Ceddi Journal of Information System and Technology (JST)
Vol. 4 No. 2 December (2025)
https://doi.org/10.56134/jst.v4i2.133

ISSN: 2829-808X (print)
ISSN: 2829-6575 (online)

Agile Laravel-Based e-HMS for Secure Appointment Booking and Role-Based Clinical Workflow Management

Mardiyya Lawal Bagiwa[*1], Barira Hamisu[2], Raji Abdullahi Egigogo[3], Muhammad Yahaya Gana[4]

[1]*Department of Computer Science and Information Technology, Al-Qalam University, Katsina, Nigeria*
[2,3]*Department of Software Engineering and Cyber Security, Al-Qalam University, Katsina, Nigeria*
[4]*Department of Computer Science and Software Engineering, Global Wealth University Togo, Togo, Nigeria*

Abstract

When appointment scheduling, treatment documentation, and messaging live in separate systems, coordinating tasks across healthcare roles becomes difficult. This study designed, implemented, and evaluated a web-based electronic healthcare management system that integrates these functions for administrators, healthcare providers, and patients. The system was developed iteratively using Laravel and the Model-View-Controller architecture. Requirements-based functional testing compared observed system behavior with predefined outcomes for authentication, appointment booking and approval, treatment documentation, and messaging. All 29 reported test cases met their expected outcomes (100%), including 9 administrator, 9 healthcare provider, and 11 patient scenarios. In the appointment workflow, patients could book appointments and healthcare providers could approve them, as specified in the corresponding test cases. The study provides a documented implementation that links role-specific requirements to integrated healthcare workflows and shows how functional test cases can verify these requirements. The findings establish that the system met the specified requirements for the scenarios tested. Separate evaluations are needed to assess security, scalability, interoperability, and clinical effectiveness. These evaluations would address properties that the reported functional tests were not designed to measure.

Keywords: Electronic Healthcare Management System; Laravel Framework; Medical Records; Modern Technologies; Appointment Booking System.

Received: 31 May 2025 **Revised:** 28 June 2025 **Accepted:** 12 November 2025 **Published:** 31 December 2025

Introduction

Digital transformation has become a defining trajectory of contemporary healthcare systems, driven by rising service demand, increasing data complexity, and the need for timely, coordinated care delivery. International health organizations consistently emphasize that digital health information systems are foundational for strengthening healthcare governance, improving service efficiency, and enabling data-informed clinical decision-making (World Health Organization, 2021). In this context, healthcare institutions' capacity to manage patient information, clinical workflows, and administrative processes through integrated digital platforms is no longer optional; it is structural.

Electronic healthcare management systems (e-HCMS) have been widely adopted to support these objectives by automating core healthcare operations, including patient registration, appointment scheduling, treatment documentation, and provider coordination. Prior research indicates that effective e-HCMS deployment can reduce administrative burden, improve information accessibility, and enhance continuity of care (Kostyshak, 2022). However, empirical evidence also shows that many existing systems remain functionally fragmented, emphasizing data storage or billing functions while offering limited integration between appointment workflows, role-based access, and interactive patient–provider communication (JingleWorks, 2020). As a result, healthcare organizations often operate multiple disconnected applications that fail to support end-to-end service workflows.

In health informatics, Health Information Systems (HIS) are commonly conceptualized as platforms that integrate electronic medical records, operational data, and decision-support functions to improve organizational performance and patient outcomes (Shailendra, 2022). Despite their conceptual maturity, HIS implementations often face persistent challenges in data security, interoperability, and usability, particularly in web-based environments where multiple stakeholder roles interact with sensitive patient information (Chris, 2023). These challenges highlight a structural issue: system designs often treat access control and workflow coordination as secondary considerations rather than as core architectural principles.

Recent state-of-the-art studies illustrate diverse efforts to address specific healthcare challenges through digital solutions. (Akinloye, F. O et al., 2020) proposed an affect-based e-healthcare system that integrates sensor data and fuzzy logic to support children with autism spectrum disorder in underserved areas, demonstrating the potential of specialized system architectures for vulnerable populations. (Tanbeer & Sykes, 2021) developed a web-based patient

*Corresponding author.
E-mail address: mardiyyalawalbagiwa@auk.edu.ng (Mardiyya Lawal Bagiwa)

© [2025] The Author(s). This article is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

portal to extend healthcare services beyond traditional clinical settings, addressing limitations of conventional portals that restrict access to previously registered patients. While these contributions advance domain-specific applications, they primarily focus on specific care scenarios or user groups, leaving broader questions of generalized workflow integration and system-wide validation underaddressed.

Parallel research on electronic medical record systems and digital appointment platforms underscores the operational benefits of digitization, including reduced service delays, improved scheduling accuracy, and enhanced clinical coordination (Garko & Mahmud, 2017; Edeh et al., 2024). Nevertheless, adoption studies continue to report barriers such as user resistance, privacy concerns, and inadequate regulatory support, especially in resource-constrained healthcare environments (Ukata & Wechie, 2019; Obimba et al., 2022). These findings suggest that technical implementation alone does not guarantee effectiveness; instead, healthcare systems require architectures that explicitly align workflow logic, access roles, and verification mechanisms within a coherent operational framework.

