

A Three-Tier, Role-Based Architecture for Electronic Patient Records Management in Nigerian Secondary Hospitals: Design and Specification

CHUKWUMA EMMANUEL CHINENYE¹, MICHAEL OGHAE IGHOFIOMONI²

¹⁻² Department of Computer Engineering, Southern Delta University, Delta State, Nigeria

Abstract- Many secondary hospitals in Nigeria still use paper-based patient records, which are linked to redundant files, slow retrieval times, and disjointed departmental care coordination. The design and technical specifications of a Hospital Management System (HMS) with integrated electronic patient records are presented in this paper. The system is based on a three-tier client-server architecture that consists of a PostgreSQL relational data layer, a Node.js/Express.js application layer, and a React.js presentation layer. Registration, appointment scheduling, clinical consultation, pharmacy, laboratory, billing, reporting, and access control are the eight integrated functional modules that make up the system. These modules operate on a relational schema consisting of seventeen tables that have been normalized to the third normal form. The NIST RBAC standard serves as the foundation for the six-role role-based access control (RBAC) paradigm used in access control. The suggested design fully addresses locally relevant requirements, such as local health-insurance billing integration, that current platforms only partially address, according to an organized, criteria-based comparison against three widely used open-source platforms, OpenMRS, Bahmni, and HospitalRun. The work is provided as a verified architectural blueprint and specification for installation and subsequent empirical assessment in a Nigerian secondary-hospital scenario; it has not yet been benchmarked on an operational deployment as a design-stage contribution.

Keywords: Hospital Management System, Electronic Patient Records, Three-Tier Architecture, Role-Based Access Control, Database Normalization, Health Informatics, Nigeria

I. INTRODUCTION

Despite rising patient numbers and increasingly multidisciplinary treatment pathways, paper-based record-keeping continues to be the predominant paradigm of clinical documentation in the majority of

Nigerian general and secondary hospitals. The health informatics literature has extensively documented the repercussions of this dependency, which include lost or duplicated patient files, delays at the point of service, and a lack of real-time clinical information sharing between departments. The technical design artifacts required for smaller secondary facilities like complete schemas, role hierarchies, and locally relevant billing workflows, remain sparsely documented in the published literature specific to the Nigerian context, despite the fact that international open-source platforms like OpenMRS have shown quantifiable gains when deployed in Nigerian teaching hospitals (Seebregts et al., 2009). The architectural design and specifications of a Hospital Management System (HMS) with electronic patient records are described in this article. The system was created as a comprehensive, implementation-ready blueprint rather than a deployed product. The specific goals are to: (i) design a three-tier client-server architecture suitable for the staffing and connectivity limitations of a secondary hospital in Nigeria; (ii) create a six-role access-control model and a normalized relational schema covering the essential clinical and administrative workflows of such a facility; and (iii) assess the final design's completeness using a structured functional-completeness comparison against three well-known open-source HMS platforms.

II. RELATED WORK

With the OpenMRS Implementers Network created especially to support its expanding base of country-level implementations, the OpenMRS platform which was coordinated by the Regenstrief Institute and Partners in Health remains the most thoroughly documented open-source medical record system used

throughout African health facilities (Seebregts et al., 2009). Many later locally developed systems, such as a number of recent Nigerian university-led hospital and patient management system projects that adopt a similar three-tier, web-based structure using modern JavaScript frameworks, have been influenced by its architecture and configurable data model.

The security posture chosen for the presentation and application tiers in this design is directly influenced by the findings of Kruse et al. (2017), who examined the technical, administrative, and physical safeguards reported across the electronic health record literature and discovered that access control and encryption were the most frequently cited technical categories. In a different study, Adenuga et al. (2017) examined the infrastructure, connection, and incentive obstacles to clinician-level technology adoption in Nigeria. They emphasized the significance of a browser-based, low-bandwidth-tolerant client as opposed to a more complex installed program.

The current work differs from the body of prior research due to two design restrictions. First, adoption hurdles or an established system without a completely defined, standardized schema and role hierarchy appropriate for independent replication are

the most common descriptions found in published HMS design studies located in Nigeria. Second, none of the reviewed open-source platforms were designed against Nigeria's current data-protection regime: the Nigeria Data Protection Act, 2023 (NDPA), which received presidential assent on 12 June 2023 and formally superseded the earlier Nigeria Data Protection Regulation (NDPR) 2019 issued by the National Information Technology Development Agency. The present design was originally specified against the NDPR 2019 framework and has been reviewed here against the now-current NDPA 2023 obligations for consent capture, audit logging, and breach notification enforced by the Nigeria Data Protection Commission (NDPC).

