

*Research Article***An Offline-First Mobile Reporting System for Digital One Health Surveillance in Resource-Constrained Settings****Edoghogho Olaye ^{a,*} , Daniel Obuh ^a**^a *Department of Computer Engineering, Faculty of Engineering, University of Benin, Edo State, Nigeria*

ARTICLE INFO

Article history:

Received 23 May 2026

Accepted 30 June 2026

*Keywords:*One Health surveillance,
mobile health (mHealth),
Design Science Research
Methodology,
offline-first systems,
conversational AI,
public health informatics

ABSTRACT

The increasing interdependence of human, animal, and environmental health highlights the need for integrated One Health surveillance systems capable of supporting timely and coordinated disease reporting. In many low- and middle-income settings, existing surveillance workflows remain fragmented, paper-based, and poorly suited for field conditions. This study employs the Design Science Research Methodology (DSRM) to design, develop, and evaluate a mobile One Health surveillance application that addresses these limitations. The proposed solution is a cross-platform, offline-capable mobile system supporting structured digital reporting, local data storage, and an AI-assisted conversational reporting interface aimed at reducing reporting burden. The artifact was demonstrated through simulated real-world usage scenarios and evaluated using functional testing and usability assessment. Results show a System Usability Scale (SUS) score of 78.4, indicating good overall usability, while the conversational interface reduced average reporting time despite a higher learning curve. The findings demonstrate the feasibility of offline-first mobile surveillance systems for improving One Health data capture in resource-constrained environments.

This is an open access article under the CC BY-SA 4.0 license.
(<https://creativecommons.org/licenses/by-sa/4.0/>)

1. INTRODUCTION

The adoption of mobile devices or smartphones has skyrocketed in recent years. Nigeria's smartphone user base has grown to roughly 50 million, reflecting one of the world's most dynamic mobile markets. Despite this rapid smartphone growth, basic phones remain the most widely used type of handset in the nation, mirroring a broader continental pattern [1]. Concurrently, the healthcare sector has increasingly leveraged mobile applications for critical tasks such as remote patient monitoring, telemedicine, and public health surveillance. These technologies have been shown to improve healthcare access, patient engagement, and the real-time tracking of diseases [2]. This established success in human-centric health models presents a compelling precedent for adapting mobile app solutions to the interconnected challenges of One Health, where monitoring the health of humans, animals, and environments is essential.

The One Health concept is founded on the understanding that the health of humans, animals, and their environments are profoundly interconnected [3]. Existing mobile health tools have limitations, including complex reporting forms that reduce usability and hinder efficient data entry.

The purpose of this study is to solve the problem of reporting One Health cases to DOHS using the Design Science Research Methodology (DSRM).

2. RELATED WORKS

The increasing adoption of mobile health (mHealth) technologies has significantly influenced how healthcare systems collect, transmit, and analyze surveillance data. Numerous studies have demonstrated the transformative impact of mobile-based solutions in improving data quality, timeliness, and accessibility across healthcare contexts. [4] conducted a foundational systematic review

* Corresponding author. E-mail address: edoghogho.olaye@uniben.edu
DOI: 10.58190/ijamec.2026.172

which showed that mobile applications enhance healthcare service delivery by enabling rapid disease reporting, patient monitoring, and improved communication between health workers and public health systems. Similarly, [5] highlighted the effectiveness of SMS and mobile platforms in infectious disease surveillance across low- and middle-income countries, demonstrating their suitability for resource-constrained environments. These findings align with [6], who emphasized that mobile health tools improve reach and engagement but face limitations such as low usability—issues that remain prevalent in current surveillance tools.

In the African context, mobile-based solutions have also shown considerable promise in supporting public health surveillance. [7] highlighted that during efforts to combat Ebola outbreaks, mobile surveillance systems played a critical role in enhancing early outbreak detection and improving response coordination.

The One Health framework provides a broader interdisciplinary context for these technological advancements. The concept, as articulated by FAO, WHO, and OIE [8], positions human, animal, and environmental health as interdependent sectors requiring integrated surveillance approaches. However, several scholars note that current surveillance mechanisms still operate in sectoral silos. [9] argue that disjointed reporting structures create blind spots, weakening cross-species disease detection and limiting timely response. [10] emphasize that despite increasing recognition of the One Health paradigm, the operationalization of integrated surveillance remains constrained by technological, institutional, and data-sharing challenges.