Despite extensive literature on e-health platforms, a clear research gap persists concerning implementable, role-based electronic healthcare management systems that integrate appointment scheduling, treatment documentation, and user communication within a single, verifiable web architecture. Existing studies frequently emphasize conceptual frameworks, isolated functional modules, or narrowly scoped applications without demonstrating end-to-end workflow conformity through systematic validation. Moreover, limited attention has been given to documenting how requirement specifications translate into functional system behavior that can be empirically verified and reproduced.

To address this gap, the present study designed and implemented a web-based electronic healthcare management system using the Laravel framework, structured around distinct workflows for administrators, healthcare providers, and patients. This research makes a twofold contribution. Methodologically, it operationalizes requirements engineering and agile development principles into a role-based healthcare workflow architecture that prioritizes access governance and process coherence. Practically, it provides requirement-driven functional validation of core healthcare processes through systematic unit testing, extending prior e-health research from conceptual or descriptive accounts to reproducible, deployment-oriented system implementations with relevance beyond local institutional contexts.

Methodology

This study employed a design-and-implementation research design grounded in an agile software development approach to support iterative refinement of requirements and progressive validation of system functionality. The agile model structured development activities into recurring cycles of requirement analysis, design, implementation, and verification, enabling continuous alignment between stakeholder needs and system behavior (Upadrista, 2015). Figure 1 illustrates the agile development model adopted in this study, highlighting the iterative nature of requirement elicitation, system design, implementation, and testing phases.

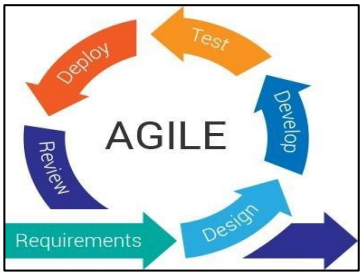


Figure 1. Agile Methodology

Research Subjects and Requirements Engineering

The analytical subjects of this study comprised three operational roles: administrators, healthcare providers, and patients. These roles informed the requirements engineering process, which was conducted as a socio-technical activity to identify functional expectations, workflow dependencies, and access constraints across stakeholder interactions. This approach ensured that requirements reflected real operational roles rather than abstract system functions (Daun et al., 2023).

System Design and Modeling

System design translated the elicited requirements into structured representations to guide implementation and maintain architectural coherence. The team first developed a structural navigation model to define the primary system interfaces

and functional menus, including authentication, appointment booking, service information, and departmental access. This structural representation ensured logical consistency between user navigation and underlying system functions. Figure 2 presents the system's structural design, depicting the main functional components and the navigation flow accessible through the system interface.

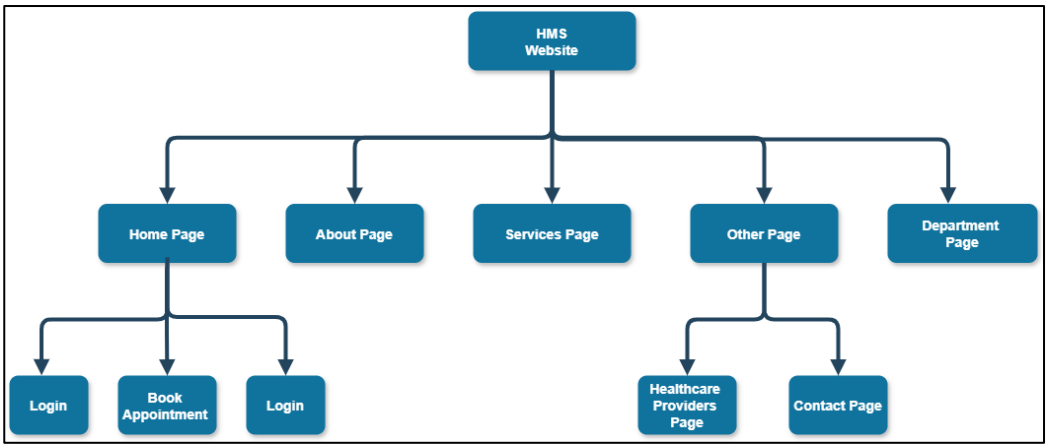


Figure 2. Structural Design of the System

To further formalize system interactions, a use case model was constructed to capture role-specific actions and permissible operations within the system. The model delineated the interactions of patients, healthcare providers, and administrators with core system functions, including appointment management, treatment documentation, and communication workflows. This use-case representation ensured the access-control logic aligned explicitly with operational responsibilities. Figure 3 shows the system's use case diagram, illustrating role-based interactions and functional boundaries among system actors.

