

Title

**AN AI-DRIVEN MULTIMEDIA FAULT REPORTING AND ANTI-VANDALISM
SYSTEM FOR MALAWI'S PUBLIC INFRASTRUCTURE**

Author

WAKISA NYONDO

Co-Author

MR. PEMPHO JIMU

ABSTRACT

Public infrastructure faults and vandalism can disrupt essential services, increase repair costs, and delay service restoration. In Malawi, reporting and follow-up may rely on phone calls, written notices, and manual coordination, making it difficult to collect consistent evidence and communicate incident locations promptly. This paper presents an AI-Driven Multimedia Fault Reporting and Anti-Vandalism System designed as a software platform for citizens, administrators, and responsible public institutions. The system is accessible through a web platform and mobile application and allows users to submit incident descriptions, photographs, audio files, and location information. AI-assisted classification analyzes submitted report information to help distinguish likely technical faults from suspected vandalism and supports severity assessment and administrative review. The platform includes offline report preparation and synchronization, geolocation mapping, report tracking, notifications, and an Emergency Response Module for prioritizing critical incidents and coordinating follow-up. The system uses existing user devices and ordinary connectivity; it does not require dedicated cameras, sensors, or Internet of Things installations. Development followed an Agile approach, and the project documents functional, security, performance, recovery, and usability testing. The reported test plan marks functional, security, performance, and recovery testing as tested, while usability testing is recorded as not fully tested. The project provides a centralized software workflow intended to improve the quality of submitted reports, strengthen coordination, and support more timely responses to infrastructure faults and suspected vandalism.

KEYWORDS: multimedia incident reporting, infrastructure faults, anti-vandalism, AI-assisted classification, emergency response, offline reporting, geolocation, Malawi.

INTRODUCTION

Public infrastructure underpins the delivery of electricity, water, public safety, and other essential services. In Malawi, institutions such as the Electricity Supply Corporation of Malawi (ESCOM), the Malawi Police Service, regional water boards, local councils, and government departments are responsible for responding to faults and incidents affecting public assets. When faults or suspected vandalism are reported late, the consequences can include service disruption, increased maintenance

costs, property damage, and reduced public confidence.

Digital reporting systems can help organize incident information and improve communication between the public and responsible institutions. The project described in this paper combines a web platform and mobile application with multimedia evidence, location information, AI-assisted classification, report tracking, and alerts. Rather than continuously monitoring infrastructure, the platform processes reports submitted by users and supports the responsible personnel in reviewing and following up on cases.

BACKGROUND OF THE STUDY

Traditional incident reporting methods often depend on calls, written records, staff observations, or physical inspections. These methods may produce incomplete descriptions, inconsistent records, missing location details, and limited visibility of an incident's progress. A report may be received by one office without being easily available to the organization responsible for repair or investigation. Centralized digital records and mobile access can help reduce this fragmentation by allowing users to attach evidence, supply a location, and check a report's status.

Research on e-government and mobile government in Malawi has considered the readiness, adoption, benefits, and challenges of digital public services [1-3]. Work on records management in Malawi's water sector also highlights the importance of organizing institutional information for reliable access and administration [4]. These concerns provide a relevant context for a software platform that centralizes infrastructure incident reporting and supports follow-up across different entities.

PROBLEM STATEMENT

The problem addressed by the project is the difficulty of reporting, classifying, locating, and following up on infrastructure faults and suspected vandalism through fragmented or manual processes. Reports may lack useful evidence or precise location information, and responsible institutions may not receive consistent alerts or a clear record of progress. The project responds to this problem with a software-based workflow that accepts user-submitted descriptions, photos, audio, and location details, applies AI-assisted classification, and provides administrative tools for prioritization, status tracking, and response coordination.

RESEARCH OBJECTIVES

The main objective is to develop a software-based multimedia reporting system that improves the management of infrastructure faults and suspected vandalism in Malawi. The specific objectives are to:

- support AI-assisted classification of submitted incidents as likely technical faults or suspected vandalism;
- enable users to submit descriptions, photos, audio evidence, and location information through the web platform and mobile application;
- provide offline report preparation and synchronization when connectivity returns;
- generate alerts and support emergency response and follow-up for critical reports; and
- evaluate core system functions, security, performance, recovery, and usability through testing.

SIGNIFICANCE AND SCOPE OF THE STUDY

The project is significant because infrastructure reporting is a coordination problem as well as a technical problem. A fault may be known to a member of the public but remain unknown to the organization responsible for addressing it. An incident may also be reported without a clear location, supporting evidence, or a record of subsequent action. By giving users a common digital channel and providing administrators with a central place to review incidents, the system is intended to improve the completeness, consistency, and traceability of reports. These are practical improvements even before any claim is made about the speed or accuracy of an AI model.

The expected beneficiaries include citizens who need a way to communicate infrastructure problems, administrators who receive and allocate reports, and technical or security personnel who investigate incidents. Entity-based access is important because different organizations may be responsible for different categories of infrastructure. A single reporting platform can organize information centrally while allowing access to be controlled according to administrative roles and entity assignments. The platform is designed to support existing institutional work rather than replace the judgment of personnel or the operational procedures of the responsible agency.

