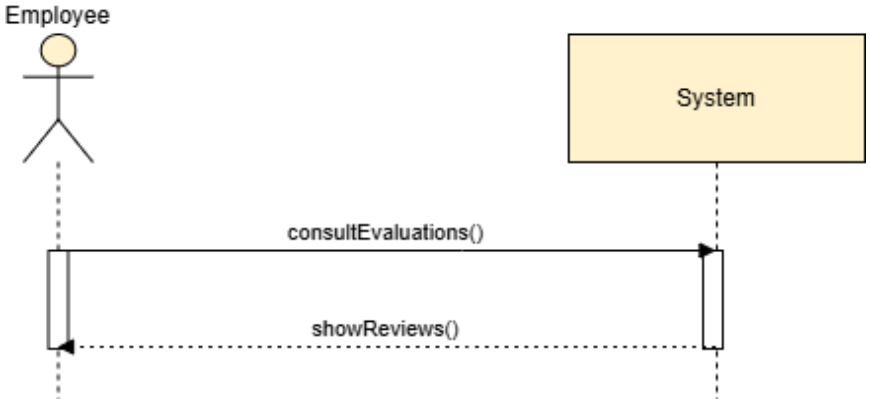


Fig. 13 use case performance evaluations

Figure 14 is the sequence diagram for the employee to consults for his or her own payrolls.

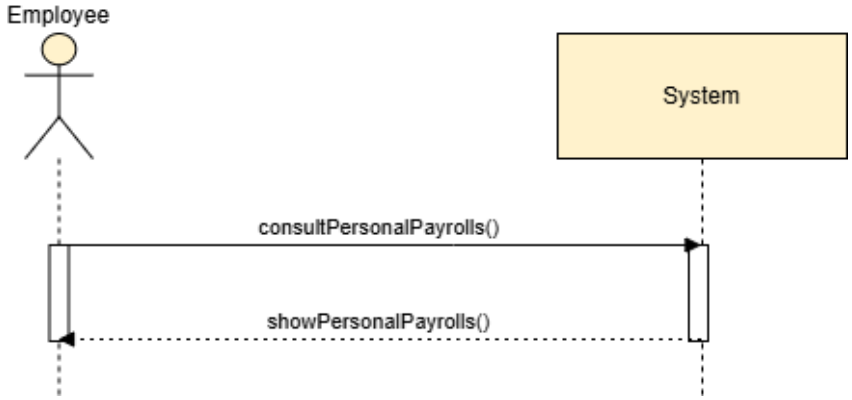


Fig. 14 use case consult payroll

Figure 15 is the sequence diagram for the administrator to manages employee information such as registrations, terminations, and modifications.

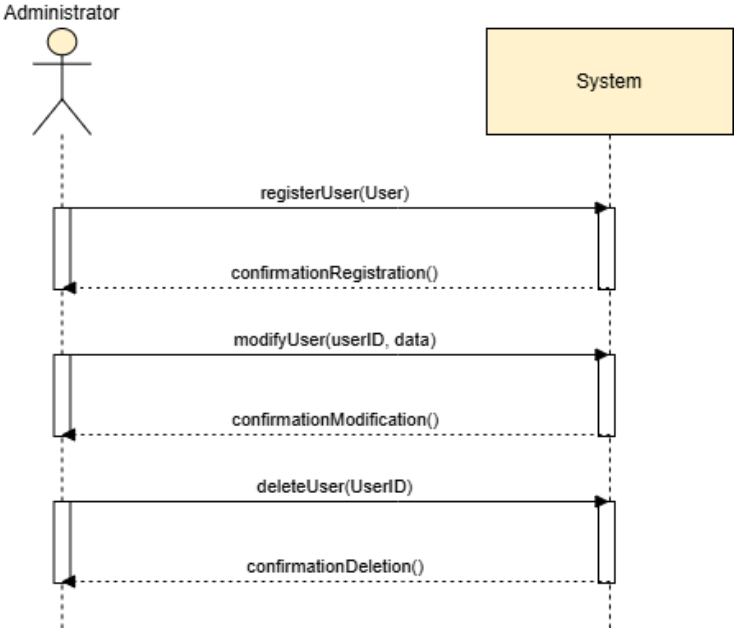


Fig.15 use case personnel management

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Figure 16 is the sequence diagram for the administrator to assigns roles and access permissions.