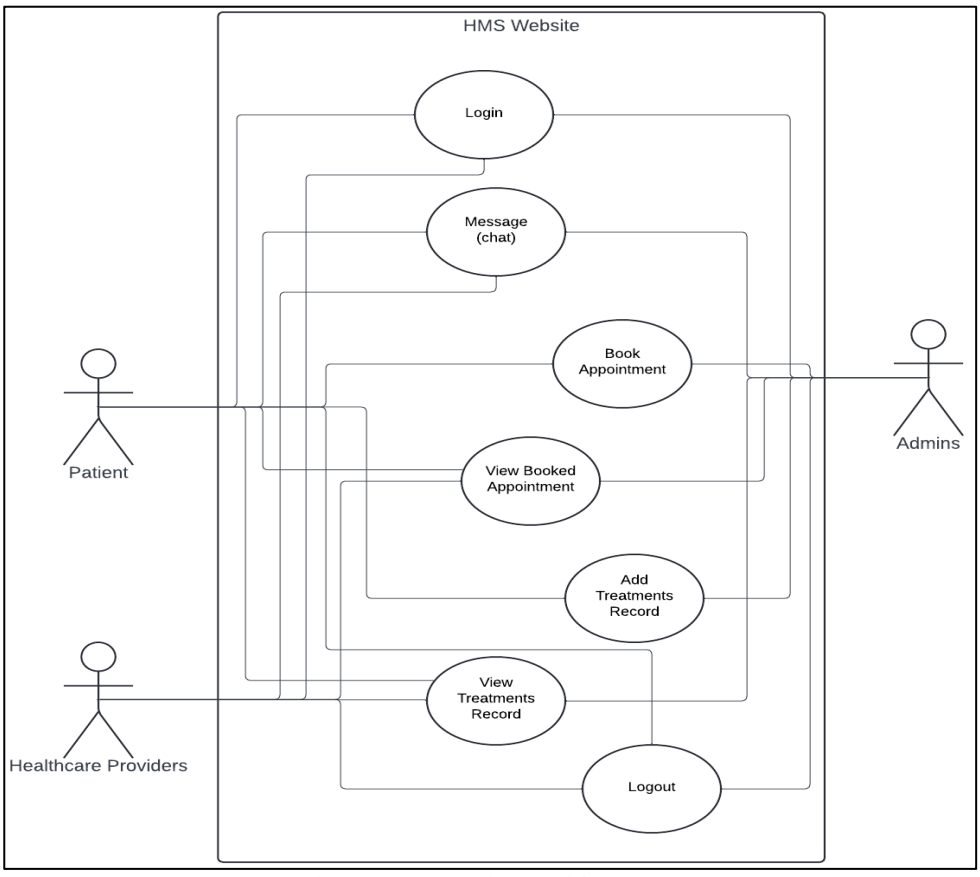


Figure 3. Use Case Diagram of the System.

Data Modeling

The system's data architecture was specified using an entity–relationship model that defines the core data entities and their relationships. These entities included users, appointments, treatments, and messaging records, structured to support role-based access and workflow traceability. The entity-relationship model served as a blueprint for database implementation and ensured consistency between data structures and application logic. Figure 4 presents the system's entity-relationship diagram, illustrating the interconnections among the primary data entities underpinning its functionality.