The scope of the project is limited to a software-based reporting and incident-management workflow. Users submit descriptions, photographs, audio evidence, and available location information using the web platform or mobile application. AI-assisted

classification helps interpret the information that has been submitted. The system can provide alerts, manage report status, and retain activity records for administrative follow-up. It does not autonomously watch public infrastructure, operate dedicated surveillance cameras, or depend on fixed sensors and Internet of Things equipment. Its performance therefore depends partly on users submitting meaningful reports and on the receiving institutions acting on the information provided.

The project also addresses constraints that are relevant in everyday use. Internet connectivity may not always be available at the place where an incident is observed. Offline report preparation and later synchronization allow the reporting process to begin without a live connection, although server-side analysis and notification depend on the report reaching the online system. Similarly, location data may be unavailable or permission may be denied; for that reason, the user can select a position manually on a map. These alternatives make the workflow more practical without requiring new hardware.

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Functional requirements describe actions that the system must support. The principal requirements documented by the project include account registration and login; role-based access; incident submission; attachment of photo and audio evidence; location selection or capture when permitted; offline preparation and synchronization; AI-assisted classification; severity and priority review; alert generation; report status tracking; user and entity administration; and access to summaries and logs. These requirements are related. For example, an uploaded file must be attached to the correct incident, the incident must be assigned to the relevant entity, and the alert must be associated with that report so the administrator can understand why it was generated.

Non-functional requirements concern the quality and reliability of these operations. Security requires that only authorized users access protected records and that administrator permissions are limited to the proper scope. Reliability requires the system to handle incomplete submissions, upload failures, and temporary loss of connectivity without silently losing report information. Usability requires forms, instructions, report status, and alerts to be understandable to people with different levels of technical experience. Performance concerns how quickly common actions complete and whether the system remains responsive as usage increases.

Maintainability requires the application modules to be organized so that updates to reporting, analysis, or notification logic can be introduced without rebuilding the entire system.

The distinction between functional and non-functional requirements is important for testing. A successful submission test verifies that a report is accepted and stored. A security test verifies that a user cannot access an administrator-only function. A recovery test checks what happens if the network or upload process fails. A usability assessment checks whether users understand how to submit evidence, select a location, and recognize the report status. Each form of testing examines a different aspect of system quality and should be reported separately rather than combined into a single general statement that the system works.

LITERATURE REVIEW

E-government and mobile government research provides a foundation for using digital platforms to improve access to public services and institutional coordination. Studies focused on Malawi discuss implementation levels, readiness, adoption, and challenges associated with public-sector digital services [1-3, 5]. Such work is relevant because infrastructure reporting is not only a technical task; it also depends on whether users and institutions can access and use a shared reporting workflow.

Digital records management is another relevant area. Reliable incident management requires reports, evidence, locations, status updates, and administrative actions to be stored in a retrievable form. Research on records management at the Northern Region Water Board in Malawi is pertinent to the need for organized institutional records [4]. A centralized system can support this need by connecting report data with the entity responsible for reviewing the case.

Artificial intelligence and computer vision provide methods for interpreting data submitted to software applications. Deep learning literature explains how models can learn useful representations from training data [6, 7], while computer-vision material describes techniques for processing images [8]. In the proposed system, AI is used as an aid to classifying user-submitted reports, not as autonomous surveillance. Classification results are intended to support administrator review and should not replace verification by authorized personnel.

Incident-reporting frameworks also illustrate the importance of consistent categories and structured

records. The National Incident-Based Reporting System (NIBRS) technical framework is an example of structured incident reporting practices [9]. For the Malawi context, broader digital transformation work emphasizes that software tools must be practical within existing organizational processes and available digital infrastructure [10]. The present project applies these general ideas to public infrastructure fault and suspected-vandalism reports.

Centralized Records and Traceability

A central record can help reduce the fragmentation that occurs when incident information remains in separate telephone conversations, paper records, personal messages, and departmental spreadsheets. A structured digital record brings the description, evidence, location, assigned entity, classification output, alert history, and status changes together. This does not guarantee that every incident will be resolved quickly, but it makes it easier for an authorized person to find the latest recorded information and understand the sequence of actions taken. Audit and activity logs provide an additional record of changes performed within the application.

Multimedia Evidence and Incident Interpretation

Text descriptions alone may omit details that influence the understanding of an incident. A photograph can provide visible context about a damaged asset, while audio can allow a user to describe observations when typing is inconvenient. The value of these inputs depends on relevance, quality, and their connection to the correct report. The upload module must therefore validate files, maintain the association between attachments and the incident record, and make evidence available to authorized reviewers. AI analysis may use supported submitted media, but the results still require interpretation in context and should not be treated as an independent confirmation that vandalism has occurred.