To address these concerns, several mobile-based surveillance innovations have emerged. A notable example is the mobile app and dashboard system developed by [11], which demonstrated that integrating mobile data capture with real-time analytics significantly enhances the accuracy, completeness, and timeliness of outbreak reporting. The system allowed health workers to submit outbreak information from any location using either a smartphone or desktop interface, while the dashboard provided daily, weekly, and historical summaries to support informed public health decision-making. This study illustrates how mobile platforms can strengthen early detection and streamline response workflows. However, despite these advancements, existing solutions still reveal persistent gaps in usability, form complexity, and overall user experience, which continue to hinder widespread adoption, particularly in resource-constrained environments.

Usability remains a persistent challenge across mHealth solutions. [12], in their review of mobile health application usability, noted that many systems fail due to overly complex interfaces and lengthy forms that discourage user engagement. [13] similarly stressed that usability

problems directly affect adoption rates, especially in field environments where health workers require fast, intuitive systems. These limitations directly align with issues identified in existing One Health surveillance tools, where tedious reporting workflows and multi-step forms reduce efficiency and compliance.

Finally, from a methodological standpoint, Design Science Research provides a structured pathway for developing solutions to these challenges. [14] positioned Design Science Research as a rigorous approach for creating innovative artifacts that solve real-world problems. [15] later formalized the Design Science Research Methodology (DSRM) process into six cyclical steps—problem identification, objective definition, design and development, demonstration, evaluation, and communication—making it well-suited for designing complex technology-based interventions such as surveillance applications.

Further, advancements in the development of Flutter-based cross-platform solutions have reinforced its viability for mobile health environments where resources are limited, and research has shown that it is a viable option for cross-platform development, achieving competitive rendering performance – and maintenance overhead – compared to approaches using native dual-codebases [21, 22]. Previously, offline-first and local-first software architectures have been developed and tested, showing that the embedding of synchronization logic on the client side, instead of depending on persistent connectivity, has a significant impact on the completeness of data in intermittent-network situations [23]. Issues related to the security of the mHealth data have also been reaffirmed in recent reviews, suggesting the need for end-to-end encryption, role-based access control, and lightweight authentication mechanisms suitable for the field environment [24, 25].

Collectively, the reviewed literature establishes that while mobile technologies significantly enhance health surveillance, existing systems remain fragmented, sector-specific, and often difficult for field workers to use. No current solution provides a unified, user-friendly mobile platform for integrated human–animal–environment reporting within the Nigerian One Health ecosystem. This gap underscores the need for the mobile surveillance system developed in this study using the Design Science Research Methodology.

3. METHODOLOGY

3.1. Research design

The research method employed is the Design Science Research Methodology (DSRM). DSRM produces knowledge through the creation and evaluation of innovative and useful artifacts aimed at solving real-world challenges [16]. The DSRM framework comprises six

iterative phases: problem identification, definition of solution objectives, design and development, demonstration, evaluation, and communication [17].

3.2. Problem identification

The problem identification phase involved qualitative data collection through semi-structured interviews and direct observations with health surveillance staff at the University of Benin Teaching Hospital. These sessions aimed to identify current challenges in field-based health surveillance reporting and gather functional requirements for the proposed mobile application. The identified user requirements are presented in Table 1.

Table 1. User Requirements

No	User requirement
1	The app must work on both Android and iOS phones without needing two separate development projects.
2	The app must be usable in field conditions with no internet connection (offline capability).
3	Reporting complex health incidents is slow and cumbersome with too many form fields to fill out.
4	The app interface should be simple, intuitive, and easy to navigate for users with varying levels of tech expertise.
5	The app needs to be fast and responsive, not sluggish, when loading data or switching screens.
6	Users require a more efficient method for submitting reports without manually completing every field.

3.3. Define objectives of a solution

Based on the identified requirements, specific solution objectives were defined to guide the development process. Table 2 maps each user requirement to its corresponding technical solution.

Table 2. Requirements-to-Solution Mapping

No	User Requirement	Proposed Solution
1	Cross-platform compatibility	Develop using a Single Codebase with Flutter
2	Offline capability	Implement Local Database (SQLite) to make the app work offline.
3	Tedious, multi-step reporting forms.	Smooth, structured workflow and data handling between each step of the reporting process. Also, dropdowns for selecting some form fields
4	Need for a simple, intuitive interface.	The use of Flutter’s rich set of customizable widgets to create clean, responsive, and user-friendly interface.
5	Requirement for a fast, responsive app.	Efficient API calls with modular widget development for fast user experience.
6	Desire for a smarter reporting method.	Conversational UI within the Flutter app that connects to the backend AI agent.