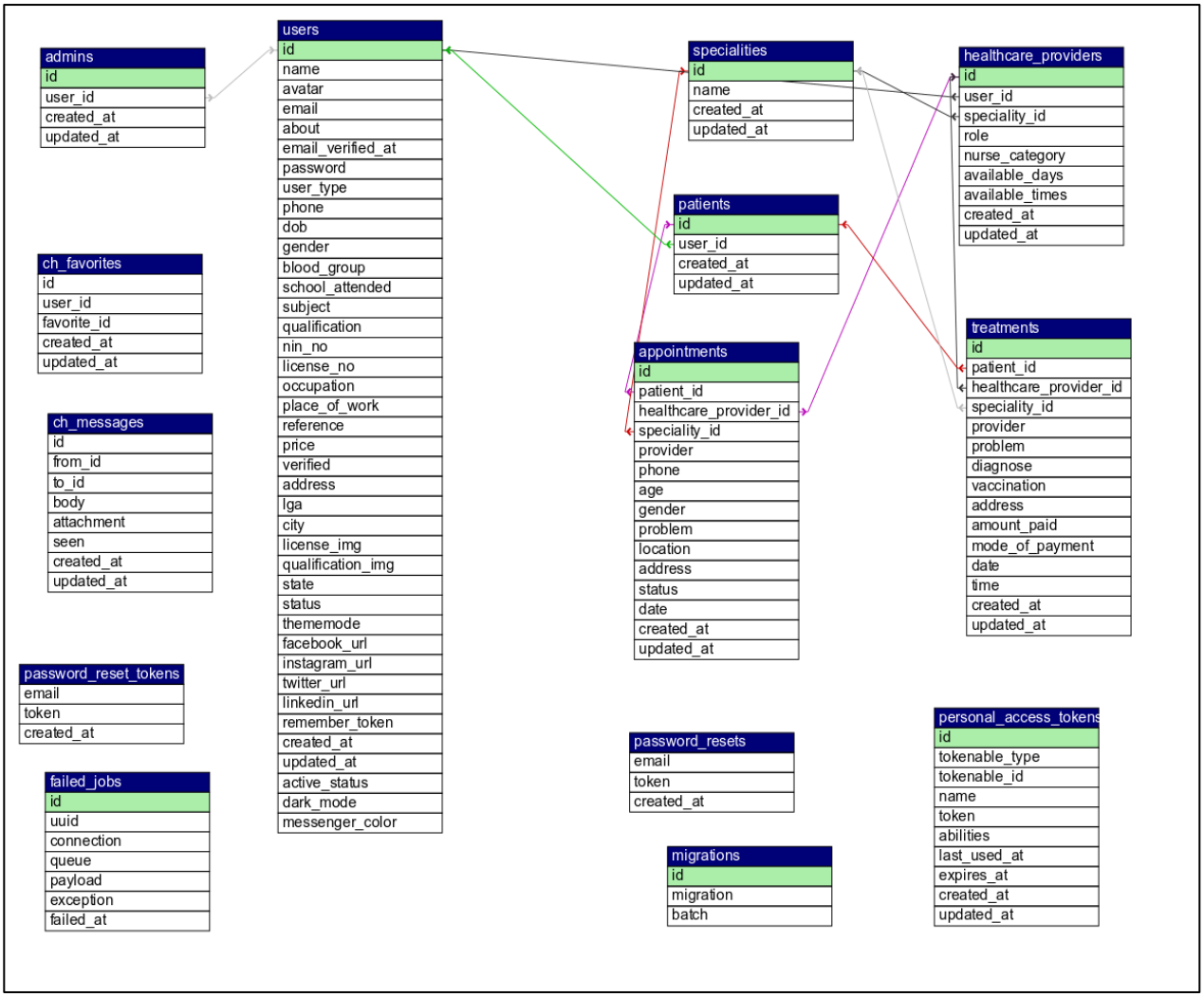


Figure 4. Entity Relationship Diagram of the System

Results and Discussion

Results

The electronic healthcare management system was successfully implemented as a web-based application using the Laravel framework, which applies the Model–View–Controller (MVC) architectural pattern to separate data management, application logic, and user interfaces (Laravel Documentation, 2024). This architectural separation facilitated modular development and supported systematic functional verification. The system operated within a standard web server environment that handled HTTP/HTTPS request–response interactions between client browsers and server-side application components, consistent with established web architecture principles (Fielding et al., 1999). Functional verification used requirement-driven unit testing to assess conformity between specified requirements and implemented features. The evaluation focused on three functional modules corresponding to administrator, healthcare

provider, and patient roles. Each test case examined authentication, workflow execution, data manipulation, and communication functions using a pass/fail criterion. Table 1 presents the unit testing results for the administrator module. All nine test cases, covering authentication, profile management, specialty configuration, appointment approval, treatment management, user management, messaging, and logout functionality, were executed successfully, indicating full conformity with specified administrative requirements.

Table 1. Functional Verification Results for Administrator Module

Test ID	Function	Description	Expected result	Actual result	Status
1	Login	Admin tries to log in to the system	Admin logs in to the system and is taken to their dashboard	Admin was logged in to the system and redirected to their dashboard	Successful
2	Profile	Admin to access their profile page and make changes	Admin to log in and access their profile page for changes	Admin was able to log in and access their profile page and was able to make changes	Successful
3	Manage Specialties	Admin tries to add, edit, update, and delete specialties in the system	Admin can create, edit, update, and able to delete new or existing specialties in the system	Admin is able to create and manage the specialties category	Successful
4	Manage Appointment	Admin tries to add, edit, update, and to delete appointments and approve them in the system	Admin adds, edits, updates, and deletes appointments and also approves them in the system	Admin was able to manage appointments and also approve them in the system	Successful
5	Manage Treatment	Admin tries to add, edit, update, and delete treatments in the system	Admin can add, edit, update, and delete treatments in the system	Admin is able to manage treatments in the system	Successful
6	Manage Healthcare Providers	Admin tries to add, edit, update, and delete healthcare providers in the system	Admin can add, edit, update, and delete healthcare providers in the system	Admin was able to manage healthcare providers in the system	Successful
7	Manage Patients	Admin tries to add, edit, update, and delete patients in the system	Admin can add, edit, update, and delete patients in the system	Admin is able to manage patients in the system	Successful
8	Chat	Admin tries to chat with any of the users (health care providers or patients) in the system	Admin chats with any of the users (health care providers or patients) in the system	Admin was able to chat with any of the users (health care providers or patients) in the system	Successful
9	Logout	Admin tries to log out of the system	Admin logs out of the system	Admin was logged out of the system	Successful

Table 2 summarizes the unit testing outcomes for the healthcare provider module. The results indicate that all tested functions, including account creation, profile updates, appointment review and approval, treatment access, patient list viewing, messaging, and logout, performed as specified, with no functional deviations detected.

