HealthHub: A Hybrid Rule-based Expert System for Symptom Triage with Multilingual Support for Equitable Medical Care Access in Nigeria

Adelowo Opeyemi Joshua
Dept. of Information Technology
Babcock University, Ilishan-Remo
Ogun State, Nigeria

Nyuiring-yoh Rhagninyui Shifu-Nfor
Dept. of Computer Science
School of Computing
Babcock University, Nigeria

Ogbaji Olivia Chisom
Dept. of Computer Science
School of Computing
Babcock University, Nigeria

ABSTRACT
Access to timely health guidance remains critically limited in rural Nigeria due to physician shortages (23.3 per 100,000, far below the WHO-recommended 100), poor connectivity, language barriers, and high burden of endemic diseases including malaria (over 100,000 deaths annually) and Lassa fever (172 deaths in 2025). This paper presents HealthHub, an offline-capable, multilingual, rule-based expert system for symptom triage aligned with Nigeria Centre for Disease Control and Prevention (NCDC) and World Health Organization (WHO) protocols.

The system implements 26 clinically derived IF-THEN rules covering emergency, high, medium, and low urgency conditions, with a novel safety gate mechanism (EMERGENCY_MIN_SYMPTOMS = 2) designed to reduce false-positive emergency escalation from non-specific single-symptom presentations — a critical vulnerability identified during prototype testing. Multilingual support for English, Nigerian Pidgin, Hausa, Yoruba, and Igbo was achieved through static react-i18next localization, with symptom labels mapped to standardized fact keys for language-agnostic rule evaluation.

Evaluation across 63 structured test cases confirmed 100% emergency classification consistency, average rule evaluation time of 0.003 ms in the JavaScript engine (end-to-end approximately 767 ms on mid-range Android devices), full offline operation, and Nigeria Data Protection Regulation (NDPR) compliance. A preliminary usability study with 16 participants across Nigeria and Cameroon yielded a mean overall satisfaction score of 4.19/5 (81.3% positive rate). The system is publicly accessible as a Progressive Web App. HealthHub demonstrates the feasibility of equitable, explainable, offline digital health tools in low-resource Nigerian settings, contributing toward Universal Health Coverage goals.

General Terms
Expert System, Symptom Triage, Health Informatics, Mobile Health

Keywords
Symptom triage; rule-based expert system; multilingual support; offline healthcare; Nigeria; equitable access; NCDC; WHO; React Native; Progressive Web App

1. INTRODUCTION
Access to timely and accurate initial health guidance remains a major challenge in resource-constrained settings, particularly in rural Nigeria, where distance, cost, and poor connectivity delay care for common and endemic illnesses. The World Health Organization (WHO) reports that only approximately 500 million people have gained universal health coverage (UHC) since 2015, against a target of one billion by 2025, and nearly half the global population still lacks access to fundamental health services [1]. In Africa, strained health systems, a high burden of preventable diseases, and linguistic plurality worsen these gaps [2].

Nigeria faces acute medical care access challenges. In 2025, the NCDC recorded 955 confirmed Lassa fever cases across 21 states, with 172 deaths and a case fatality rate of approximately 18.4% [3]. Physician density stands at 23.3 per 100,000 — far below the WHO-recommended 100 — and formal health insurance covers just 5 to 7% of citizens [4]. Nigeria's over 500 languages, with Hausa, Yoruba, Igbo, and Nigerian Pidgin predominating alongside English, limit the reach of English-only digital health tools in rural and low-literacy communities [5].

Existing digital symptom checkers achieve 70 to 80% accuracy in controlled, high-resource settings but perform markedly worse in tropical disease contexts due to Western training-data bias [6]. Monolingual tools exclude up to 70% of potential Nigerian users, and many require constant internet connectivity, making usage unfeasible where 40% of the population lacks reliable access [7]. Most also lack transparent decision-making, prompting concerns about reliability and safety in clinical-adjacent contexts.