3.4. System design

3.4.1. Use case analysis

Use case diagrams were developed to model system functionality and clarify ambiguities in user interactions [18]. Figure 1 illustrates the primary use cases for field workers (actors), including authentication, location updates, manual report submission, outbreak monitoring, and AI-assisted reporting.

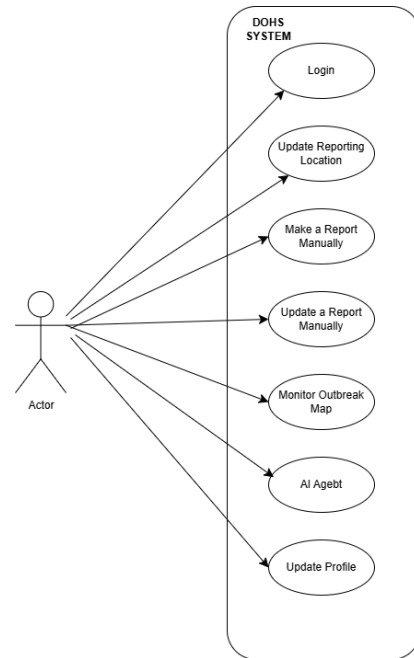


Figure 1. Use Case Diagram

3.4.2. Activity modelling

Activity diagrams were created to represent the sequential and concurrent workflows within the application [19]. Two critical workflows were modeled and presented:

Authentication Workflow: The login process implements secure authentication where users receive credentials via email. Upon entering valid credentials, the system verifies the account against the backend database. Successful authentication redirects users to the dashboard, while failed attempts trigger appropriate error notifications.

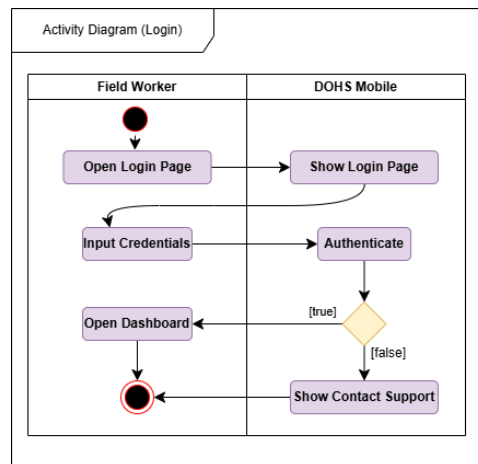


Figure 2. Login Activity Diagram

Manual Report Submission Workflow: This workflow depicts the multi-step process for creating and submitting field reports, including data validation and backend API communication.

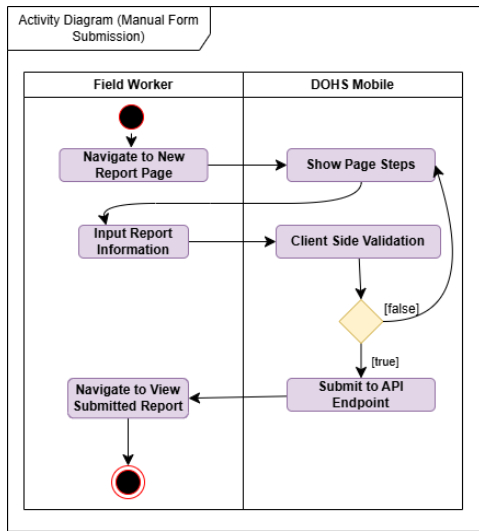


Figure 3. Manual Form Submission Workflow

3.5. Implementation

3.5.1. Development environment

The mobile application was developed using Flutter, leveraging the Dart programming language. Flutter's cross-platform capabilities enabled simultaneous deployment to both Android and iOS platforms from a single codebase. The development process utilized Flutter's package ecosystem (Pub) for dependency management and incorporated Flutter's stateful hot reload feature to accelerate iterative development [20].

3.5.2. Architectural design

The application implements a Provider-centric architecture pattern, establishing unidirectional data flow and clear separation between business logic and user interface components. The architecture consists of three primary layers:

Presentation Layer: Flutter widgets that render the user interface.

