

Title

**DESIGN AND IMPLEMENTATION OF MALA-LINK: A CITIZEN-CENTRED INTER-
AGENCY SERVICE PLATFORM FOR MALAWI**

Author

HILARYS TEMBO

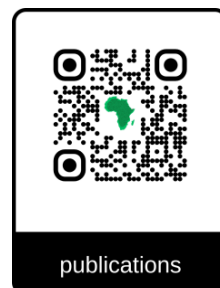
Co-Author

Mrs FANNY CHATOLA



Issued October 2026 Certificate

AR2026DY8URE



ABSTRACT

In order to access essential public and private services such as banking, healthcare, education, voting, employment, mobile money registration, and international travel heavily depends on the timely issuance of the Public services e.g National Identity Card, passport, and license. But Public-service application processes can require citizens to interact with separate administrative workflows and may provide limited visibility into application progress. This paper presents Mala-Link, a citizen-centred platform designed to consolidate selected public-service (like the National registration bureau, Immigration and Directory of Road Traffic Safety Services) application workflows in Malawi through a common web and mobile interface. The web frontend is implemented with Next.js, the mobile application with React Native, and the backend with Node.js and Express. PostgreSQL stores operational application data, accessed through Prisma ORM. The system supports application submission, document uploads, status tracking, administrative status updates, notifications, English and Chichewa language support, and a lightweight blockchain-style audit ledger. The ledger uses SHA-256 hashes and previous-hash links to make inconsistent changes to recorded audit events detectable. It is a single-process, locally persisted ledger rather than a decentralised blockchain network. Functional test cases documented in the project report cover login, application submission, document upload, status tracking, and administrative status updates. Informal user interaction also formed part of development, but no formal survey records are available. The platform does not connect to the production databases of the relevant government agencies; therefore, the work demonstrates an implemented application prototype rather than a live inter-agency government integration. The paper describes the system design, its audit-ledger mechanism, preliminary functional evidence, and the limitations that must be addressed before operational deployment.

KEYWORDS: Blockchain, E-government, digital public services, citizen service

platform, application tracking, audit trail, SHA-256

INTRODUCTION

The National Identity Card, issued by the National Registration Bureau (NRB), serves as the primary legal identity document for Malawian citizens. It is a prerequisite for accessing most government and financial services. Similarly, the Directorate of Road Traffic and Safety Services (DRTSS) issues driving licences and Certificates of Fitness (CoF), while the Department of Immigration handles passport services. Despite the critical importance of these services, Malawian citizens continue to face significant challenges in obtaining and renewing these documents.

BACKGROUND OF THE STUDY

The National Registration Bureau has been grappling with massive backlogs in the processing and printing of National IDs. With approximately two million national IDs expired nationwide, and over three million Malawians expected to renew their IDs during the intensified 2025–2026 renewal exercise, many citizens experience prolonged delays. Rural residents are particularly affected, as they are forced to travel long distances to registration centres, only to endure long queues and repeated visits due to missing documents, biometric mismatches, printer shortages, and vague feedback such as “come back next week.” This leads to delayed processing, and poor communication with citizens.

RESEARCH OBJECTIVES

General Objectives

- To Design and Develop a citizen-centered inter agency system for National

Registration Bureau, Immigration and Directory of Road Traffic Safety Service.

Specific Objectives

- To Implement a Category-Based Notification System for Citizen alert services
- To Develop a Real-Time Application Tracking System within NRB, Immigration and DRTSS
- To Integrate a secure Digital Document Upload and Storage for document submission by the citizens
- To enable interoperability between key institutions which are the NRB, Immigration and DRTSS

LITERATURE REVIEW

Alzoubi, Y., and Mishra, A. (2025), in their journal titled “*A comparative analysis of e-government services in ten African countries*”, focused on demonstrating how “information integration”—connecting various departments through a single virtual interface—drastically increases citizen convenience and decreases operational costs. But their research was strictly analytical and did not provide a deployable software solution for these integration challenges. We aim to address this by practically developing a centralized system that acts as the technical bridge connecting NRB, DRTSS, and Immigration databases into one usable platform.

Molson, J. and Glorindal, G. (2023), In their study titled “*Online National Citizen's ID Renewal System*”, focused on developing a web-based portal specifically for renewing national identity cards in Malawi. The system allows citizens to submit renewal applications online in a single session, including uploading supporting documents and basic status tracking, with the goal

of eliminating repeated physical visits to NRB offices and reducing long waiting times.