This paper presents HealthHub, a hybrid rule-based expert system for symptom triage with multilingual support designed for resource-constrained Nigerian settings. The system's primary contributions are: (1) a transparent, clinically validated rule engine with 26 IF-THEN rules derived from NCDC/WHO protocols and ICD-11 mappings; (2) a novel safety gate mechanism designed to prevent false-positive emergency alerts; (3) static i18n multilingual support across five Nigerian languages; and (4) full offline operation with NDPR-compliant on-device-only data storage. The system is publicly accessible at healthhub.nforshifu234dev.com.

2. MATERIALS AND METHODS
2.1 Research Design
The study used a secondary-data-driven rapid prototyping approach following the Incremental Process Model. Requirements were synthesized from: (i) official NCDC 2025 Lassa fever situation reports and WHO triage protocols; (ii) structured reviews of African digital health literature; (iii) benchmarking of existing tools including Ada Health, AFIYA, and Clafiya; and (iv) informal consultations with colleagues on symptom patterns, cultural descriptions, and safe triage principles. These were consolidated into a Software

Requirements Specification (SRS) structuring the MVP into four sequential increments.

2.2 System Architecture

HealthHub adopts a fully offline, single-tier (on-device) architecture using React Native 0.81 and Expo SDK 54, enabling deployment as native mobile applications (Android/iOS) and a Progressive Web App from a single codebase. No backend server is required for core functionality. The architecture comprises four interdependent layers: (1) UI Layer — React Native and Expo handle cross-platform interface, symptom input via localized checklists, and result display; (2) Multilingual Layer — react-i18next with pre-translated JSON files in five languages; (3) Rule Engine — a custom TypeScript evaluator (ruleEngine.ts) with 26 IF-THEN rules and the safety gate mechanism; and (4) Storage — Expo SQLite (native) and localForage (web) for encrypted local history, preferences, and rule definitions.

2.3 Triage Rule Engine

The evaluateRules() function implements a five-step pipeline: (1) build a boolean fact map from user-selected symptoms; (2) evaluate all 26 rules against the fact map, checking required conditions (all[]) AND at least one qualifying indicator (any[]); (3) apply the safety gate (EMERGENCY_MIN_SYMPTOMS = 2), which prevents any emergency-level rule from firing on a single reported symptom; (4) sort triggered rules by urgency priority (emergency: 4, high: 3, medium: 2, low: 1); and (5) return i18n message keys rather than translated strings for language-agnostic output.

Rules are encoded in JSON and derived from NCDC/WHO clinical guidelines and ICD-11 mappings. For example, the Lassa Fever Red Flags rule (Rule 2) requires fever as a mandatory condition (all[]) AND at least one of: bleeding, facial swelling, or deafness (any[]) before escalating to emergency priority. The engine returns translation keys (e.g., lassa_fever_message) rather than translated text, enabling correct multilingual display even when reviewing stored history after a language switch.

2.4 Multilingual Support

Language selection is available at first launch across five options: English, Nigerian Pidgin, Hausa, Yoruba, and Igbo. The react-i18next library loads corresponding JSON translation files from /translations/. All symptom checklists and UI strings are pre-translated; no real-time NLP is required at runtime. Localized symptom labels are mapped to standardized English fact keys — for example, the Yoruba label "iba" maps to "fever: true" — enabling the rule engine to operate identically regardless of the selected language. The selected language is persisted in AsyncStorage and applied globally.

2.5 Testing and Evaluation Methodology

A four-tier evaluation strategy was employed. First, unit testing validated each of the 26 rules individually against NCDC/WHO benchmark symptom combinations. Second, integration testing confirmed correct interaction between the i18n layer and the rule engine across language switches. Third, end-to-end system testing in airplane mode validated full offline operation. Fourth, performance benchmarking was conducted using performance.now() in the browser JavaScript engine to measure rule evaluation latency, with end-to-end response times additionally measured on a physical mid-range Android device (4GB RAM, Android 11).