Business Logic Layer: Provider classes (e.g., DOHSAuthProvider) utilizing ChangeNotifier for state management.

Data Layer: RESTful API client for backend communication and SQLite for local data persistence.

State changes in Provider objects trigger notifications to registered listeners, resulting in reactive UI updates without manual intervention.

3.5.3. Data exchange protocol

Communication between the mobile client and backend server follows RESTful API principles, with all data exchanges formatted in JSON. This standardized approach ensures efficient serialization, platform independence, and straightforward integration with the backend infrastructure.

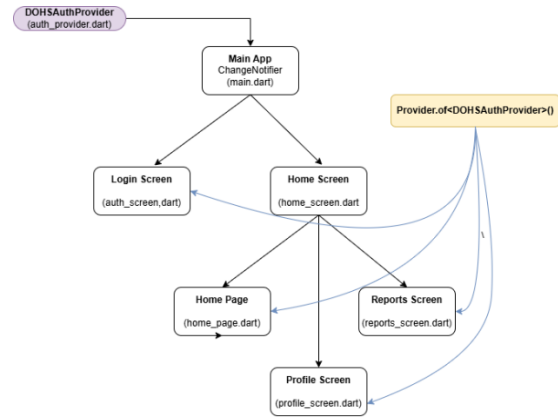


Figure 4. Provider Pattern Workflow for managing Authentication State in Application Lifecycle.

3.5.4. AI Agent Architecture and Prompt Engineering

The AI-assisted conversational reporting module is built on a locally hosted large language model (LLM), specifically DeepSeek-R1:8B, served via Ollama on the backend server. This design eliminates reliance on external API services, preserves data privacy in a health surveillance context, and reduces operational costs typically associated with cloud-based LLM inference.

Core AI logic is implemented directly within the backend server rather than through an external orchestration layer. When a field worker submits a natural language report through the conversational interface, the Flutter client sends the input to the backend via a REST API. The backend then constructs a structured prompt and invokes the locally hosted LLM through the Ollama API. The prompt is engineered to instruct the model to function as a One Health surveillance data extractor, returning structured JSON output containing required fields such as disease name, onset date, confirmation date, symptoms, and outcome status.

Upon receiving the model's response, the backend performs schema validation to ensure that all mandatory fields have been correctly extracted. If any fields are missing, incomplete, or ambiguous, the system initiates a clarification loop by generating targeted follow-up questions, which are sent back to the user through the Flutter interface. This iterative interaction continues until all required fields are sufficiently captured and validated. Once complete, the structured report is processed and stored in the system.

3.6. Evaluation methods

3.6.1. Black-box testing

Functional testing verified that all system components performed according to specifications across both Android and iOS platforms. Test cases covered critical functionalities including authentication, navigation, data submission, and offline operation.

3.6.2. Usability testing

A convenience sample of 25 university students

participated in task-based usability testing. Participants completed structured tasks while metrics were collected on:

- Task completion rates (success/failure).
- Time-on-task (seconds).
- System Usability Scale (SUS) scores.

Participants were selected because they were easily accessible and allowed for rapid evaluation of the system. The university environment allowed for recruitment, testing, and feedback to be done efficiently, multiple times, and in a timely fashion, which helped refine the user interface and get feedback in a controlled manner. This sampling approach, however, makes it difficult to generalize to the larger population of healthcare field workers because the field workers face unique contextual pressures such as time constraints or domain-specific communication. Evaluation with target end-users is therefore still an important goal for future work (as described in Section 5).

4. RESULTS AND DISCUSSION

4.1. System implementation

The development process began with low-fidelity prototyping using Figma, allowing stakeholders to review and provide feedback before full implementation. This approach minimized development rework by addressing design concerns during the planning phase. The final implementation consists of several integrated modules deployed on Android devices for field testing.

4.1.1. Authentication module

The login interface (Figure 5) implements secure credential-based authentication. Field workers access the system using unique usernames and system-generated passwords distributed via secure email channels, ensuring only authorized personnel can submit surveillance data.