Table 2: Healthcare Providers Requirement

Test ID	Function	Description	Expected result	Actual result	Status
1	Login	Healthcare providers try to log in to the system	Healthcare providers log in to the system and are taken to their dashboard	Healthcare providers were logged in to the system and redirected to their dashboard	Successful
2	Create account	Healthcare provider to create in the system	Healthcare Professionals create a new account in the system	Healthcare professionals are able to create a new account in the system	Successful
3	Profile	Healthcare provider to access their profile page and make changes	Healthcare provider need to log in and access their profile page for changes	The healthcare provider was able to log in and access their profile page and was able to make changes	Successful
4	View Appointments Scheduled	Healthcare provider to view their scheduled appointments made by patients	Healthcare provider view their scheduled appointments made by patients	The healthcare provider was able to view appointments scheduled for them by their patients	Successful
5	Approve Appointment	Healthcare providers to approve appointments scheduled for them by the patient	Healthcare providers approve the appointment scheduled for them	The healthcare provider was able to approve the appointment scheduled for them	Successful
6	View Treatments	Healthcare providers can view treatments done after the	Healthcare providers can view treatments done after	The healthcare provider can view treatments added to the system by the patient.	Successful

Test ID	Function	Description	Expected result	Actual result	Status
		patient adds them to the system.	the patient adds them to the system.		
7	View and reply to chat messages	Healthcare provider to view messages and reply in the system	Healthcare providers view messages and reply or chat in the system	The healthcare provider was able to view the message and reply or chat in the system	Successful
8	View patient	Healthcare to see/view the list of attended or treated patients in their home dashboard	Healthcare sees/views the list of attended or treated patients in their home dashboard	Healthcare was able to see/view the list of attended or treated patients in their home dashboard	Successful
9	Logout	Healthcare providers try to log out of the system	Healthcare provider logs out of the system	The healthcare provider was logged out of the system	Successful

Table 3 reports the functional verification outcomes for the patient module. Eleven test cases were evaluated, encompassing account creation, authentication, appointment booking and status updates, treatment record entry and review, healthcare provider browsing, messaging, and logout. All test cases executed successfully, demonstrating consistent system behavior across patient-oriented workflows.

Table 3: Patient Requirement

Test ID	Function	Description	Expected result	Actual result	Status
1	Login	The patient tries to log in to the system	Patient logs in to the system on their dashboard	The patient was logged into the system and redirected to their dashboard	Successful
2	Create account	Patient to be created in the system	The patient creates a new account in the system	The patient was able to create a new account in the system	Successful
3	Profile	Patient to access their profile page and make changes	Patient logs in and accesses their profile page for changes	Patient was able to log in and access their profile page and was able to make changes	Successful
4	Book Appointment	Patient are to book appointments with their desired healthcare provider	Patient books appointments with their desired healthcare provider	The patient was able to book appointments with their desired healthcare provider	Successful
5	View appointment booked	Patient to view all the appointments they have booked	Patients view all the appointments they have booked	The patient was able to view all the appointments they had booked	Successful
6	Add treatment	Patient to add the treatment done or given to them by the Healthcare provider in the system	Patient adds the treatment done or given to them by the Healthcare provider in the system	Patient was able to add the treatment done or given to them by the Healthcare provider in the system	Successful
7	View treatments	Patient to view all the treatments they added, which are given to them by the Healthcare providers	Patients view all the treatments they have added, which are given to them by the Healthcare providers	Patient was able to view all the treatments they added, which were given to them by the Healthcare providers	Successful
8	View Healthcare Providers	Patient can view all the healthcare providers in the system and also access their profiles	Patients can view all the healthcare providers in the system and also access their profiles	The patient was able to view all the healthcare providers in the system and also access their profiles	Successful
9	Update Appointment Status	Patient to update appointment status to complete or not complete	Patient updates appointment status to complete or not complete	Patient was able to update the appointment status to complete or not complete	Successful
10	Chat or message healthcare providers	Patient to chat or send messages to the healthcare provider in the system	Patient chat or send messages to the healthcare provider in the system	The patient was able to chat or send messages to the healthcare provider in the system	Successful
11	Logout	Patient tries to log out of the system	Patient logs out of the system	Patient was logged out of the system	Successful

In addition to functional verification, the system interface supports role-specific interaction flows. Figure 5 illustrates the system home page, which provides centralized navigation to core services. Figure 6 displays the login interface used for authenticated access. Figures 7 and 8 present the registration interfaces for patients and healthcare providers, respectively, reflecting differentiated onboarding processes. Figure 9 shows the administrator dashboard, which consolidates system management functions within a single control interface.



Figure 5. Healthcare System Home Page

Figure 6 depicts the Healthcare system login page. In the middle of the page, an input area lets users enter their email and password. Below that is a 'Login' button that users can click to enter the system. The clean, intuitive page design makes it easy for users to access the system and perform hospital management tasks.