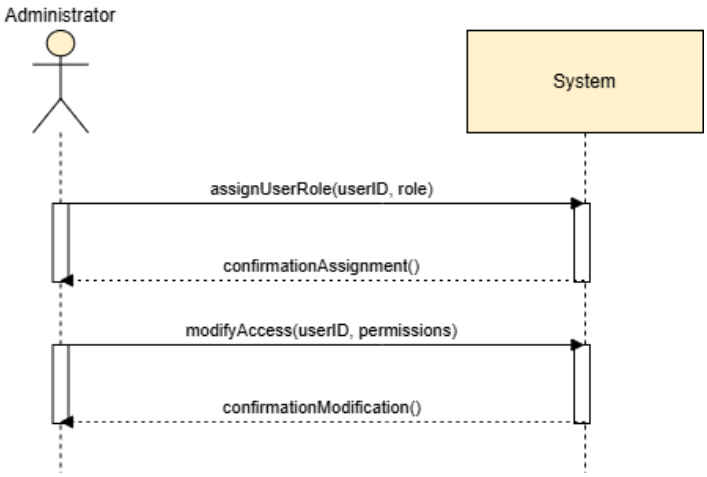


Fig. 16 use case access and roles management

Architectural Design

The system architecture evolved from monolithic and layered patterns to a microservices-based design as indicated in figure 17. Microservices provide high scalability, functional isolation, and independent maintainability—each module (user management, attendance, payroll, evaluations) operates as an autonomous service with its own database, enabling distributed deployment and updates without system-wide disruption (Newman, 2021). This approach proved ideal for AAUCA's need for gradual digital transformation and future expansion.