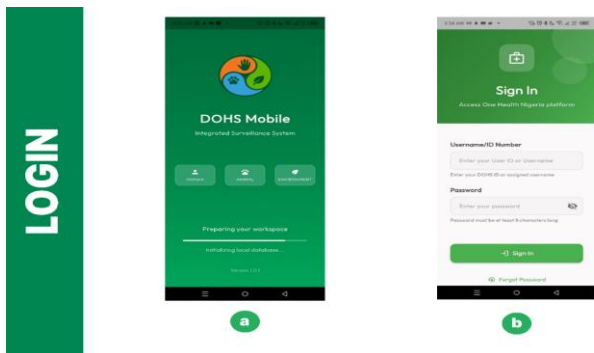


Figure 5. Login Interface

4.1.2. Manual reporting module

The report creation module represents the core functionality of the DOHS application. The traditional reporting interface employs a multi-step wizard approach to guide users through data collection (Figure 6):

Step 1: Disease identification with dropdown menus for standardized disease codes and classification categories.

Step 2: Temporal data entry including dates of onset and confirmation.

Step 3: Clinical details with text areas for symptom descriptions and outcome status selection.

Required field indicators ensure data completeness before submission. This structured approach balances standardized data capture with clinical detail documentation.

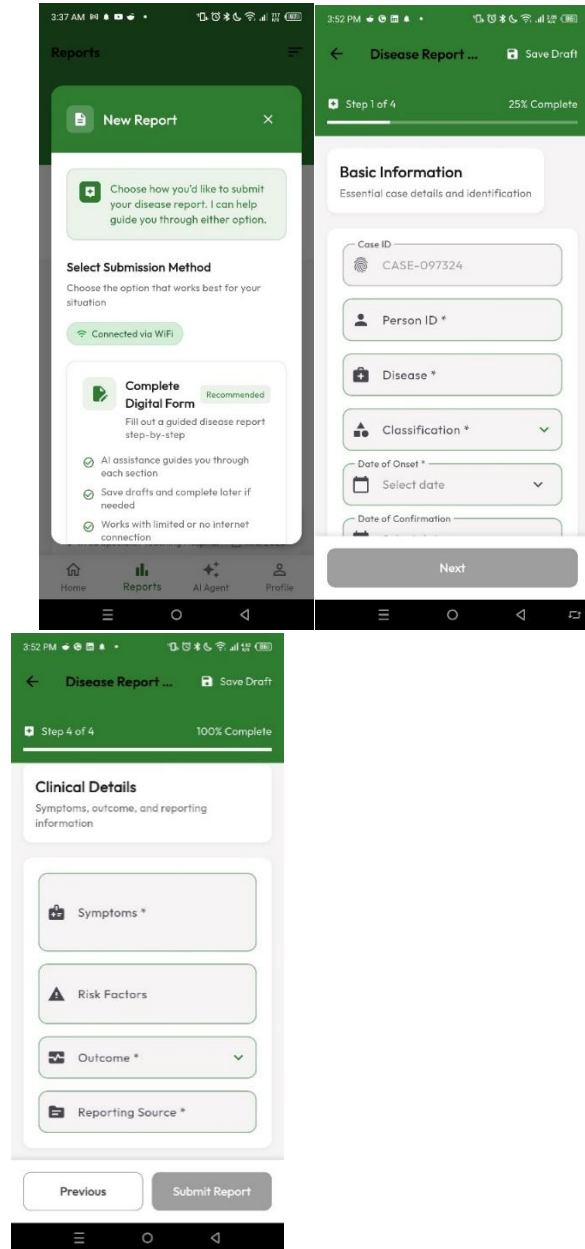


Figure 6. Manual Form Workflow

4.1.3. AI-assisted conversational reporting module

The AI-assisted conversational reporting agent provides an alternative data submission pathway using natural language processing (Figure 7). The chat-style interface reduces the cognitive burden associated with structured forms by allowing field workers to describe cases in natural language.

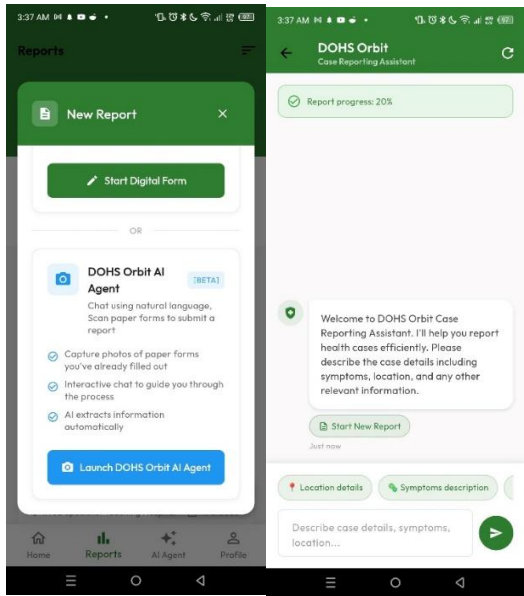


Figure 7. AI Agent for Assisted Reporting Interface.