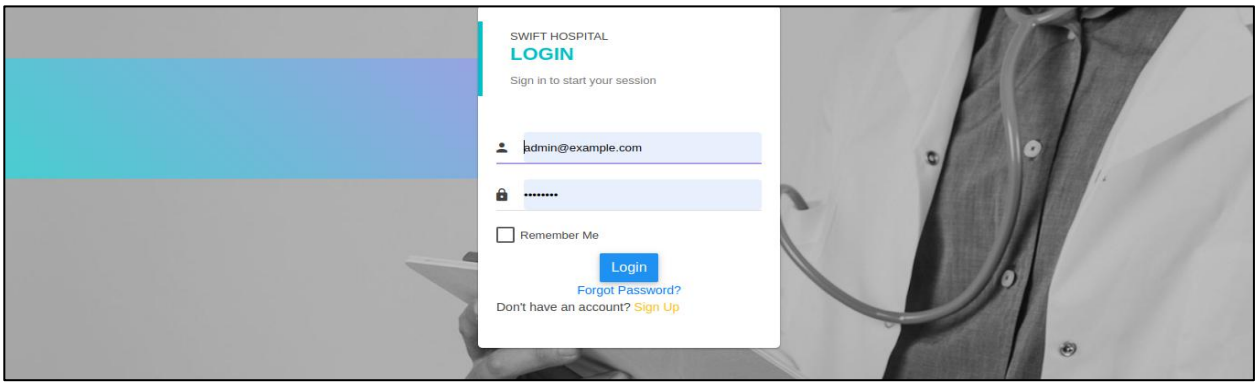


Figure 6. Healthcare System Login Page

Figure 7 displays the patient registration page. This page allows patients to register or create a new account.

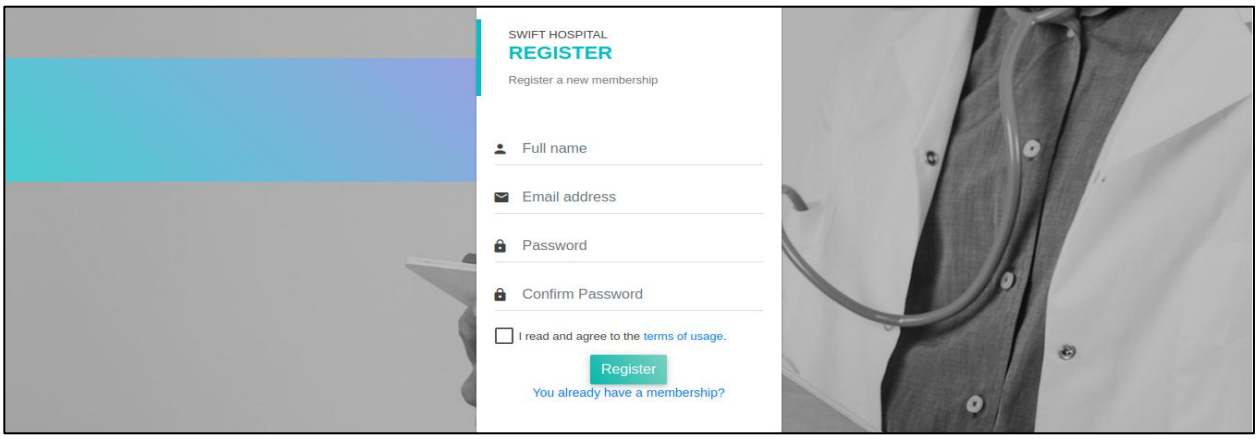


Figure 7: Registration Page for Patient

Figure 8 depicts the Registration page for Healthcare providers, where new healthcare providers can register or create a new account.

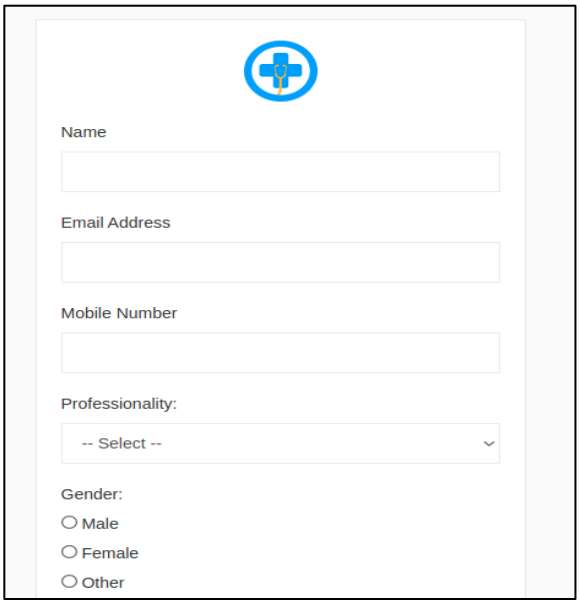


Figure 8: Registration Page for Healthcare Providers

Figure 9 shows the system dashboard, the admin dashboard for Healthcare management. On this page, the admin can access the sub-admin menus: appointments, treatments, and account settings.