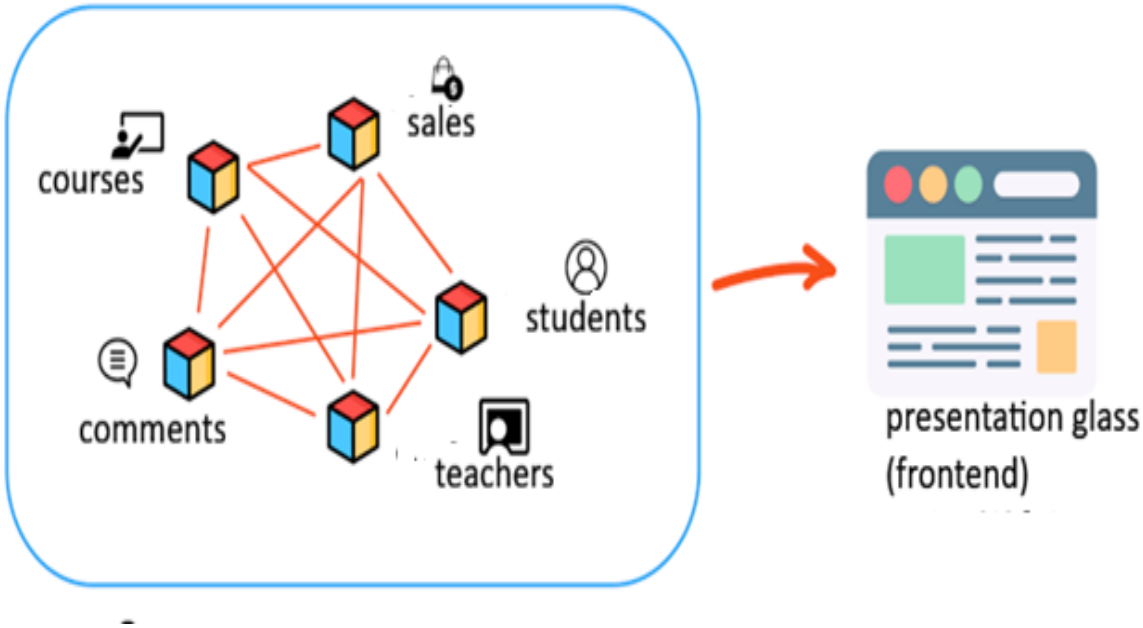


Fig. 17 Microservices architecture

Database Design

The relational database model in figure 18 comprised eight normalized tables ensuring data integrity through primary/foreign key relationships. The Users table served as the central entity, related to Attendances, Evaluations, Permissions, Payrolls, Payroll_History, Personal_Statistics, and Files tables. A System_Actions table provided comprehensive audit logging for security compliance. The data dictionary detailed each field's

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

type, size, constraints, and relationships, serving as a development reference . Normalization reduced redundancy while maintaining referential integrity across all transactional operations [14].

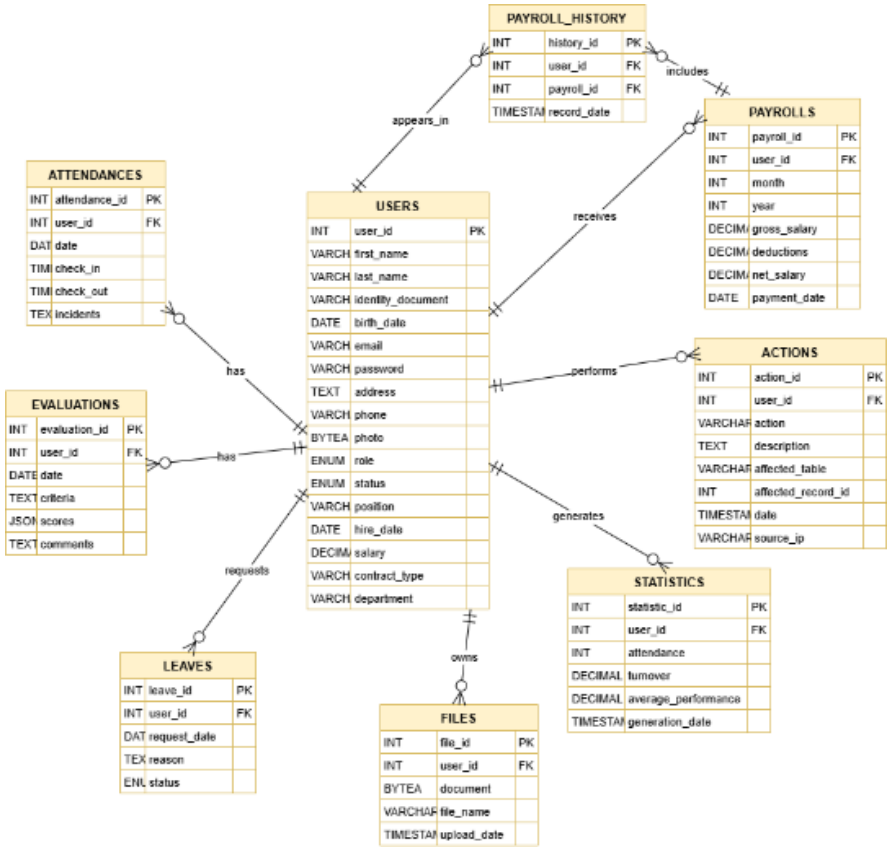


Fig. 18 entity-relation model

RESULTS

System Performance and Outcomes

The deployed AHRAMS system delivered significant improvements. The HR Manager's dashboard Figure 19 provides real-time statistics on total employees, pending permissions, and payroll status, with intuitive navigation to core functions. Employee portals in figure 20 offer personalized access to payroll history, leave balances, performance evaluations, and quick action buttons. The Digitizer interface in figure 21 streamlines document management with OCR processing capabilities and visual progress tracking. The Administrator panel in figure 22 centralizes personnel management, reporting, and system configuration tools.