The AI agent processes user input through the backend server and returns contextual responses (Figure 8):

Success messages: Confirm successful data extraction and report submission as shown in Figure 8.

Clarification requests: Prompt users for missing required fields.

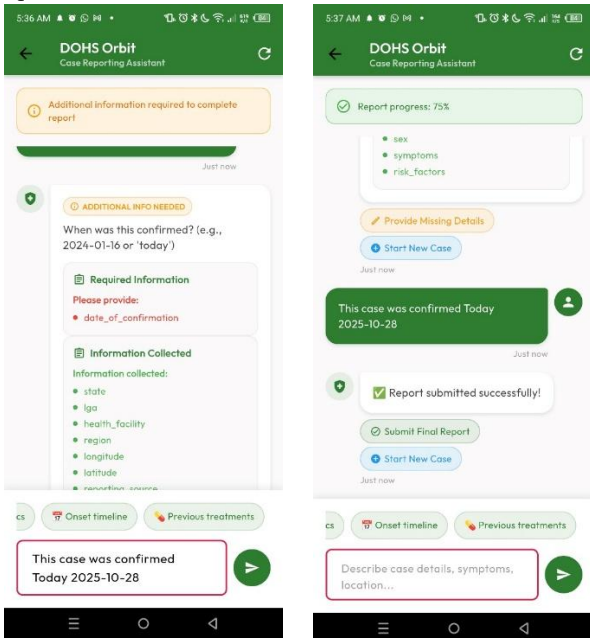


Figure 8. AI reporting agent response message types

4.2. Functional testing results

Black-box testing validated all critical system functionalities across both Android and iOS platforms. Table 3 summarizes the functional test results.

Table 3. Functional testing results

No	Test Case	Result	Status
1	App Launch & Splash Screen	The app opens correctly on both Android and iOS, displaying the DOHS branding and initializing the local database.	Valid
2	User Login & Authentication	The login screen accepts credentials, communicates with the backend, receives a JWT token, and successfully navigates to the Home Dashboard.	Valid
3	Bottom Navigation Bar	Tapping between Home, Reports, and Profile sections correctly switches the main view without reloading the entire app.	Valid
4	Manual Form Reporting	The multi-step form navigates through all steps, validates input, and successfully submits a complete report to the backend (or saves it locally if offline).	Valid
5	AI-assisted conversational reporting	The chat interface sends user messages and displays intelligent replies from the AI agent, eventually resulting in a fully populated and submitted report.	Valid
6	Viewing Report List	The Reports screen correctly displays a scrollable list of the user's past reports, loaded from the local database and/or backend.	Valid

All tested functionalities achieved "Valid" status, indicating the system meets its technical specifications.

4.3. Usability testing results

4.3.1. Task completion analysis

Task-based usability testing with 25 participants revealed differential performance across interface types (Figure 9). Traditional interface tasks demonstrated high success rates:

- Manual form submission: 88%
- Report retrieval: 100%
- Dashboard navigation: 92%

In contrast, the AI-assisted conversational reporting task achieved a 64% success rate, representing a statistically significant difference ($\chi^2 = 4.90, p < 0.05$).

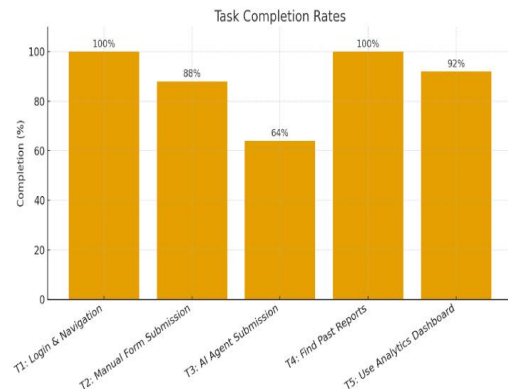


Figure 9. Chart of task completion rates

4.3.2. Efficiency metrics

Time-on-task analysis revealed that successful users completed AI-assisted reporting faster ($M = 132s$, $SD = 47.5s$) compared to manual form submission ($M = 158s$, $SD = 31.2s$). However, the higher standard deviation for AI tasks indicates greater variability in user performance, suggesting an initial learning curve for the conversational interface (Table 4).