AI-Assisted Classification and Human Review

Machine-learning literature describes methods for learning patterns from labelled or structured data [6-8]. In this project, the intended role of AI is to help classify submitted reports into likely technical faults or suspected vandalism and support severity assessment. Such a classification can help direct a report to the appropriate review pathway, but it cannot by itself establish the cause of a fault or identify a responsible person. Classification quality depends on the quality and representativeness of the underlying data, the selection of categories, and the

way uncertain cases are handled. Human review is therefore an important part of the workflow, especially where an incorrect classification could delay emergency attention or create an unjustified suspicion.

Offline and Mobile Access

Mobile government research is relevant because the reporting interaction often occurs away from an office computer [1, 3, 5]. A mobile application can make it more convenient for a user to submit evidence at the point where an incident is observed. However, mobile access alone does not address all connectivity limitations. Offline preparation and later synchronization provide a way to hold report details locally until a connection is restored. For dependable operation, the application should make clear whether a report is saved locally, pending synchronization, or confirmed as received by the central service. Users should not be led to believe that an offline report has already reached an authority when it is still stored on the device.

Position of the Present Project

Taken together, the reviewed areas point to a system that combines digital access, structured incident records, evidence handling, location information, assisted classification, and administrative workflow. The project's contribution is the integration of these functions into one software-based platform adapted to reports about public infrastructure in Malawi. The contribution should be understood as a practical system design and implementation rather than evidence of a measured nationwide reduction in infrastructure faults. Demonstrating that wider outcome would require operational deployment data, comparison with baseline processes, and evaluation across participating institutions.

METHODOLOGY

The project used an Agile development approach. Agile divides development into manageable cycles that include planning, implementation, testing, review, and improvement. This approach fits a modular reporting system because authentication, multimedia upload, classification, mapping, alerts, offline reporting, and administration can be developed and checked as related components before being integrated into the overall application.

Research Design Approach

The work follows a design-and-development approach: identify limitations in existing reporting practices, define functional requirements, design the

software architecture, implement the application modules, and evaluate the integrated system through user-facing tests. The main artifact is a web and mobile software platform that processes submitted reports and stores results centrally. No dedicated camera, sensor, edge device, or other IoT equipment is part of the implementation.

System Development Process

Development activities covered requirements analysis, interface and architecture design, module implementation, integration, testing, and refinement. The application includes user authentication and role-based access, incident submission, multimedia uploads, AI-assisted analysis, GPS or map-based location, offline synchronization, notifications, report status tracking, dashboards, and administration. These functions form a software workflow from initial report preparation through classification, review, and follow-up.

Agile Development Cycles

The Agile approach organizes work into short cycles so that individual functions can be built and reviewed before they are integrated. For this project, the work can be understood in a sequence of related cycles. Initial planning identifies users, incident categories, the organizations that may receive reports, and the basic data that each report must contain. Interface design then establishes how users sign in, describe an incident, attach evidence, identify a location, and review submission status. Module development implements these interactions and connects them to storage, classification, and alert functions. Integration work checks that information moves correctly between modules. Testing and refinement are used to find errors and improve the user workflow before release.

The value of an iterative process is that a reporting application is not one isolated form. Authentication determines what a person can do; entity assignment influences where a report is routed; multimedia uploads need a stable connection to the report record; AI classification requires the report data; and alerts depend on the report's category, status, and location information. Developing and testing related pieces in cycles allows interface and data-flow issues to be identified earlier. The available project report identifies Agile as its development methodology, but it does not provide sprint dates, team velocity, or detailed iteration metrics, so no quantitative claims about the development process are made here.

Research Design and Evaluation Logic

The work follows a design-and-development research approach. The practical artifact is the software system, while the evaluation asks whether its documented functions behave as intended when users interact with them. Black Box Testing is suitable for checking observable operations such as registration, report submission, file upload, alerts, offline synchronization, and status tracking. This form of testing does not establish the internal correctness of every AI model or prove long-term operational impact. Separate evidence is needed to evaluate classification accuracy, false-positive rates, response time, and user satisfaction. The study therefore distinguishes between implementing a capability, confirming its basic function, and demonstrating its real-world effectiveness.

Stakeholders and User Roles

The main user groups are ordinary users who submit reports, administrators who review reports for an assigned entity, and higher-level administrators who manage system-wide settings or accounts. Responsible organizations can include electricity services, water boards, police, local councils, and public-property departments. These groups do not necessarily need identical permissions. A citizen should be able to submit a report and view its recorded progress, while an administrator may need to verify details, update statuses, send feedback, or assign follow-up. Entity-scoped administration helps limit exposure of incident records to staff who do not need access to them. The precise operational responsibilities of each participating institution remain a matter for local configuration and policy.

SYSTEM ARCHITECTURE

The architecture is organized around four cooperating software elements: the presentation layer, application or logic layer, database layer, and AI-assisted processing services. The presentation layer provides the web and mobile interfaces used by citizens and administrators. The application layer handles authentication, report validation, multimedia submission, location processing, offline synchronization, classification requests, status updates, and alerts. The database layer stores user accounts, incident reports, uploaded evidence, location data, classification outputs, notifications, and activity records. AI-assisted services process information supplied with reports and return outputs for use in the incident workflow.