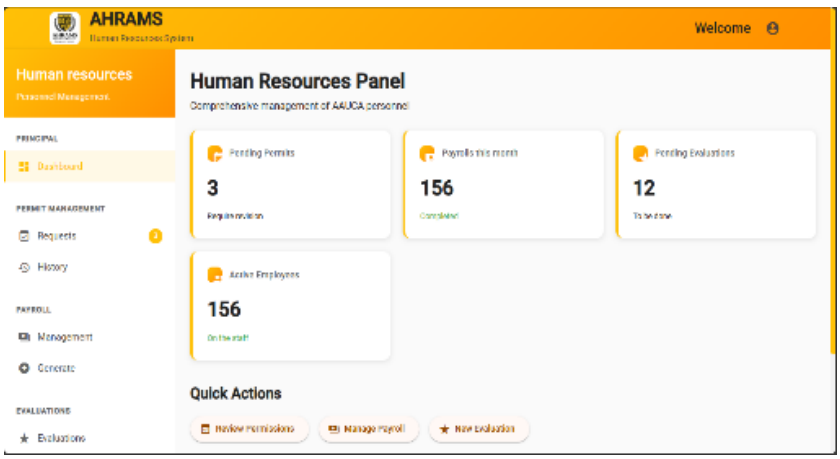


Fig. 19. Final result of the HR manager's home screen

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

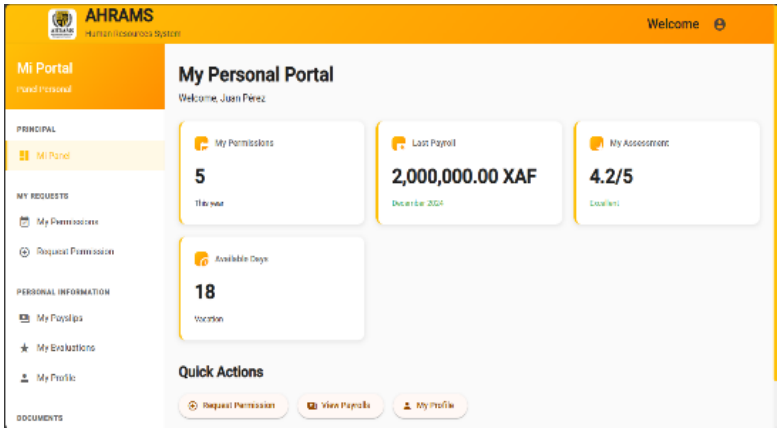


Fig. 20 final result of the employee home screen4

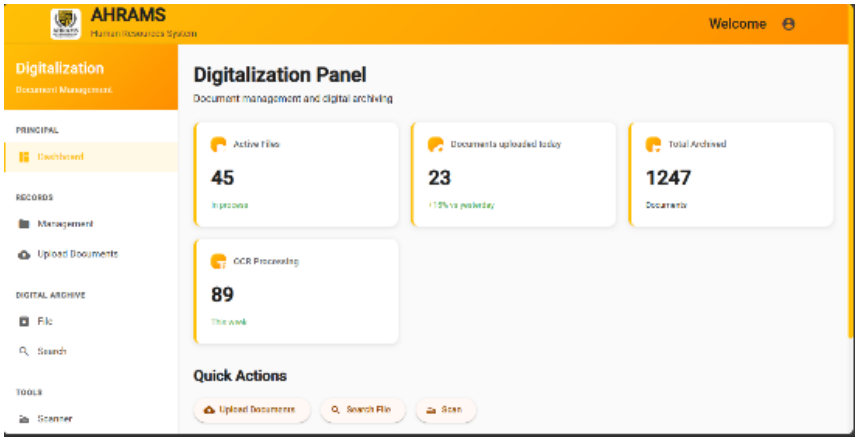


Fig. 21 Final result of the digitizer home page

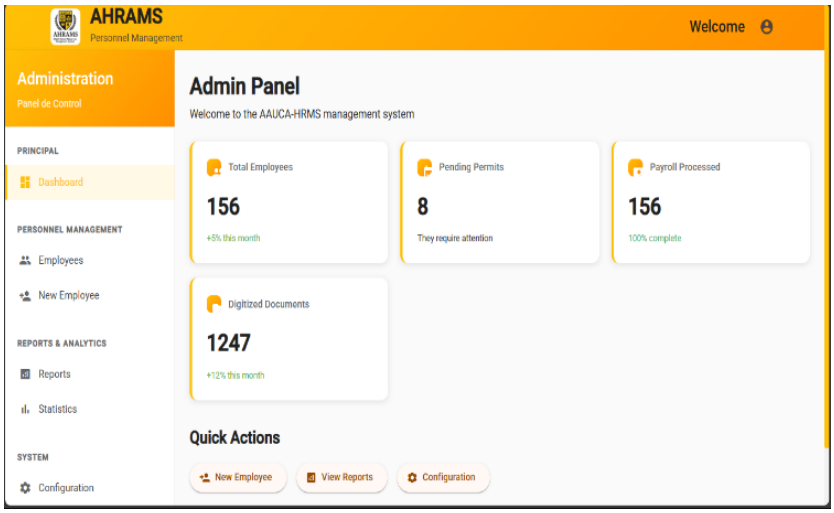


Fig.22 System Administrator Home Screen Result

Quantitative results as indicated in table 1 demonstrate system's impact: payroll processing time per cycle improved by 80%; error rate in payroll calculation was decreased by 90%; employ query resolution time was 85% faster; administrative labour hours per month was reduced by 75%; paper usages was reduced by 90%; payroll approval cycle was 70% faster; employee satisfaction increased by 20%; regulatory compliance accuracy improved by 15%; operational cost per payroll cycle was reduced by 71%.. The microservices architecture enabled independent scaling of high-usage modules (payroll, attendance) during peak periods, while maintaining system stability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

Table 1. payroll system empirical results comparison

Metric	Before Automation (Manual System)	After Automation (Digital Payroll System)	Observed Improvement (%)
Payroll Processing Time per Cycle	6–7 days	4–6 hours	↑ 80% faster
Error Rate in Payroll Calculations	7–10%	0.5–1%	↓ 90% fewer errors
Employee Query Resolution Time	2–3 days	2–4 hours	↑ 85% faster
Administrative Labor Hours per Month	160 hours	40 hours	↓ 75% reduction
Paper Usage (Forms, Reports)	~500 sheets/month	<50 sheets/month	↓ 90% reduction