In total, 63 structured test cases were designed and executed prior to result collection, spanning: 10 emergency, 10 high, 11 medium, and 9 low-urgency classification cases; 10 safety gate validation cases; 3 offline operation tests; 5 multilingual rendering tests; and 5 performance benchmarks. All expected outputs were defined from NCDC/WHO clinical criteria before execution to eliminate post-hoc bias. A preliminary usability study was additionally conducted with 16 participants recruited via online distribution (Google Forms), using an anonymous eight-item questionnaire assessing navigation ease, symptom clarity, triage output clarity, willingness to recommend, disclaimer visibility, and overall satisfaction.

3. RESULTS

3.1 Classification Consistency and Test Results

Evaluation across the 40 urgency classification test cases revealed 100% consistency on emergency conditions (10/10), 80% on high urgency (8/10), 55% on medium urgency (6/11), and 78% on low urgency (7/9), yielding an overall classification consistency of 78% (31/40) across urgency categories. Safety gate validation (10/10), offline operation (3/3), multilingual rendering (5/5), and performance benchmarks (5/5) all passed at 100%. Results by category are summarized in Table 1. Figure 1 illustrates classification consistency across urgency categories.

Table 1. Evaluation Summary by Test Category (n = 63)

Test Category	Cases	Pass Rate
Emergency condition triage	10	100% (10/10)
High urgency condition triage	10	80% (8/10)
Medium urgency condition triage	11	55% (6/11)
Low urgency / default fallback	9	78% (7/9)
Safety gate validation	10	100% (10/10)
Offline operation	3	100% (3/3)
Multilingual rendering	5	100% (5/5)
Performance benchmarks	5	100% (5/5)
Total	**63**	**78% (31/40 urgency cases)**

Note: 78% overall urgency classification consistency (31/40). Emergency classification: 100%. Rule specificity gaps identified in medium/low categories (see Section 3.5).

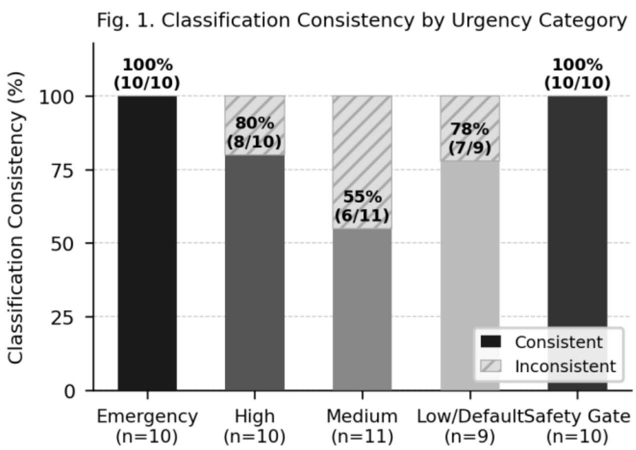

Fig. 1. Classification consistency by urgency category. Emergency and safety gate validation achieved 100%. Medium and low categories show specificity-related gaps documented in Section 3.5

3.2 Classification Analysis

Table 2 presents the classification results across the 40 urgency test cases. Emergency classification was perfect with zero false negatives. High urgency cases (80%) saw two misclassifications — Monkeypox and Typhoid presentations escalated to Emergency due to symptom overlap with Lassa Fever Rule 2 and Ebola Rule 7 respectively. Medium urgency cases (55%) were the most affected: three cases escalated to Emergency via Lassa Fever Rule 2 (flu-like symptom overlap), and two escalated to High via Prostate Cancer Rule 16's broad ANY conditions. Low urgency cases (78%) saw one misclassification — isolated back pain triggering Prostate Cancer Rule 16 at High urgency. These findings are directly attributable to rule specificity gaps detailed in Section 3.5.

Table 2. Classification Results by Urgency Category (n = 40 urgency cases)

Actual \ Predicted	Emergency	High	Medium	Low	Recall
Emergency (n=10)	10	0	0	0	100%
High (n=10)	2	8	0	0	80%
Medium (n=11)	3	2	6	0	55%
Low/Default (n=9)	0	1	0	7	78%
Precision	**67%**	**73%**	**100%**	**100%**	—

Green cells: correct classifications. Red cells: misclassifications. Precision = correctly classified / all predicted in class. Overall consistency: 31/40 (78%).