A typical workflow begins when a user enters an incident description and attaches available photo or audio evidence. The user provides location information through device permissions where available or selects a location on a map. When the user is offline, the application can retain a report for later synchronization. Once submitted, the report is stored and made available for AI-assisted classification and administrative review. If the case requires urgent attention, the notification and Emergency Response Module helps the responsible personnel prioritize the case and coordinate follow-up.

Presentation Layer

The presentation layer includes the screens and forms used to register, sign in, submit an incident, attach evidence, choose a location, view personal reports, and manage administrative tasks. It should provide clear feedback after each action, such as whether a report is saved, uploaded, pending synchronization, or successfully submitted. The distinction is particularly important on mobile devices and during intermittent connectivity. Input forms should present required fields consistently and make error messages understandable, rather than relying on technical messages that ordinary users may not know how to interpret.

Application and Business-Logic Layer

The application layer coordinates the workflow and enforces rules. It checks the submission, verifies the user and applicable permissions, associates the report with an entity, stores the information, invokes the AI-analysis process, tracks status, and sends alerts where appropriate. It is also responsible for deciding how incomplete submissions and failed operations are handled. Keeping these functions in a logical layer helps separate the user interface from the rules controlling report processing. The API routes shown in the submitted code sample indicate separate backend paths for authentication, reports, vandalism records, uploads, analysis, and statistics, though the report does not document every implementation detail of each route.

Database and Evidence Storage

The database layer maintains the records needed for account management and incident follow-up. The project describes storage for user accounts, entity assignments, incident reports, alert management, AI-analysis results, and activity logs. A report record needs to remain associated with its uploader, responsible entity, location, current status, and

supporting media. These relationships allow administrators to retrieve complete report histories and prevent media from becoming detached from the incident it is intended to support. The project mentions SQLite in its database-management description. The paper does not assume a separate sensor database or time-series storage because the solution processes user-submitted reports rather than continuous sensor streams.

AI-Assisted Processing Services

AI services are treated as a processing component called within the software workflow. The service receives report information and returns a classification or analysis output that can be saved with the incident record. The interface should not represent this output as an unquestionable verdict. Where a classification is ambiguous, the administrator should be able to review the original description and evidence and make an informed decision. Model performance should be measured on a documented test set before numerical accuracy claims are made. The architecture described here keeps AI analysis linked to reported cases; it does not include autonomous event detection from installed cameras or physical sensors.

End-to-End Data Flow

The overall data flow begins with report preparation and ends with administrative follow-up. A user authenticates or accesses the permitted submission interface, enters the incident information, provides available location details, and attaches supporting evidence. The application validates required fields and supported file formats. If the device is offline, the report remains pending locally until connectivity returns. After synchronization, the server stores the record, makes the information available to the analysis process, and records the resulting classification. The alert workflow then makes relevant details available to the correct administrator. The administrator reviews the report, changes its status as work progresses, communicates with the user if additional information is required, and records the eventual outcome.

Each transition in this data flow should be visible and traceable. A report can be saved locally without yet being available to an administrator; an upload can be received without the report being fully reviewed; an AI result can be available without the case being resolved; and a status marked resolved should reflect an administrative update rather than simply the completion of classification. Distinguishing these states helps avoid misleading progress indicators and

supports accurate reporting about the work performed on each incident.

CORE SYSTEM MODULES

Authentication and Access Control

The authentication module manages registration, login, and secure access. Role-based permissions separate ordinary users from administrators and super administrators. Entity assignments restrict administrators to reports related to the organization they manage, supporting controlled access to report data.

Incident Reporting and Mobile Application

The incident reporting module is available through the web platform and the implemented mobile application. Users identify the affected entity, describe the fault or suspected vandalism, attach supporting evidence, and provide location details. Offline reporting allows a report to be prepared and saved without an active connection; pending reports are synchronized when connectivity returns.

Multimedia Upload Module

The multimedia upload module accepts user-selected or user-recorded photo and audio files from existing devices. It validates supported file types and sizes and associates each file with the relevant incident record. The uploaded evidence is available for authorized review and AI-assisted analysis. The system does not depend on dedicated cameras, external microphones, or continuous capture equipment.

AI Analysis Module

The AI analysis module reviews submitted descriptions and supported media to assist with classification of likely technical faults and suspected vandalism. It can support severity assessment and provides classification outputs as part of the incident record. These results guide administrators but do not replace human verification or independently detect events that have not been reported.

Emergency Response and Alert Generation

The Emergency Response Module and alert workflow help bring critical cases to the attention of responsible personnel. Alerts can present the incident type, description, severity information, and available location details. Administrators and relevant response personnel use this information to prioritize the report, communicate with the appropriate team, and record follow-up actions.

Location and Mapping

The mapping module displays reported incidents on digital maps. It uses the user's device location when permission is granted or a location selected manually on the map. The map helps responsible teams identify where the reported incident occurred; the solution does not use separately installed tracking devices or sensors.