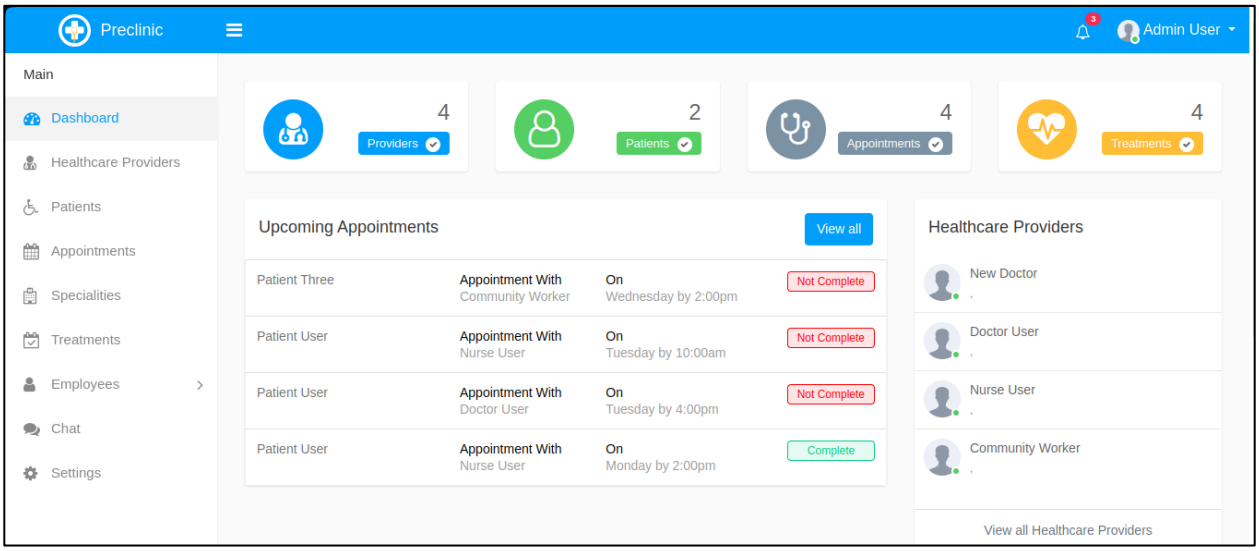


Figure 9. System Dashboard

Discussion

The results demonstrate that the implemented system achieved full functional conformity with its specified requirements across all user roles. This finding is significant in light of previous research indicating that many healthcare information systems fail to operationalize end-to-end workflows, instead offering isolated functionalities that require manual coordination (Kostyshak, 2022). By validating all role-specific functions through requirement-driven testing, the present system addresses a recurrent limitation in e-health platform development: the gap between conceptual design and verifiable implementation.

The successful execution of appointment booking and approval workflows across patient and healthcare provider modules aligns with prior studies emphasizing the role of digital scheduling systems in reducing service delays and

improving coordination between patients and providers (Garko & Mahmud, 2017). Unlike systems that restrict scheduling to administrative intermediaries, the role-based workflow implemented here enables direct interaction while preserving authorization boundaries, thereby supporting operational efficiency without compromising governance.

From a system architecture perspective, the MVC-based framework supported modular verification and reduced coupling between interface design and business logic. This architectural choice is consistent with software engineering practices that advocate separation of concerns to improve maintainability and scalability in web-based systems (Contributors, 2024). Although this study did not evaluate performance and scalability metrics, the modular structure provides a foundation for future optimization and extension.

Including treatment documentation and in-system messaging further differentiates the system from appointment-only platforms. Prior research notes that limited communication channels and fragmented record-keeping reduce continuity of care and increase reliance on informal or external messaging tools (Tanbeer & Sykes, 2021). By integrating these functions within a single authenticated environment, the system supports traceable interactions and reduces the risk of information loss, although further evaluation is required to assess clinical impact and user experience.

Despite these contributions, interpret the findings within the study's scope. Functional conformity does not directly imply usability, clinical effectiveness, or regulatory compliance. Previous studies have highlighted that healthcare system adoption is influenced by factors such as user acceptance, privacy concerns, and institutional readiness (Ukata & Wechie, 2019; Obimba et al., 2022). Consequently, future research should complement functional testing with usability evaluation, security assessment, and interoperability analysis to determine deployment viability in real healthcare settings.

Conclusions and Suggestions

Conclusions

This study presented a role-based electronic healthcare management system that integrates appointment scheduling, treatment documentation, and user communication within a single web-based architecture. Requirement-driven functional verification showed full conformity between the specified workflows and the implemented system behavior across the administrator, healthcare provider, and patient modules. Beyond confirming technical feasibility, these findings contribute to the health informatics literature by showing that explicit role modeling and workflow alignment can reduce functional fragmentation, a common problem in web-based healthcare systems.

From a theoretical perspective, the study extends existing discussions on healthcare information systems by operationalizing requirements engineering and agile development principles into a coherent, verifiable system architecture. Rather than treating access control and workflow coordination as peripheral concerns, the proposed design embeds them as core architectural elements, thereby offering a structured reference for analyzing workflow integrity in digital health platforms.

From a practical standpoint, the implemented system provides a reproducible blueprint for healthcare organizations transitioning from manual or partially digitized processes to integrated web-based management solutions. The emphasis on functional validation enhances deployment readiness by ensuring that system features align with operational requirements before large-scale adoption. Collectively, the study advances understanding of how modular, role-based architectures can support scalable, secure healthcare management across diverse institutional contexts.