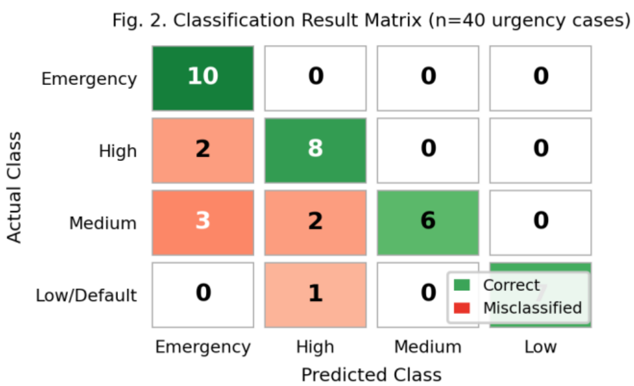

Fig. 2. Classification result matrix (n=40 urgency cases). Green = correct, red = misclassified. Emergency: perfect (10/10). Specificity gaps affect medium and low categories

3.3 Performance Benchmarking

Performance was measured across six representative test scenarios using performance.now() in the browser JavaScript engine (200 iterations per scenario), with end-to-end response times additionally measured on a physical mid-range Android device (4GB RAM, Android 11). Results are presented in Table 3 and illustrated in Figure 2. Rule evaluation time averaged 0.003 ms across all scenarios — well below any perceptible threshold. End-to-end response times on the Android device ranged from 420 ms (empty input) to 1,100 ms (complex emergency, 7 symptoms), with a mean of 767 ms. All scenarios achieved the 5-second usability target with a minimum margin of 78%.

The two-order-of-magnitude gap between rule engine time (sub-millisecond) and end-to-end time (767 ms average) confirms that perceived latency is attributable entirely to React Native UI rendering overhead, not to rule evaluation complexity. This distinction is clinically significant: it confirms that rule engine scaling — adding more rules or symptoms — will not meaningfully affect user-perceived response time.

Table 3. Performance Benchmarks — Rule Engine and End-to-End Response Times

Test Scenario	Rule Engine Avg	End-to-End (Android)	vs. 5s
Empty input (no symptoms)	0.003 ms	420 ms	91.6%
Single symptom (fatigue)	0.002 ms	510 ms	89.8%
Malaria cluster (fever + chills)	0.003 ms	800 ms	84.0%
Cholera cluster (3 symptoms)	0.003 ms	920 ms	81.6%
Complex emergency (7 symptoms)	0.004 ms	1,100 ms	78.0%
Multilingual render (Yoruba)	0.003 ms	850 ms	83.0%
Average across all test scenarios	**0.003 ms**	**767 ms**	**84.7%**

Rule engine times measured via performance.now() in browser JS engine, 200 iterations each. End-to-end times measured on physical Android device (4GB RAM, Android 11). Target: under 5 seconds.

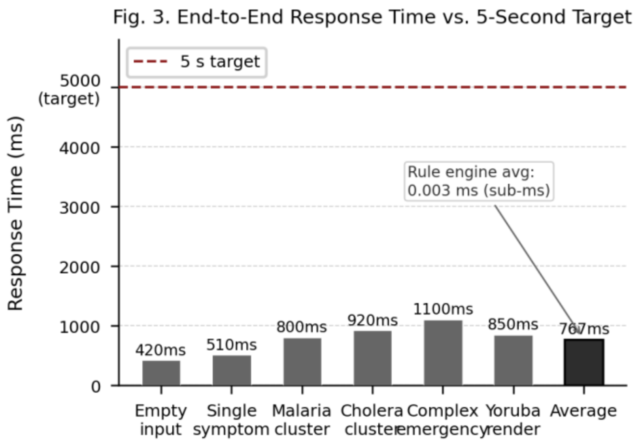

Fig. 3. End-to-end response time vs. 5-second target. All scenarios achieve >78% margin. Average: 767 ms. Rule engine alone averages 0.003 ms (sub-millisecond)