They emphasised secure online submission, error reduction in data entry, and improved record management through a centralised database. But the system was scoped only to national ID renewal and lacked real-time multi-stage tracking, automated SMS or app notifications for collection readiness and integration with other services such as driving licences, passports, or banking KYC processes.

UNDP portfolio report (IDT4M, 2026) describes Malawi’s **National Registration and Identity System (NRIS)** as the country’s “*foundational digital public infrastructure*” for civil registration and identity. NRIS continuously registers births, deaths, and issues biometric national IDs, enabling secure identity verification across agencies. It is indeed a robust back-end system. However, the report is descriptive and technical – it does not include any service-delivery module. It focuses on database and process, not on the front-end user experience. Our integrated platform is designed to complement NRIS: we will *connect* citizens to this identity infrastructure via web/mobile portals and agency APIs. In doing so, we fill the gap between a strong digital ID registry (NRIS) and an actual multi-service delivery channel, turning that identity into streamlined services for the public.

In their study titled “*Future of e-Government: An Integrated Conceptual Framework*”, **Malodia, S. et al. (2021)**, focused on conceptualizing a multi-dimensional e-government model that integrates back-end stakeholders for seamless front-end citizen services. They examined real-world examples such as India’s Passport Seva portal, which enables online application, document upload, biometric appointment booking, and status tracking for passport services. The framework emphasised hyper-integration across government departments to deliver empowered citizenship and reduced bureaucracy. But the model was largely conceptual and country-specific (India), lacking practical implementation

details for African contexts, hybrid SMS/web notifications tailored to processing stages, and full coverage of interconnected services like national ID, driving licences, and financial KYC in a single platform.

In their study titled “*Evaluating E-Government Development among African Union Member States*”, **Tiika, B.J. et al. (2024)**, focused on analysing the progress and impact of e-government platforms across Africa, highlighting successful systems such as Rwanda’s Irembo (for birth certificates, driver’s licences, and business registration) and South Africa’s eNatis (for vehicle licensing and renewals). The study detailed how these platforms reduce paperwork, queues, and processing times through online portals and digital service delivery. They stressed the role of robust policies in improving public administration efficiency.

But the platforms operate in silos without a unified citizen hub, real-time cross-agency tracking, category-based personalised notifications for collection readiness, or seamless integration with national ID databases and banking services, leaving gaps in Malawi-like contexts with high rural populations and multiple expired documents.

METHODOLOGY

The Mala-Link system was developed using the Agile Software Development Methodology, which proved to be the most suitable approach for this project. Agile was chosen because of the dynamic nature of the requirements — integrating multiple government agencies (NRB, DRTSS, and Immigration), evolving stakeholder needs, and the necessity to deliver a user-centric platform in a resource-constrained academic environment.

Research and Development Approach

This study followed a software design and implementation approach to develop Mala-Link, a

citizen-centred platform intended to consolidate access to selected public-service application workflows in Malawi. The work translated identified service-delivery needs into a functional software system and examined whether selected features operated as intended. The development process involved requirements identification, system design, implementation of the principal modules, and functional checks. The system supports citizen account access, application submission, digital document uploads, application status tracking, notifications, and administrative management. English and Chichewa language support is included in the citizen-facing interface. The objective was to demonstrate the technical feasibility of bringing these citizen-facing workflows into a common platform. The study did not involve deployment as an officially connected government service or a controlled measurement of government service-delivery outcomes.

SYSTEM DESIGN AND DEVELOPMENT

System Architecture and Technologies

Mala-Link uses a web and mobile architecture consisting of client interfaces, a backend API, a relational database, and a separate audit-ledger service. The web frontend was developed with Next.js and the mobile application with React Native. The backend was implemented using Node.js and Express. PostgreSQL stores operational data, while Prisma ORM provides the application’s database-access layer. The backend exposes functionality to the web and mobile clients and coordinates operations involving application records.

The blockchain audit service is a secondary logging component. PostgreSQL remains the system of record for citizens, applications, forms, and cards. The audit ledger records selected events indicating that an operation occurred; it does not replace the operational database or store the full application record. The ledger is persisted as a JSON file at `api/data/blockchain.json`. The architecture is therefore a single-process, locally

persisted, blockchain-style audit ledger, not a distributed or decentralised blockchain network.