Database, Administration, and Report Tracking

The database module stores account information, administrator assignments, reports, multimedia references, classification results, alerts, and system logs. The administration module allows authorized staff to review reports, update statuses, manage users, and monitor activity. Report tracking records stages such as Pending, Under Review, In Progress, and Resolved, allowing users and administrators to follow progress and add resolution notes.

Notifications, Analytics, and Reporting

The notification module confirms submission and informs administrators when incidents need attention. Dashboard and analytics functions summarize reports by entity, incident type, severity, and status. Search, filtering, report generation, feedback, and activity logging support day-to-day administration, accountability, and review of recurring cases.

User and Entity Management

User management supports the maintenance of account details, roles, and permissions. Administration is responsible for actions such as viewing users, updating eligible information, and enabling or disabling accounts according to policy. Entity management represents organizations or departments that receive and manage reports. Keeping entity information separate from individual user accounts allows staff membership and report routing to be managed without treating every user as part of the same organization. Correct entity assignment is important because it affects who receives the report and which administrator is permitted to review it.

Feedback, Activity Logs, Search, and Report Generation

The feedback and response function allows an administrator to request missing information or communicate the action taken on a report. This can help the reporting user understand whether more evidence is needed and whether the incident is under review. Audit and activity logs record key actions such as user access, report submission, status

updates, analysis results, and administrative changes. Search and filtering let authorized users retrieve records by incident type, date, location, status, severity, or entity. Report generation summarizes incident counts, response timelines where recorded, and the distribution of cases across categories. The usefulness of such summaries depends on accurate and consistent data entry and status updates.

Dashboard and Analytics Functions

Dashboards provide a role-specific view of the information most relevant to a user. A citizen needs a simple view of submitted reports and their status. An administrator may need incoming alerts, reports awaiting review, case severity, and lists filtered by entity. Summary charts can help an organization see which types of incidents are being reported and how many cases are pending or resolved. These summaries should be interpreted as statistics of reports entered into the system, not necessarily a complete count of all real-world infrastructure faults, because incidents that are never reported will not appear in the data.

Relationship Between Modules

The modules depend on one another. Authentication controls who can submit and review reports. Incident reporting creates the record to which evidence and location are attached. Multimedia upload supplies supporting files. AI analysis adds a classification result. Alert generation makes selected cases visible to administrators and response personnel. Administration and report tracking maintain the status history, while notifications and feedback communicate changes. Dashboards, search, and report generation use the records created by the other modules. This relationship means that an error in one component can affect another; for example, incorrect entity assignment can lead to a report being shown to the wrong administrator even if its description and attachments are otherwise valid.

REPORT PROCESSING AND DETECTION TECHNIQUES

The system workflow can be described through a sequence of software procedures. First, input validation checks the required fields and the format of submitted descriptions, multimedia files, and location details. Second, accepted reports are stored and linked to the appropriate user and entity. Third, AI-assisted classification analyzes the submitted information to help distinguish likely faults from suspected vandalism. Fourth, location processing associates coordinates or a map-selected point with the report. Fifth, status management records changes

made during administrative review. Finally, the alert workflow notifies responsible personnel when an incident requires attention.

These procedures use different types of information for different purposes. Validation protects data quality; AI-assisted classification supports the initial interpretation of the report; mapping supports location-based follow-up; status management preserves the case history; and alerts shorten the communication path between a report and the relevant office. Offline synchronization allows report preparation to continue when the device temporarily has no connection, while server-side processing takes place after synchronization.

Input Validation and Report Integrity

Validation begins by checking whether required fields are present and whether the submitted values have acceptable formats. A report should identify the nature of the incident, contain a useful description, and provide location details either from an authorized device location or a manually selected point. Attachments need to be checked against supported formats and size limits. Validation should reject malformed inputs with a clear message and should not create a misleading success message when a required step has failed. The report identifier should be maintained across upload and analysis operations so that every result and file can be connected to the correct incident.

AI-Assisted Classification Workflow

Once a valid report is available, the analysis component processes supported report information and produces an output intended to help distinguish a likely technical fault from suspected vandalism. A high-level workflow is: receive an accepted report; select the available fields and evidence supported by the model; perform preprocessing required by the analysis service; obtain the classification output; associate the output with the report record; and present the result to an authorized reviewer. Severity information can be used as an additional aid to prioritization where supported. The documented project does not provide enough numerical evidence to specify a confidence threshold or claim a particular rate of correct classification, so such model parameters are not stated here.

Location Processing and Map Selection

Location processing has two valid input routes. If the user grants permission and the device supplies a location, the application can associate the available coordinates with the report. If this is not possible or

the location is inaccurate, the user can select the incident point manually on a digital map. The system should make it clear which location has been attached to the record and allow the user to review it before submission. Accurate coordinates do not themselves verify that the described event took place; they make it easier for the responsible organization to identify the site that the reporter has indicated.