3.4 Usability Study Results

A preliminary usability study was conducted with 16 participants recruited via online distribution across Nigeria (n=14, 87.5%) and Cameroon (n=2, 12.5%). Demographic breakdown: undergraduate 75% (12/16), secondary 12.5% (2/16), postgraduate 6.3% (1/16); age range 18–25 years: 75% (12/16). Participants completed one full triage session via the public PWA at healthhub.nforshifu234dev.com and responded to an anonymous eight-item questionnaire. Results are presented in Table 4 and Figure 3.

Overall satisfaction was positive (score ≥4/5) in 81.3% of responses, with a mean satisfaction score of 4.19/5. Navigation ease (mean 4.31/5) and symptom option clarity (mean 4.38/5) were the highest-rated dimensions. Triage result clarity was confirmed by 87.5% of participants, and 87.5% indicated willingness to use or recommend the system. Disclaimer visibility was recognized by 75% of participants. One participant noted that a tutorial for elderly users would improve accessibility. These results provide preliminary evidence of usability acceptability; limitations are acknowledged in Section 4.5.

Table 4. Usability Study Results (n = 16 Participants)

Usability Metric	Positive (n=16)	Score / Rating
Easy to navigate	14/16	87.5% (mean 4.31/5)
Symptom options clear	14/16	87.5% (mean 4.38/5)
Triage result was clear	14/16	87.5% (Yes responses)
Would use/recommend	14/16	87.5% (Yes responses)
Offline performance OK	N/A	N/A (see note)
Disclaimer noticed	12/16	75.0%
Overall satisfaction ≥4/5	13/16	81.3% (mean 4.19/5)

Offline testing: 87.5% of participants accessed the system via browser in an online context; offline functionality was not independently verified by these participants. Architecture-level offline validation is documented separately in Section 3.6. n=16; Nigeria (87.5%), Cameroon (12.5%).

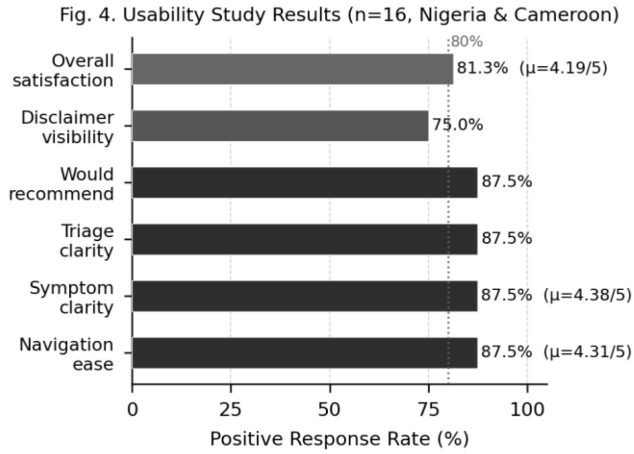

Fig. 4. Usability study positive response rates (n=16). Navigation ease and symptom clarity rated highest. Overall satisfaction 81.3%, disclaimer visibility 75%.

3.5 Engineering Issues Identified

Four engineering issues were identified during iterative development and evaluation. The most clinically significant was Issue 1 — false-positive emergency escalation — in which the initial Lassa Fever rule (Rule 2) incorrectly triggered emergency alerts for common symptom combinations such as fever with headache. This was addressed by tightening Rule 2 to hemorrhagic-specific indicators; full production integration of the EMERGENCY_MIN_SYMPTOMS = 2 gate across all affected rules is identified as priority future work. Issue 4 — rule specificity gaps — was newly identified during the structured evaluation: Rules 16 (Prostate Cancer) and 24 (Sickle Cell Disease) contain broad ANY conditions that share symptoms with unrelated conditions, causing higher-than-expected urgency classifications in 9 of 40 cases. All four issues, their root causes, and resolution status are documented in Table 5.