Payroll Approval Cycle	3–4 days	1 day	↑ 70% faster
Data Retrieval Time (Payroll History)	30–45 minutes	<2 minutes	↑ 95% faster
Employee Satisfaction (Survey Index)	65%	92%	↑ 20% increase
Compliance Accuracy (Tax & Deductions)	80%	99%	↑ 15% improvement
Operational Cost per Payroll Cycle	$1,200	$350	↓ 71% cost savings

CONCLUSION AND RECOMMENDATION

Conclusion

The AHRAMS project successfully demonstrates how modern software engineering principles can address real-world administrative challenges in higher education. By combining agile Scrum methodology, UML modeling, normalized database design, and microservices architecture, the development team delivered a robust, scalable, and flexible HR management system tailored to AAUCA's specific needs. The iterative development approach facilitated continuous stakeholder feedback, ensuring the final product aligned with institutional workflows while remaining adaptable to evolving requirements. The microservices architecture proved particularly valuable, enabling functional isolation and independent deployment that supports long-term maintainability and scalability.

System testing confirmed that all functional and non-functional requirements were met, including role-based access control, data security, responsive design, and performance benchmarks. The implemented solution reduces administrative burden, enhances data accuracy, and provides leadership with timely analytics for evidence-based decision-making. Beyond immediate operational benefits, AHRAMS positions AAUCA as a regional leader in digital transformation, offering a replicable model for other Central African institutions facing similar challenges.