Offline Storage and Synchronization

Offline reporting separates preparing a report from delivering it to the central service. While disconnected, the application can retain the draft report and its pending attachments on the user's device. When a connection becomes available, the application attempts to synchronize the pending content and reports whether synchronization succeeded. The application should avoid creating duplicate incidents if a user retries a submission after an uncertain network response. A robust design therefore distinguishes a locally saved draft, a synchronization-pending report, a successfully submitted report, and a report that has been accepted for review. Online AI analysis and server-side alerting can proceed only when the corresponding information has reached the backend.

Emergency Prioritization and Alerting

The alert workflow converts an incident record into an item that can receive timely administrative attention. The record should carry the details needed for a recipient to understand what was reported: incident category, description, available severity information, location, and a route back to the complete report. The alert should be associated with the proper entity and should avoid exposing private report details to unauthorized users. Once the recipient has reviewed the case, the report status and follow-up notes help establish what action has been taken. Alerts support coordination, but the software does not itself guarantee that a field team has been dispatched or that an incident has been resolved.

Report Status and Audit Trail

Report tracking provides a structured history of progress. The project describes status values such as Pending, Under Review, In Progress, and Resolved. These statuses communicate the position of the report in the administrative workflow. The transition should be made by an authorized user or a documented application process and should be recorded to support traceability. The audit trail can show who performed a change and when it was recorded, where such fields are implemented. A clear status model helps users

understand the difference between a submitted report and a completed response and gives administrators a way to identify cases that remain open.

IMPLEMENTATION AND PHASE II FEATURES

The project report describes a software implementation with web interfaces, backend/API services, database storage, multimedia upload handling, and AI-assisted analysis. The submitted code sample includes an Express-based backend with routes for authentication, reports, vandalism records, uploads, analysis, and statistics. Because the wider project description also mentions other technology options, this paper describes the implementation at the level of its verified application functions rather than claiming a single technology stack beyond what the included code demonstrates.

The Phase II implementation incorporates five main features: a mobile application; photo and audio uploads; AI-assisted incident classification; offline report preparation and synchronization; and an Emergency Response Module with alert support. These functions are incorporated into the existing reporting and administrative workflow. The design remains software-only and uses devices already available to users rather than installed cameras, dedicated sensors, or IoT hardware.

Mobile Application and Cross-Device Use

The mobile application extends access beyond a desktop browser and supports report preparation in contexts where a user observes a problem outside an office. Mobile access should preserve the same core reporting logic as the web interface: the report has a description, an incident category or selected entity, optional media, location information, and a recorded status. Consistency between interfaces is important because users may submit one report using a phone and later check it through another supported interface. The available project description confirms mobile application access as an implemented Phase II feature but does not document the full platform-specific technology, supported operating-system versions, or app-store distribution details; those are therefore not asserted in this paper.

Photo and Audio Evidence Handling

Photo and audio uploads allow the user to provide more than a written account. The upload interaction should indicate the accepted file types and size limits, show the attachment before submission where supported, and return a clear message if a file is rejected. The server must associate each attachment with its incident record and make it available only

through authorized application functions. Since the platform uses existing user devices, file selection or user-initiated recording is part of the reporting process; continuous capture from dedicated equipment is outside the system scope. Evidence handling should also account for file privacy, because photographs and recordings may inadvertently contain people, property details, or other sensitive information.

Software-Only Deployment Model

The system is intended to be deployed as software services: the web interface and mobile application communicate with backend functions, those functions store and retrieve records from the database, and supported analysis services process submitted information. Implementation tasks therefore focus on application configuration, hosting, database management, permissions, API availability, file storage, backup, and monitoring. The solution does not require the installation or maintenance of a field hardware network. This scope can reduce the need for separate equipment procurement, but the system still requires suitable hosting, operational support, connectivity for online processing, and procedures for account administration and data backup.

Security, Privacy, and Responsible Use

Reports may contain location information, names, photographs, audio, and allegations of suspected vandalism. Access to this information should therefore be limited to users who need it to perform their duties. Passwords should be stored securely, permissions should be enforced by the backend rather than only by hiding interface elements, and input validation should be applied to both form fields and file uploads. Sensitive details should not be placed unnecessarily in public alerts. Administrators should treat AI classifications as decision-support information and verify a case before taking consequential action. These controls are especially important when the system handles reports about suspected deliberate misconduct rather than a straightforward equipment fault.

Privacy-aware reporting also requires transparency to users. The application should explain why location permission is requested, allow manual location selection where appropriate, and make it clear when a report or attachment has been submitted. Users should not be encouraged to upload irrelevant personal material. A defined retention and access policy would help participating organizations determine how long report evidence and audit records should be kept and which personnel may review

them. The project includes authentication, role-based access, and activity logging as system functions; this discussion identifies how those functions contribute to safer operation without claiming that a formal privacy audit has been completed.

TESTING AND EVALUATION

The project uses Black Box Testing, which checks the system through its observable inputs and outputs rather than examining internal code structure. The documented test plan covers functional behavior, security, performance, usability, and recovery. Functional testing is marked as tested and includes mobile reporting, photo and audio uploads, AI-assisted classification, offline reporting, and emergency-response alerts. Security testing is marked as tested for authentication and access control. Performance testing is marked as tested for response time and multiple-user handling. Recovery testing is marked as tested for handling failed submissions or system errors. Usability testing is recorded as not fully tested.