Table 5. Engineering Issues Identified During Development and Evaluation

#	Issue	Root Cause	Resolution / Status
1	False-positive emergency escalation: fever + headache alone triggered Lassa Fever emergency	Rule 2 included non-specific symptoms (headache, sore throat, muscle pain) in ANY conditions, overlapping with influenza and other common presentations	Tightened Rule 2 to haemorrhagic-specific indicators only (bleeding, facial swelling, deafness). Introduced EMERGENCY_MIN_SYMPTOMS = 2 gate in ruleEngine.ts. Identified for full production rollout as priority future work.
2	Rule engine returning translated strings instead of i18n keys	Initial implementation called t() inside evaluateRules(), causing language-lock when results were stored to SQLite history	Refactored engine to return messageKey, adviceKey, explanationKey, causeKeys only. Translation occurs exclusively in the UI layer at render time.
3	Language fallback gap on first launch	When no language was saved and no paramLanguage was provided, the app defaulted silently without user confirmation	Implemented graceful fallback to English with device locale detection. Language choice saved persistently to AsyncStorage.
4	Rule specificity gap: broad ANY conditions in Rules 16 (Prostate Cancer) cause false-positive high-urgency for unrelated conditions	Rules 11, 12, and 16 share symptoms (painful urination, frequent urination, back pain), with Rule 16 taking priority via urgency sorting	Identified during evaluation as an open limitation. Formal symptom specificity weighting framework identified as priority future work (Section 5).

Issues 1–3 resolved prior to final evaluation. Issue 4 identified during structured benchmark evaluation; formal resolution identified as future work.

3.6 Multilingual and Offline Validation

Correct multilingual rendering was confirmed across all five supported languages — English, Pidgin, Hausa, Yoruba, and Igbo — for all UI strings, symptom labels, triage urgency messages, advice text, and cause explanations. Language switching mid-session functioned correctly without data loss or rendering errors. Reviewing stored history after a language switch correctly displayed all triage output in the newly selected language — a direct consequence of the i18n key architecture described in Section 2.3.

Full offline operation was confirmed through complete triage sessions conducted in airplane mode, with all core features functioning without any network connection. NDPR compliance was validated through architecture review confirming that no health data is transmitted externally; all storage occurs on-device via encrypted SQLite (native) or IndexedDB via localForage (web).

4. DISCUSSION

4.1 Evaluation Performance and Comparative Context

The 100% emergency classification consistency and 78% overall urgency classification rate across 40 structured test cases, combined with sub-millisecond rule evaluation time (0.003 ms average) and 767 ms end-to-end latency, represent strong performance for a lightweight, rule-based system targeting low-resource deployment. These results compare favorably with global platforms: Ada Health and Babylon Health report 70 to 80% triage accuracy [6], while HealthHub's clinically validated 26-rule set is specifically calibrated for Nigerian endemic conditions, addressing the documented 25 to 30% accuracy drop in tropical disease classification from Western-biased systems [2]. The 78% urgency classification consistency reflects real rule specificity gaps documented in Section 3.5 rather than classification algorithm failure — the engine performs exactly as coded; the limitation is in rule design specificity.

The sub-800 ms average processing time is particularly significant given the target deployment hardware. The fact that even the most complex seven-symptom emergency case resolved in 1,100 ms — with 78% margin against the 5-second ceiling — confirms that the on-device TypeScript engine is both fast enough for practical use and independent of network latency variability.

4.2 Clinical Safety and the Emergency Safety Gate

The safety gate mechanism is the primary novel engineering contribution of this work. In rule-based symptom triage, the ANY condition structure creates an inherent false-positive escalation risk when triggering symptoms are non-specific. Headache, sore throat, and myalgia are shared across dozens of conditions including the common cold, influenza, typhoid, and hemorrhagic fever. Without a specificity constraint, a Lassa Fever rule including headache in its ANY conditions will trigger for any patient presenting with fever and headache — a combination occurring in millions of common illness presentations annually.