Suggestions

Future work should extend the current system beyond functional conformity to comprehensive deployment evaluation. First, strengthen security mechanisms through multi-factor authentication, fine-grained authorization policies, and audit logging to better align with healthcare information security standards. Second, empirical user studies involving healthcare providers and patients should assess usability and workload impact to evaluate acceptance and interaction efficiency. Third, interoperability capabilities can be introduced by mapping core data entities to established health information standards, enabling integration with external clinical systems and regional health infrastructures. Finally, structured training programs and operational guidelines should accompany system deployment to support organizational readiness and sustainable use.

Acknowledgments

The authors acknowledge assistance with preparing the test environment and reviewing appointment workflows for patients, healthcare providers, and administrators.

Funding Statement

This research received no external funding.

Author Contributions

Mardiyya Lawal Bagiwa led the study design, specified the requirements for the three user roles, and prepared the initial manuscript. Barira Hamisu contributed to the Laravel implementation of the appointment booking and approval workflows. Raji Abdullahi Egigogo prepared and executed the role-specific functional tests and recorded the observed outcomes. Muhammad Yahaya Gana interpreted the test results and revised the discussion to clarify the scope of the findings. All authors reviewed and approved the final manuscript.

Research Ethics and Informed Consent

The evaluation used predefined scenarios to test appointment booking and approval, treatment documentation, and messaging across the three user roles. Testing used simulated accounts and synthetic records and did not involve human participants or identifiable patient information. Ethics approval and informed consent were therefore not applicable.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

Tables 1–3 report the expected and observed outcomes for all 29 functional test cases, grouped by administrator, healthcare provider, and patient roles. The article does not include the application source code.

Declaration of Generative AI Use

OpenAI assisted in revising the abstract, including the study's contribution and the interpretation of the functional test results. The authors take full responsibility for the final manuscript.

References

Akinloye, F. O., Obe, O., & Boyinbode, O. (2020). Development of an affective-based e-healthcare system for autistic children. Scientific African, 9, e00514. https://doi.org/10.1016/j.sciaf.2020.e00514

Brook, C. (2018, June 18). What is a health information system? Fortra's Digital Guardian. https://www.digitalguardian.com/blog/what-health-information-system

Daun, M., Grubb, A. M., Stenkova, V., & Tenbergen, B. (2023). A systematic literature review of requirements engineering education. Requirements Engineering, 28(2), 165–196. https://doi.org/10.1007/s00766-022-00381-9

Edeh, S. E., Emewu, B. M., Edeh, M. I., Ikporo, S. C., & Ugwu, C. F. (2024). Design and implementation of an SMS and email based interactive online medical record management system. IOSR Journal of Computer Engineering, 26(2), 50–58.

Fielding, R. T., Gettys, J., Mogul, J. C., Frystyk, H., Masinter, L., Leach, P. J., & Berners-Lee, T. (1999). Hypertext Transfer Protocol—HTTP/1.1 (RFC 2616). Internet Engineering Task Force. https://doi.org/10.17487/RFC2616

Garko, A. B., & Mahmud, U. (2017). Design and implementation of outpatient management system. International Journal of Advanced Academic Research (Science, Technology & Engineering), 3(6), 13–22.

Jungleworks. (2020). The complete blueprint for healthcare management solutions (White paper). https://jungleworks.com/download/Healthcare-Management-Solutions.pdf

Kostyshak, N. (2022, March 22). Healthcare management system (HMS): Features and benefits. Otakoyi. https://otakoyi.software/blog/healthcare-management-system-features-and-benefits

Laravel Documentation. (2024). Laravel: The PHP framework for web artisans. https://laravel.com/docs

Obimba, K. C., Arize, I. N., Ezenduka, C. C., & Kanayo, K. A. (2022). Examination of health record documentation and management practices among record personnel in two tertiary facilities in Enugu State, Nigeria. African Journal of Health Economics, 11(1), 117–129. https://doi.org/10.35202/AJHE.2022.1110117

Shailendra, S. (2022, June 16). What is health information system and its significance in the healthcare sector? Mobisoft Infotech. https://mobisoftinfotech.com/resources/blog/importance-of-health-information-system/

Tanbeer, S. K., & Sykes, E. R. (2021). MyHealthPortal—A web-based e-healthcare web portal for out-of-hospital patient care. Digital Health, 7, 2055207621989194. https://doi.org/10.1177/2055207621989194

Ukata, P. F., & Wechie, B. A. (2019). Electronic record management and national development: A case of Nigeria. International Journal of Social Sciences and Management Research, 5(7), 76–90.

Upadrista, V. (2015). Agile methodology. In The art of consultative selling in IT (pp. 99–106). CRC Press. https://doi.org/10.1201/b18065-15

World Health Organization. (2021). Global strategy on digital health 2020–2025. World Health Organization. https://www.who.int/publications/i/item/9789240020924