Future Research Directions

While AHRAMS fulfills its core objectives, several enhancement opportunities exist. Integration with external systems—such as accounting platforms, academic management systems, and national social security databases—would create a more comprehensive institutional ecosystem. Implementing advanced analytics and machine learning could enable predictive modeling for employee turnover, automated performance insights, and optimized staffing projections. Developing a dedicated mobile application would improve accessibility for employees and managers, particularly in contexts with limited desktop computer access. Security enhancements, including multi-factor authentication and end-to-end encryption for sensitive financial data, would strengthen

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue V, May 2026

compliance with international data protection standards. Finally, migrating to cloud infrastructure would increase system availability, disaster recovery capabilities, and reduce local hardware dependency.

Recommendations

Based on project outcomes, several recommendations emerge for AAUCA and similar institutions. First, maintain agile methodologies for future software projects, as their iterative nature and stakeholder focus significantly enhance productivity and adaptability. Second, invest in continuous staff training to maximize system utilization and ensure users can leverage advanced features effectively. Third, establish a formal maintenance plan combining corrective updates (bug fixes) and evolutionary enhancements (new features) to preserve system relevance and performance. Fourth, promote microservices architecture for future developments, given its proven effectiveness in managing complexity and enabling gradual modernization. Fifth, implement rigorous database monitoring with automated daily backups to prevent data loss and ensure business continuity. Finally, foster a culture of digital innovation by encouraging cross-departmental collaboration and knowledge sharing about technology adoption.

REFERENCES

1. Papazoglou, M. P. (2012). Service-oriented computing: Concepts, characteristics and directions. Wiley.
2. Wikipedia. (2023, November 15). Sistema de administración de recursos humanos. https://es.wikipedia.org/wiki/Sistema_de_administraci%C3%B3n_de_recursos_humano
3. García, R., & Ruiz, F. (2018). Introducción a la arquitectura de software. Universidad de Sevilla.
4. Newman, S. (2021). Building microservices: Designing fine-grained systems (2nd ed.). O'Reilly Media.
5. Bass, L., Clements, P., & Kazman, R. (2013). Software architecture in practice (3rd ed.). Addison-Wesley Professional.
6. IONOS. (n.d.). Diagrama de casos de uso: Conceptos básicos y ejemplos. IONOS Guía Digital. https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/
7. SAP. (n.d.). SAP SuccessFactors. https://www.sap.com/products/human-capital-management.html
8. Oracle. (n.d.). Oracle human capital management cloud. https://www.oracle.com/human-capital-management/
9. Zoho. (n.d.). Zoho People. https://www.zoho.com/people/N.
10. OrangeHRM. (n.d.). Open source HR software. https://www.orangehrm.com/
11. Mang Lin, I. (2022). Warexpress: Sistema de gestión de stock para superficies de almacenamiento [Trabajo Fin de Grado, Universidad Politécnica de Madrid]. Repositorio Institucional UPM. https://upm.renati.sunedu.gob.pe/handle/20.500.12759/8362
12. Moya Serrat, E. J., & Callejas Pizanan, L. V. (2016). Desarrollo de una aplicación web para gestión del talento humano [Trabajo de Titulación, Universidad Politécnica Salesiana]. UPS. https://dspace.ups.edu.ec/handle/123456789/14024
13. De Bustos Pérez, J. (2022). LaborFlix: Aplicación web de gestión de recursos humanos relacionada con la administración de horarios, equipos y permisos [Trabajo Fin de Grado, Universidad de Valladolid]. UVaDOC. http://uvadoc.uva.es/handle/10324/55240
14. Elmasri, R., & Navathe, S. B. (2021). Fundamentals of database systems (7th ed.). Pearson.
15. Gomaa, Hassan (2011). Software modeling and design : UML, use cases, patterns, and software architectures. Cambridge University Press. 32 Avenue of the Americas, New York, NY 10013-2473, USA