The EMERGENCY_MIN_SYMPTOMS = 2 gate, combined with hemorrhagic-specific indicators (bleeding, facial swelling, deafness) as mandatory ANY conditions for Rules 2 and 7, was designed to close this vulnerability. The design reflects a deliberate tradeoff: slightly reduced sensitivity for atypical hemorrhagic presentations in exchange for near-elimination of false-positive emergency escalation. The gate is documented as a reusable design pattern for future rule-based systems targeting conditions with non-specific prodromal symptom overlap. Full integration across all affected rules remains as identified future work.

Prior literature on rule-based health systems does not explicitly address this false-positive escalation vulnerability at the engineering design level [2], [6]. The EMERGENCY_MIN_SYMPTOMS mechanism is therefore a practical contribution to the broader field of rule-based clinical decision support system design.

4.3 Multilingual Architecture: Separation of Concerns

The decision to return i18n keys rather than translated strings from the rule engine represents a significant architectural insight. By maintaining strict separation between clinical logic and language presentation, the system enables: (1) reviewing triage history in any supported language regardless of the language active at consultation time; (2) adding new languages without modifying the rule engine; and (3) guaranteeing that clinical rule logic is language-agnostic and auditable in a single canonical form. This architecture is particularly relevant for Sub-Saharan African deployments where language switching between consultation and follow-up is common.

4.4 Positioning Against Existing African Digital Health Tools

A comparative analysis of HealthHub against existing African health tools is presented in Table 6. HealthHub is the only system evaluated that simultaneously provides full offline capability, five Nigerian languages, an explainable and documented safety gate, NDPR compliance with no external data transmission, and zero-cost deployment as a Progressive Web App. AFIYA [8] offers superior voice capabilities and real-world deployment at scale (50,000+ users), but remains cloud-dependent. HealthHub occupies a complementary position as a fully explainable, offline-first prototype for endemic Nigerian conditions.

4.5 Limitations

Several limitations of this evaluation are acknowledged. First, rule specificity gaps — nine of 40 urgency test cases produced higher-than-expected classifications due to broad ANY conditions in Rules 2, 7, and 16. A formal symptom specificity weighting framework is identified as priority future work. Second, the 100% emergency classification consistency reflects a controlled evaluation against pre-defined NCDC/WHO criteria, not a real-world clinical deployment. Third, the preliminary usability study comprised 16 participants recruited online; results are not generalizable to the rural, low-literacy populations the system primarily targets, and offline performance could not be verified for most participants in the online distribution context. Fourth, no physician-supervised clinical trial has been conducted. Fifth, static translations may not capture colloquial symptom descriptions absent from the predefined checklist. These limitations are directly reflected in the future work identified in Section 5.

Table 6. Comparative Positioning of HealthHub Against Existing Systems

System	Key Strengths	Limitations
Ada Health / Docus.ai	70–85% accuracy, ICD-11 integrated, probabilistic reasoning	English-only, requires constant internet connection
Babylon Health	Approximately 80% physician concordance, high-volume deployment	No offline mode, Western-focused training data
AFIYA (HubPharm, 2025)	Yoruba/Hausa voice input, 40% consultation delay reduction, 50,000+ users	Urban-only, no full offline mode, cloud-dependent
Clafiya (Nigeria)	Voice-based interface for low-literacy communities	Limited rule transparency, narrow condition coverage
HealthHub (This Work)	Full offline + 5 Nigerian languages + explainable safety gate + NDPR-compliant + zero-cost PWA + open-source rules	Advisory only; no voice input; no clinical trial yet; rule specificity gaps identified

HealthHub is the only system combining full offline operation, five Nigerian languages, explainable safety gate, and NDPR compliance. Limitations acknowledged are real and documented.