The test plan indicates that the main functional workflow has been evaluated, but the project report does not provide a detailed numerical dataset, AI classification accuracy, precision, recall, or measured average response time. Accordingly, this paper does not claim quantitative performance improvements that are not documented. A fuller evaluation would report the size and composition of the test dataset, class-wise model performance, false-positive and false-negative rates, response-time measurements, and usability feedback from intended users.

Functional Test Areas

Functional tests examine whether the application supports the expected tasks. Registration and authentication tests check whether a user can create or access an account and whether invalid credentials are rejected. Incident-submission tests check whether the form captures the report details and whether incomplete required fields produce a useful validation message. Multimedia tests verify that supported photo and audio files can be attached and that unsupported types or oversized files are handled appropriately. Location tests check whether a permitted device location or a manually selected point is associated with the intended report. These tests help confirm that a complete report can move through the application without losing its identifying information.

The Phase II feature tests should also distinguish online from offline conditions. For mobile reporting,

a user should be able to complete the same core reporting actions supported by the system. For offline reporting, the test should confirm that the report is saved locally, visibly marked as pending, and synchronized when connectivity returns. It should also check behavior when synchronization fails or is retried. AI-assisted classification tests should verify that a supported submitted report returns a stored output and that the result is visible in the intended administrative workflow. Emergency-response tests should confirm that qualifying reports create or display the expected alert details and that the receiving administrator can open the associated case.

Security, Performance, Recovery, and Usability

Security testing examines authentication, authorization, and the protection of incident data. Examples include checking whether an ordinary user can access restricted administration functions and whether an administrator sees only the reports permitted by their entity assignment. Upload endpoints should reject invalid files and handle malformed requests safely. Performance testing examines response and processing behavior, including how the application handles multiple users or repeated submissions. The project test plan marks performance testing as tested but does not supply the actual response-time values or number of concurrent users, so these results cannot be quantified in this paper.

Recovery testing addresses interruptions and failures. It should examine what happens after a failed submission, an interrupted upload, a backend error, or a temporary network loss. The expected outcome is not necessarily that every action succeeds immediately; it is that the application provides understandable feedback and preserves enough state for an appropriate retry or recovery. Usability testing is a separate concern because a technically correct form may still be confusing to its intended users. In the project report, usability is recorded as not fully tested. A complete usability assessment would ask representative users to submit a report, attach evidence, choose a location, find the status of a previous report, and interpret an alert, while recording completion issues and feedback.

Interpreting the Test Evidence

A test plan marked as tested is evidence that a test activity was reported as performed, but it does not provide all the information required to assess how rigorous or comprehensive the test was. Stronger reporting would include the test environment, input conditions, expected result, observed result, pass or

fail status, defect encountered, and any retest performed after a correction. For an AI classification component, this should be complemented by a labelled evaluation set and metrics that show how well each class is recognized. Without that detail, the responsible conclusion is that the report documents testing activity but does not establish a numerical level of model accuracy or improvement in public-service response time.

RESULTS AND DISCUSSION

The documented implementation brings reporting, evidence upload, location information, AI-assisted classification, alert generation, and administrative tracking into a common software workflow. The mobile application and offline synchronization address access limitations that can arise when users are away from reliable connectivity. Photo and audio evidence provide additional context beyond a short written description, while location information helps responsible personnel identify the affected site. The Emergency Response Module and alerts make urgent reports easier to prioritize and follow up.

The design also establishes a clearer record of how a report moves from submission to resolution. Role-based access helps separate responsibilities across entities, and the central record supports search, summaries, and activity tracking. These capabilities align with the wider need for digital public service delivery and reliable institutional records identified in literature on e-government and records management [1-5]. AI-related literature supports the use of learned models for interpreting submitted data [6-8], but the practical value of the classification module depends on the quality and representativeness of training and evaluation data as well as administrator verification.

There are limitations in the evidence currently documented. Usability testing is not fully completed, and detailed quantitative measurements for classification performance and response-time improvement are not provided. The platform also depends on users reporting incidents and providing usable information; it is not an autonomous surveillance system and does not continuously observe infrastructure. Further evaluation should prioritize real-world testing with relevant institutions, feedback from citizens and administrators, and transparent reporting of classification metrics and operational response times.

Illustrative Reporting Scenarios

A technical-fault scenario illustrates the basic workflow. A user notices a water-pipe leak or an

electricity-related fault and opens the reporting interface. The user selects the responsible entity, writes a description of the observable problem, attaches a photo or audio explanation if available, and supplies location information. The system validates the information and records the report. AI-assisted classification can help indicate that the report appears to describe a technical fault. An administrator reviews the submitted evidence, updates the status, and coordinates the appropriate follow-up. This scenario illustrates the intended software sequence; it is not presented as a measured real-world case from a participating utility.