Functional Modules

- **User management and access control.** Supports user registration and authentication and provides access to citizen or administrative functionality. The project documentation describes password hashing, token-based authentication, and role-based access control.
- **Application management.** Allows citizens to submit service-specific application information and supports the creation and management of application records.
- **Document submission.** Provides a mechanism for submitting supporting documents and associating them with application workflows.
- **Application tracking and notifications.** Allows citizens to view application status and includes notification functionality for relevant application events.
- **Administrative management.** Allows authorised administrative users to review application records and update application statuses.
- **Web and mobile access.** Provides a Next.js web client and a React Native mobile client for interacting with the platform.
- **Language support.** Includes English and Chichewa translations in the citizen-facing interface.
- **Blockchain-style audit ledger.** Records selected application events in a hash-linked sequence and provides a validation operation for detecting inconsistent changes to the ledger.

Blockchain Audit Ledger Design

The audit component is a lightweight, educational, append-only ledger that adopts selected

blockchain data-structure and integrity-verification ideas. It is not a distributed blockchain: it uses one process and one persisted JSON file, without peer-to-peer networking, a consensus protocol, mining, cryptocurrency, or independent validator nodes. Its purpose is to make inconsistent alterations to recorded audit events detectable, rather than to provide decentralised trust between independent organisations.

Block Structure and Hashing

Each block contains an index, an ISO-8601 timestamp, an event-data object, a previousHash field, and a hash field. The hash is calculated with SHA-256 over the block index, timestamp, JSON representation of the event data, and previousHash. SHA-256 is supplied by Node.js's built-in crypto module, avoiding the need for an additional cryptographic package.

The first block is a genesis block with index 0, previousHash set to the literal string "0", and data identifying it as a GENESIS event. Each subsequent block stores the hash of the block immediately before it in previousHash. Since a block's own hash includes its previousHash value, the sequence links each block to its predecessor.

Chain Validation

The isChainValid() operation traverses the ledger and checks four conditions: (1) the genesis block has the expected index and previousHash value and its hash can be recalculated successfully; (2) each later block has an index exactly one greater than its predecessor; (3) each stored block hash matches a fresh recalculation from its current fields; and (4) each block's previousHash equals the actual hash of the preceding block. The validator stops at the first detected inconsistency and returns a diagnostic result identifying the invalid block and reason, or a valid result with the

block count.

Recorded Events and Data Minimisation

The ledger records four event categories: APPLICATION_SUBMITTED when a citizen successfully creates an application; STATUS_CHANGED when staff change an application status to a status other than REJECTED, including COLLECTED; APPLICATION_REJECTED when an application is rejected; and CARD_ISSUED when staff issue an identity card for an application in the READY state. The implementation intentionally does not define an APPLICATION_APPROVED event because the application-status model has no distinct approval state.

An event includes an event identifier, event type, application identifier, actor identifier, relevant status fields, and a timestamp. The ledger does not include full names, national identity numbers, passport numbers, addresses, passwords, password hashes, document contents, or other application-form contents. This data-minimisation decision avoids duplicating sensitive identity data in a log designed to preserve historical event evidence.

Write Ordering and Error Handling

The implementation appends an audit event only after the corresponding PostgreSQL write succeeds. Call sites use blockchainService.safeRecord(). If writing to the ledger fails, safeRecord() catches the error, logs it, and returns a warning string that is attached to the response as auditWarning. The main application operation is not rolled back solely because the audit-log write fails, but the logging failure is surfaced rather than silently ignored. This design favours continuity of the primary application workflow; it also means that an operational database update can succeed even when the corresponding audit event is not persisted, which should be considered in monitoring and

reconciliation.

Data Management and Inter-Agency Scope

PostgreSQL is the operational data store, accessed through Prisma ORM. The audit ledger is a secondary record of selected events and does not replace the database. Although the platform is designed around services associated with the National Registration Bureau, the Directorate of Road Traffic and Safety Services, and the Department of Immigration, it does not connect to those agencies' production databases. The implementation therefore demonstrates consolidated application workflows in its own environment rather than verified interoperability with official government systems.