III. SYSTEM ARCHITECTURE

A. Three-Tier Design

In order to achieve scalability, maintainability, and security in business web applications, the system uses a three-tier client-server architecture that divides display, application, and data concerns into independently deployable layers (Fowler, 2002). Table 1 summarises the technology stack selected for each tier and the justification for each component.

Table 1: Summary of Technology Stack

Layer	Technology	Role in Architecture
Presentation	React.js (SPA) + Tailwind CSS	Renders departmental interfaces; communicates with the application tier via a JSON REST API
Application	Node.js + Express.js	Enforces business logic, input validation, session management, and API routing
Application (security)	JSON Web Tokens (RFC 7519); bcrypt	Stateless authentication and adaptive password hashing
Data	PostgreSQL 15 + Sequelize ORM	Relational storage in third normal form; parameterised queries
Infrastructure	Managed cloud hosting; TLS 1.3	Availability, automated backup, and encryption in transit

Given the limited on-site IT support typical of secondary hospitals, the presentation tier is a single-page application built with React.js that communicates with the application tier exclusively thru a JSON REST API, enabling any department to access the system from a standard browser without locally installed client software. The application

layer, which is implemented in Node.js and Express.js, handles authentication using stateless JSON Web Tokens (JWT; RFC 7519) and hashes passwords using bcrypt. It also enforces business rules and verifies incoming data. The data tier is a PostgreSQL relational database hosted on a managed

cloud instance with automated backup and point-in-time recovery.

B. Role-Based Access Control

According to the proposed NIST RBAC standard (Ferraiolo et al., 2001), access control is based on a Role-Based Access Control (RBAC) paradigm, in which each user's capabilities are restricted to the minimal set of tasks necessary for their job function. Six roles are defined, summarised in Table 2.

Table 2: Summary of Users Role

Role	Accessible Modules	Key Restriction
Super Admin	All modules plus system configuration	No patient-care functions
Administrator	Registration, appointments, billing, reports	No clinical documentation access
Doctor	Clinical notes, lab requests, prescriptions, history	Cannot modify billing or admin data
Nurse	Vital signs, nursing notes, drug administration	Cannot prescribe or order laboratory tests
Pharmacist	Pharmacy module, drug inventory, prescriptions	Cannot modify clinical notes
Lab Scientist	Lab requests, results entry, specimen tracking	Cannot access billing or prescriptions

C. Functional Modules

Patient registration and appointment scheduling; clinical consultation, which is organized around the Subjective-Objective-Assessment-Plan (SOAP) documentation format with an embedded ICD-10 lookup; pharmacy and drug dispensing, which includes a first-expiry-first-out inventory rule; laboratory request and results management with automatic abnormal-result flagging; billing, which

includes local health-insurance tariff and co-payment handling; administrative reporting; and the access-control framework outlined in Section 3.2. Each module exchanges data with the others exclusively through the application tier's validated API rather than through direct database access, preserving a single point of business-rule enforcement.

IV. DATABASE DESIGN

Seventeen interconnected tables make up the relational schema, which has been normalized to the third normal form (3NF) in accordance with the traditional relational design criteria developed by Codd (1970) and expanded by Fagin (1977): the first normal form eliminates repeating groups and enforces atomic field values; the second normal form eliminates partial dependency on composite keys; and the third normal form eliminates transitive dependency among non-key attributes. At the expense of the extra join operations needed for a fully normalized schema, applying 3NF throughout reduces data redundancy and maintains referential integrity across the patient, clinical, pharmacy, laboratory, and billing entities. This trade-off is lessened at the application layer thru targeted indexing on high-frequency lookup fields like hospital number, phone number, and date of birth.

V. COMPARATIVE EVALUATION AND DISCUSSION

Twelve functional criteria covering clinical, pharmacy, laboratory, billing, and security dimensions were assessed against three widely used open-source platforms: OpenMRS, Bahmni, and HospitalRun. Each platform's published documentation served as the basis for rating in order to place the design in relation to current practice (OpenMRS Community, 2021). Ratings fall into one of three categories: Not Supported (NS), Partially Supported (PS), or Fully Supported (FS). A representative subset of the comparison is shown in Table 3.