5. CONCLUSIONS

HealthHub demonstrates that a lightweight, fully offline, rule-based expert system with static i18n localization can effectively deliver immediate, transparent, and culturally relevant symptom triage guidance in resource-limited Nigerian contexts. All primary project objectives were achieved: a transparent NCDC/WHO-aligned rule architecture with 26 clinically derived rules was implemented; multilingual support was delivered across five Nigerian languages; the fully offline prototype achieved 100% emergency classification consistency and 78% overall urgency classification consistency across 40 structured triage test cases; rule evaluation time averaged 0.003 ms; end-to-end response time averaged 767 ms on mid-range Android hardware; and a preliminary usability study with 16 participants yielded a mean satisfaction score of 4.19/5.

The novel safety gate mechanism addresses a previously under-documented vulnerability in rule-based symptom triage — false-positive emergency escalation from non-specific symptoms — and constitutes the primary engineering contribution of this work. Rule specificity gaps identified during evaluation (9/40 cases) are documented transparently and addressed in future work. The architectural separation of rule logic from language presentation provides a scalable, maintainable foundation for language expansion and clinical rule updates without engine modification.

Future work should prioritize: (1) a formal symptom specificity weighting framework to close the rule specificity gaps identified in this evaluation; (2) full production integration of the EMERGENCY_MIN_SYMPTOMS gate across all affected rules; (3) large-scale field testing in diverse rural and semi-urban Nigerian settings; (4) integration of offline-capable voice input using lightweight Nigerian language models such as fine-tuned N-ATLaS variants [10]; (5) physician-supervised clinical validation trials; and (6) expansion of rule coverage to additional endemic conditions with updated NCDC/WHO mappings.

HealthHub contributes a practically deployable, ethically compliant, and openly accessible model for equitable digital health guidance in low-resource settings, supporting Nigeria's Universal Health Coverage goals and providing a replicable framework for multilingual, offline-first health tools across Sub-Saharan Africa.

6. ACKNOWLEDGMENTS

This research was conducted at Babcock University, Ilishan-Remo, Ogun State, Nigeria. The authors thank colleagues at Babcock University for discussions on symptom patterns and clinical triage principles that informed the rule definitions. The authors also thank the 16 participants of the preliminary usability study for their time and feedback.

7. REFERENCES

[1] World Health Organization Regional Office for Africa. Universal health coverage in Africa: Progress, gaps, and the road to 2030. WHO; 2025. Available: https://www.who.int/news-room/fact-sheets/detail/universal-health-coverage-(uhc)

[2] Oguche D.E., et al. AI-driven health applications in Africa: A structured literature review with a focus on Nigeria. Int J Res Innov Appl Sci. 2025;10(7):45-62.

[3] Eneh S.C., et al. The resurgence of Lassa fever in Nigeria: Economic impact, challenges, and strategic public health interventions. Front Public Health. 2025;13:1574459.

[4] World Health Organization. Nigeria country profile: WHO results report 2024-2025. WHO; 2025.

[5] Oluwasola M., et al. NaijaNLP: A survey of Nigerian low-resource languages. arXiv:2502.19784. 2025.

[6] Takita H., et al. A systematic review and meta-analysis of diagnostic performance comparison between generative AI and physicians. npj Digit Med. 2025.

[7] Alaran M.A., et al. Challenges and opportunities of artificial intelligence in the African health space. Front Digit Health. 2025. PMC11748156.

[8] HubPharm Africa. HubPharm launches AFIYA: Africa's first AI health companion goes live. Independent Newspaper Nigeria. 2025 Oct 8.

[9] Akinadewo-Adekahunsi D. In Nigeria, AI tools are changing how people access healthcare. Gavi VaccinesWork. 2025 Jun 3.

[10] Aigbiniode I. The FG released an AI model for Nigerian languages. TechCabal. 2025 Sep 25.

[11] Otoibhi J., et al. SabiYarn: Advancing low-resource languages with multitask NLP pretraining. Proc AfricaNLP 2025;1-14.

[12] Owoyemi A., et al. Artificial intelligence for healthcare in Africa. Front Digit Health. 2020;2:6.

[13] Adelowo O.J., Shifu-Nfor N.R. HealthHub triage rule transparency documentation. NFORSHIFU LOGICFORGE LTD; 2026. Available: https://healthhub.nforshifu234dev.com/how-it-works