Evaluation Procedure

The evaluation evidence consists of functional test cases recorded in the project report and informal user interaction during development. The documented functional tests cover authentication, application submission, document upload, status tracking, and administrative status updates. Informal interaction was used as formative input, but no formal survey records, structured participant dataset, or documented quantitative usability measurements are available. Accordingly, this study does not report a participant count, satisfaction percentage, average usability score, or statistically analysed user preference.

The evaluation is limited to selected functionality exercised in the implemented environment. It does not establish reductions in government-office queues, official processing times, or administrative workload. Such outcomes would require a separate study with a defined measurement protocol, suitable baseline data, and authorised participation by the relevant agencies.

RESULTS AND EVALUATION

Functional Test Results

The project report records five functional test cases as passing. These tests cover important interactions between citizens, application records, uploaded documents, and administrative workflows. Table 1 summarises the outcomes as recorded in the project report; the table should be read as reported functional evidence, not as independently audited test results.

The recorded outcomes indicate that the tested workflows operated as expected in the environment described by the project report. The tests cover the main sequence of submitting an application, attaching supporting documentation, checking its status, and updating the status through an administrative interface. However, the available documentation does not provide sufficient underlying test artefacts to independently verify the full test environment, execution dates, number of executions, failure rates, or total application coverage. Additional testing is needed to evaluate edge cases, security, reliability, performance, and cross-device compatibility.

Blockchain Integrity Validation

The ledger implementation provides a validation operation that recalculates block hashes and checks the links between consecutive blocks. According to the implementation specification, a healthy chain returns a valid result and block count, while an inconsistent block returns a diagnostic result identifying the first invalid block and the reason for failure. This mechanism is designed to detect changes that leave the stored hash inconsistent with the block data or that break a previous-hash link.

A reproducible demonstration can be performed by creating several real audit events, validating the ledger, manually changing an event field in a non-final block without changing its stored hash, and validating again. The expected outcome is that validation reports an invalid chain. This is a described test procedure, not a claim that a new

experiment was executed for this paper. Before submission, the authors should run the procedure and preserve the actual validation output if they wish to report it as an observed result.

Interpretation of the Blockchain Design

The principal contribution of the audit component is the application of hash-linked records and integrity validation to selected application events. It provides a way to detect inconsistent edits to the persisted ledger without placing the full operational record on the ledger. The explicit exclusion of sensitive personal data from event payloads is particularly relevant to a system that handles identity-related applications.

The security claim must nevertheless remain bounded. Because the ledger is stored in a single JSON file on one server and has no independent nodes or consensus protocol, an attacker with sufficient access to rewrite the complete file and recompute all hashes could create a self-consistent replacement ledger. The validation mechanism detects inconsistent tampering; it does not by itself prove that the ledger has never been rewritten by an actor with full write access. The implementation should therefore be described as a blockchain-style, tamper-evident audit ledger rather than a decentralised blockchain or a tamper-proof system.

Informal User Feedback

Informal user interaction and feedback formed part of the development process. Such interaction can help identify confusing workflows and practical interface issues. However, because no formal survey records or structured participant responses are available, the evidence cannot support a defensible participant count, satisfaction percentage, average usability score, or statistically supported claim about user preferences. The feedback is therefore treated as formative input rather than as a formal user study.

Limitations

- Functional evidence covers five documented test cases and does not establish exhaustive coverage of the application.
- Informal feedback was not collected through a formal instrument, limiting conclusions about usability and satisfaction.
- No recorded performance measurements are available to support claims about response time, throughput, scalability, or resource consumption.
- Mala-Link is not connected to the production databases of the relevant government agencies; operational interoperability and authoritative identity verification were not demonstrated.
- The audit ledger is single-process and locally persisted. It has no decentralisation, consensus mechanism, peer-to-peer networking, or Byzantine fault tolerance.
- The ledger validation mechanism detects inconsistent alterations, but cannot prevent an attacker with full write access from replacing the entire ledger with a consistently recomputed version.
- Because the audit write occurs after the operational database write, a ledger failure can leave a successful application operation without a corresponding persisted audit event; the implementation surfaces a warning, but reconciliation and monitoring remain important.

DISCUSSION

Mala-Link brings web and mobile access, application management, document submission, status tracking, administrative functions, language support, and a blockchain-style audit component together within one software platform. The five documented functional test cases provide preliminary evidence that selected application

workflows operate as intended.