Table 4. Task performance metrics

Task	Success Rate (%)	Mean Time-on-Task(s)	Std. Deviation
Manual Form Submission	88	158	31.2
AI Agent Submission	64	132	47.5

The usability results were gained from a student participant sample ($n=25$) and are thus to be regarded in this context. The performance of field-based health workers can differ from what was observed in the field because of domain familiarity and other environmental factors as well as their experience using mobile applications.

4.4. Deployment and knowledge transfer

Following successful evaluation, the application was deployed through multiple channels:

End-user Distribution: The mobile application was made available via the DOHS website for authorized field workers to download and install on their devices.

Documentation: Comprehensive user documentation was developed to explain module functionality and support future system maintenance and feature updates.

Stakeholder Presentation: The completed system was demonstrated to end users and stakeholders, showcasing its functionality and potential impact on One Health surveillance efforts.

Academic Dissemination: Research findings were prepared for journal publication to contribute to the scholarly literature on mobile health surveillance systems and Design Science methodologies.

5. CONCLUSION

This paper utilizes the Design Science Research Methodology to design, develop, and test a mobile One Health surveillance application that helps overcome the challenges of the current paper-based and fragmented reporting processes in the Department of Health Services (DOHS). The system created is based on Flutter, which incorporates structured digital reporting forms, offline data capture, and an AI-assisted conversational reporting feature to improve the reporting efficiency, data quality, and the overall user experience.

System demonstrations established that the application functions as intended in its essential capabilities, and the usability testing yielded a System Usability Scale (SUS)

score of 78.4, indicating good overall usability and acceptance by the users of the application. Though the AI-assisted conversational reporting element showed a steeper learning curve and needs additional improvements, it still showed decreases in reporting time and strong potential for improvement in the future.

In general, the final artifact represents a viable and extensible solution to enhancing real-time One Health information capture and enabling more integrated surveillance in the human, animal, and environmental health sectors. Further enhancements to the conversational reporting agent are planned for future work, along with increased field testing on health professionals and further incorporation of advanced analytics to improve timely detection and response to an outbreak.

Declaration of Ethical Standards

This study did not involve medical, clinical, or invasive procedures. Participant involvement in requirements-gathering and usability evaluation was voluntary, with no sensitive or identifiable personal data collected.

Credit Authorship Contribution Statement

All authors contributed substantially and equally to this study, including conceptualization, methodology, implementation, analysis, and manuscript writing. All authors approved the final manuscript.

Declaration of Competing Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Funding / Acknowledgements

This work was supported by the Tertiary Education Trust Fund National Research Fund grant (TETFund Ref: NRF2023/CC/STI/00031/VOL.1).

Data Availability

Data were collected through interviews and usability testing, and are available from the corresponding author upon reasonable request.

References

- [1] A. Owoyemi et al., "Mobile health approaches to disease surveillance in Africa; Wellvis COVID triage tool," *Digital Health*, vol. 7, p. 2055207621996876, 2021.
- [2] A. O. Babatunde et al., "Leveraging mobile health technology towards achieving universal health coverage in Nigeria," *Public Health in Practice*, vol. 2, p. 100120, 2021.
- [3] S. J. Pitt and A. Gunn, "The one health concept," *British Journal of Biomedical Science*, vol. 81, p. 12366, 2024.
- [4] C. Free, G. Phillips, L. Felix, L. Galli, V. Patel, and P. Edwards, "The effectiveness of M-health technologies for improving health and health services: a systematic review protocol," *BMC Res Notes*, vol. 3, no. 1, p. 250, Dec. 2010, doi: 10.1186/1756-0500-3-250.
- [5] C. Déglise, L. S. Suggs, and P. Odermatt, "SMS for disease control in developing countries: a systematic review of mobile health applications," *J Telemed Telecare*, vol. 18, no. 5, pp. 273–281, July 2012, doi: 10.1258/jtt.2012.110810.
- [6] A. B. Labriue, L. Vasudevan, E. Kochi, R. Fabricant, and G.