Table 3: Comparison of each platform

Criterion	Proposed Design	OpenMRS	Bahmni	HospitalRun
Patient registration & search	FS	FS	FS	FS
Appointment scheduling with SMS	FS	PS	PS	PS
SOAP clinical documentation	FS	FS	FS	FS
Electronic prescription workflow	FS	PS	FS	PS
Pharmacy inventory management	FS	NS	FS	PS
Laboratory request & results	FS	FS	FS	PS
NHIS/local-insurance billing	FS	NS	NS	NS
RBAC (six-role hierarchy)	FS	FS	FS	PS
NDPA/NDPR-aligned security controls	FS	NS	NS	NS

The comparison shows that the native local health-insurance billing workflow and its explicit alignment with Nigeria's current data-protection obligations under the NDPA 2023 are the main differentiators of the proposed design compared to the three comparator platforms; neither of these are natively addressed by platforms created primarily for regulatory contexts outside of Nigeria. Because of the depth of their multi-year, multi-country implementer communities, OpenMRS and Bahmni continue to be the more functionally mature platforms on core clinical documentation (Seebregts et al., 2009). An implementation of the current design would benefit from reusing some OpenMRS data-model conventions rather than replicating them from scratch. Interpretation should be guided by the two constraints of this judgment. First, response-time, uptime, and concurrency metrics mentioned elsewhere for this architecture are architectural forecasts rather than observed outcomes; it is a design-stage, documentation-based comparison rather than an empirical benchmarking exercise on operating systems. Second, the comparator systems' completeness scores are based on their publicly available default settings; institution-specific OpenMRS or Bahmni customization might close the gaps found here. The immediate next step for this study is determined to be empirical validation thru a pilot deployment.

VI. CONCLUSION

A three-tier, role-based Hospital Management System architecture for secondary hospitals in Nigeria has been described in this work. It consists of eight integrated clinical and administrative modules,

a six-role RBAC model, and a normalized seventeen-table schema. The design's main contributions are its native local-insurance billing integration and compliance with the current Nigeria Data Protection Act, 2023, according to a systematic assessment against three well-known open-source platforms. Future research will put this specification into practice and evaluate its usability and empirical performance at a real secondary hospital.

REFERENCES

- [1]. Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377–387. <https://doi.org/10.1145/362384.362685>
- [2]. Fagin, R. (1977). Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2(3), 262–278. <https://doi.org/10.1145/320557.320571>
- [3]. Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, 4(3), 224–274. <https://doi.org/10.1145/501978.501980>
- [4]. Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley.
- [5]. Jones, M., Bradley, J., & Sakimura, N. (2015). JSON Web Token (JWT) (RFC 7519). Internet Engineering Task Force. <https://doi.org/10.17487/RFC7519>
- [6]. Kruse, C. S., Smith, B., Vanderlinden, H., & Nealand, A. (2017). *Security techniques for the*

- electronic health records. *Journal of Medical Systems*, 41(8), Article 127.
<https://doi.org/10.1007/s10916-017-0778-4>
- [7]. National Information Technology Development Agency (NITDA). (2019). Nigeria Data Protection Regulation 2019. Federal Republic of Nigeria. <https://nitda.gov.ng/ndpr>
- [8]. Nigeria Data Protection Commission (NDPC). (2023). Nigeria Data Protection Act, 2023. Federal Republic of Nigeria. <https://ndpc.gov.ng>
- [9]. OpenMRS Community. (2021). OpenMRS developer guide: Architecture and module development. OpenMRS Inc. <https://wiki.openmrs.org/display/docs/Developer+Guide>
- [10]. Adenuga, K. I., Iahad, N. A., & Miskon, S. (2017). Towards reinforcing telemedicine adoption amongst clinicians in Nigeria. *International Journal of Medical Informatics*, 104, 84–96.
<https://doi.org/10.1016/j.ijmedinf.2017.05.008>
- [11]. Seebregts, C. J., Mamlin, B. W., Biondich, P. G., Fraser, H. S. F., Wolfe, B. A., Jazayeri, D., Allen, C., Miranda, J., Baker, E., Musinguzi, N., Kayiwa, D., Fourie, C., Lesh, N., Kanter, A., Yiannoutsos, C. T., Bailey, C., & the OpenMRS Implementers Network. (2009). The OpenMRS Implementers Network. *International Journal of Medical Informatics*, 78(11), 711–720.
<https://doi.org/10.1016/j.ijmedinf.2008.09.005>