The audit ledger is a notable part of the implementation because it makes selected application events traceable through a hash-linked sequence and supplies a mechanism for detecting inconsistent changes. Its use of event identifiers, actor identifiers, status fields, and timestamps—rather than full application contents—also limits the amount of sensitive information duplicated in the ledger. However, the design does not provide decentralised trust or protection against wholesale ledger replacement by an attacker who controls the server and ledger file.

The platform's consolidated interface may provide a foundation for improving the organisation of citizen-facing workflows, but technical implementation alone does not prove improvements in public-service outcomes. The absence of live agency integration means that the current version should not be interpreted as a replacement for official government registration or document-processing systems. Future work should prioritise authorised agency integration, additional security and privacy testing, reproducible audit-ledger validation, and structured usability and performance evaluation.

CONCLUSION

This paper presented the design and implementation of Mala-Link, a citizen-centred web and mobile platform intended to consolidate selected public-service application workflows in Malawi. The implemented functions include application submission, document upload, status tracking, administrative status management, notifications, and English/Chichewa language support. The project report documents five functional test cases that passed; this provides limited evidence that selected workflows operate, but it does not establish comprehensive reliability, usability, scalability, or readiness for national deployment.

A distinctive implementation element is the lightweight blockchain-style audit ledger. It

records selected application events in hash-linked blocks, uses SHA-256 to calculate block hashes, and provides chain-validation checks for detecting inconsistent changes to stored records. The design also minimises data placed in the ledger by excluding direct personal identifiers and document contents. These choices demonstrate how tamper-evident audit concepts can be incorporated into a conventional application without replacing the operational PostgreSQL database.

The ledger must not be interpreted as a decentralised or tamper-proof blockchain. It is persisted in a single JSON file on one server, has no distributed consensus, and cannot reliably resist an attacker who can rewrite the entire file and recompute its hashes. In addition, an audit-write failure can occur after a successful database operation, leaving an application change without a corresponding ledger entry. Mala-Link is also not connected to live production databases or APIs of the relevant government agencies, so operational inter-agency integration has not been demonstrated.

Future work should include a reproducible ledger-integrity experiment with recorded outputs, broader and documented usability testing, security testing, and measurements of response time and reliability. For a production deployment, the audit component would require stronger persistence and access controls, monitoring and recovery procedures, and a clearly defined consistency strategy between operational transactions and audit records. Any real inter-agency deployment would additionally require formal institutional agreements, approved integration interfaces, privacy and data-governance controls, and evaluation with intended users. Within these boundaries, Mala-Link provides an implemented prototype and a practical basis for further investigation of citizen-facing digital public services and auditable application workflows in Malawi.

REFERENCES

1. Alzoubi, Y. I., & Mishra, A. (2025). A comparative analysis of e-government services in ten African countries. *JeDEM – eJournal of eDemocracy and Open Government*, 17(3), 172–200. <https://doi.org/10.29379/jedem.v17i3.1107>
2. Malodia, S., Dhir, A., Mishra, M., & Bhatti, Z. A. (2021). Future of e-government: An integrated conceptual framework. *Technological Forecasting and Social Change*, 173, Article 121102. <https://doi.org/10.1016/j.techfore.2021.121102>
3. Molson, J., & Glorindal, G. (2023). Online national citizen's ID renewal system. *Journal of Information Systems in Developing Countries*.
4. Olumekor, M., Mangai, M. S., Madumo, O. S., Mohiuddin, M., & Polbitsyn, S. (2025). Influences on e-governance in Africa: A study of economic, political, and infrastructural dynamics. *Public Administration*. <https://doi.org/10.1111/padm.13013>
5. Tiika, B. J., Tang, Z., Azaare, J., Dagadu, J. C., & Otoo, S. N.-A. (2024). Evaluating e-government development among Africa Union member states: An analysis of the impact of e-government on public administration and governance in Ghana. *Sustainability*, 16(3), Article 1333. <https://doi.org/10.3390/su16031333>
6. United Nations Department of Economic and Social Affairs. (2024). *United Nations e-government survey 2024: Accelerating digital transformation for sustainable development*. United Nations. <https://desapublications.un.org/publications/un-e-government-survey-2024>
7. United Nations Development Programme. (2026). *IDT4M portfolio report: National Registration and Identity System (NRIS) in Malawi*.