- Mehl, "mHealth innovations as health system strengthening tools: 12 common applications and a visual framework," *Glob Health Sci Pract*, vol. 1, no. 2, pp. 160–171, Aug. 2013, doi: 10.9745/GHSP-D-13-00031.
- [7] E. Tambo, E. C. Ugwu, and J. Y. Ngogang, "Need of surveillance response systems to combat Ebola outbreaks and other emerging infectious diseases in African countries," *Infect Dis Poverty*, vol. 3, no. 1, p. 29, Aug. 2014, doi: 10.1186/2049-9957-3-29.
- [8] National Framework for One Health. FAO, 2021. doi: 10.4060/cb4072en.
- [9] D. Destoumieux-Garzón et al., "The One Health Concept: 10 Years Old and a Long Road Ahead," *Front. Vet. Sci.*, vol. 5, p. 14, Feb. 2018, doi: 10.3389/fvets.2018.00014.
- [10] A. L. Okello, K. Bardosh, J. Smith, and S. C. Welburn, "One Health: Past Successes and Future Challenges in Three African Contexts," *PLoS Negl Trop Dis*, vol. 8, no. 5, p. e2884, May 2014, doi: 10.1371/journal.pntd.0002884.
- [11] E. Ahn et al., "A Mobile App and Dashboard for Early Detection of Infectious Disease Outbreaks: Development Study," *JMIR Public Health Surveill*, vol. 7, no. 3, p. e14837, Mar. 2021, doi: 10.2196/14837.
- [12] B. C. Zapata, J. L. Fernández-Alemán, A. Idri, and A. Toval, "Empirical Studies on Usability of mHealth Apps: A Systematic Literature Review," *J Med Syst*, vol. 39, no. 2, p. 1, Feb. 2015, doi: 10.1007/s10916-014-0182-2.
- [13] M. Khosravi, S. M. Mojtabaiean, and M. Aghamaleki Sarvestani, "A systematic review on factors influencing Middle Eastern women's utilization of healthcare services: The promise of mHealth," *SAGE Open Medicine*, vol. 12, p. 20503121241276678, Jan. 2024, doi: 10.1177/20503121241276678.
- [14] Hevner, March, Park, and Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, p. 75, 2004, doi: 10.2307/25148625.
- [15] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.
- [16] T. Tuunanen, R. Winter, and J. vom Brocke, "Dealing with complexity in design science research: a methodology using design echelons," *MIS quarterly*, vol. 48, no. 2, pp. 427–458, 2024.
- [17] B. Gledson, K. Rogage, A. Thompson, and H. Ponton, "Reporting on the development of a web-based prototype dashboard for construction design managers, achieved through design science research methodology (DSRM)," *Buildings*, vol. 14, no. 2, p. 335, 2024.
- [18] D. K. Kim, "Development of Mobile Cloud Applications using UML," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 8, no. 1, 2018.
- [19] T. Ahmad, J. Iqbal, A. Ashraf, D. Truscan, and I. Porres, "Model-based testing using UML activity diagrams: A systematic mapping study," *Computer Science Review*, vol. 33, pp. 98–112, 2019, doi: <https://doi.org/10.1016/j.cosrev.2019.07.001>.
- [20] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, "Application development using flutter," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 2, no. 8, pp. 1262–1266, 2020.
- [21] S. Durai, C. Shyamalakumari and T. Sujithra, "Cloud Computing based Multipurpose E-Service Application using Flutter," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 1122–1126, doi: 10.1109/ICCMC53470.2022.9753968.
- [22] A. M. Sattar et al., 'Accelerating cross-platform development with Flutter framework', *J. Open Source Dev.*, vol. 10, no. 2, pp. 1–11, 2023.
- [23] R. Sugumar, 'Federated AI in Offline-First Mobile Health Architectures for Privacy-Preserving Clinical Intelligence', *Int. J. Sci. Res. Technol.*, vol. 8, no. 4, pp. 14589–14600, 2025.
- [24] A. K. I. Riad et al., 'Enhancing hipaa compliance in ai-driven mhealth devices security and privacy', in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, 2024, pp. 2430–2435.
- [25] R. Rezaee, M. Khashayar, S. Saeedinezhad, M. Nasiri, and S. Zare, 'Critical criteria and countermeasures for mobile health developers to ensure mobile health privacy and security: mixed methods study', *JMIR MHealth UHealth*, vol. 11, p. e39055, 2023.