A suspected-vandalism scenario follows the same initial steps but involves a report of apparent deliberate damage, tampering, or theft affecting public infrastructure. The submitted description and evidence are analyzed to support classification, while the Emergency Response Module helps make the incident visible to responsible personnel when it needs prompt attention. The administrator must review the information before deciding the appropriate next action. This separation is important because the system processes a suspicion submitted by a user; its classification does not establish intent, responsibility, or legal proof.

An offline scenario demonstrates why local preparation and synchronization are distinct states. A user at a location with no reliable connection prepares the description and selects available evidence. The application stores the pending report locally and indicates that it has not yet been received by the central system. After connectivity returns, the application synchronizes the report. Only after successful synchronization can the central backend store it, run any server-side analysis, and send the associated alert. Making those states visible reduces the risk that a user assumes an urgent incident has already been received when it is still waiting on the device.

Operational Implications

The combined workflow can improve the organization of incident information by linking evidence, location, classification, alerts, and administrative status to one report record. It can also support accountability by keeping activity logs and making the current report status visible to authorized users. These functions are useful for organizations that need a consistent place to receive and manage information from citizens and staff. Their practical effect depends on operational adoption: an institution must designate personnel to review incoming cases, keep entity assignments accurate, maintain the

application, and update report status when action is taken. Software alone cannot guarantee that a reported fault will be repaired or that an emergency response will occur.

Limitations of the Current Evidence

The project documentation describes the system's architecture, modules, implementation functions, and test areas, but it does not provide detailed experimental tables for the AI component. The paper therefore cannot independently verify classification accuracy, precision, recall, false-alarm rate, or performance by incident category. It also does not include a measured comparison of response times before and after deployment, a quantified cost-benefit analysis, or a completed usability study. In addition, real operational outcomes would depend on network availability, completeness of user-submitted reports, quality of evidence, administrator workload, and the processes followed by each participating entity. These limitations do not negate the functional scope of the application; they define what can and cannot be concluded from the evidence currently recorded.

The software-only scope is another important boundary. The platform does not continuously observe infrastructure and cannot automatically identify an event that no user reports. It depends on the availability of an existing device for submitting a report and uses device location only when permission and the relevant service are available. Manual map selection provides an alternative, but its accuracy depends on the user's selection. The system can structure and analyze the report that it receives; it cannot replace physical verification, engineering assessment, police investigation, or other professional actions required to resolve an incident.

DATA LIFECYCLE AND INFORMATION GOVERNANCE

A useful way to understand the system is to follow an incident record across its lifecycle. The lifecycle begins when a user starts a report and enters the details of the observed problem. Before the report is submitted, the application validates the available fields and handles the evidence attachments. For an offline report, the initial record remains on the device and is marked as awaiting synchronization. After a successful transfer, the central application stores the report and makes it available for analysis and administrative review. This sequence requires clear state management so that locally stored data is not confused with a report that has been received by the institution.

Once the record has reached the central system, it may be updated as the incident moves through review, response, and resolution. The report should keep a consistent identifier so that classification outputs, alerts, evidence files, user communication, and status updates remain linked. If an administrator requests more information, the feedback should refer to the same report rather than create an unrelated duplicate. When the report is resolved, the recorded status and resolution note provide a basis for later searching and summary reporting. This lifecycle makes the system more useful for accountability because it preserves the progression of an incident instead of storing only the initial complaint.

Information governance concerns who can access records, how changes are recorded, and how the organization uses stored information. The project's role-based access model and entity assignments can help restrict report access to appropriate users. Audit logs help record important activities, including report submission and administrative changes. These controls should operate consistently across the web and mobile interfaces and the backend APIs. A permission check performed only in the user interface would be insufficient because a request could still be made directly to a backend endpoint. For that reason, access rules should be applied by the application logic whenever protected records are retrieved or changed.

Data quality also affects analysis and management decisions. If incident categories are inconsistent, locations are missing, or reports are left in outdated statuses, dashboard summaries may be misleading. The application can reduce such problems by validating required information, using consistent status values, associating files with their parent reports, and asking administrators to record important changes in the case history. However, validation cannot guarantee that every user description is accurate or that every manually selected point is exact. Reports should be treated as statements requiring review, especially when the incident involves suspected vandalism or other consequential allegations.

ORGANIZATIONAL WORKFLOW AND ADOPTION

The system is intended to support the existing responsibilities of organizations rather than impose one identical response process on every institution. An electricity-related report may need technical inspection, while a report of suspected deliberate damage may require security review as well as an operational response. The application can make the

report information and history available, but each participating organization remains responsible for defining who reviews new cases, how urgent cases are escalated, who can change statuses, and what evidence is needed before a case is closed. Clear responsibility is essential because an alert that has no assigned recipient or review procedure may not lead to action.

A practical operating procedure should define ownership at each stage. Users need understandable guidance on what information to provide and how to report an urgent event. Administrators need rules for validating a report, checking AI-assisted results, selecting a priority, communicating with the reporting user, and assigning follow-up. Response personnel need access to the location and supporting evidence appropriate to their duties. Supervisors may use summaries and activity records to identify delayed cases or incomplete records. These are organizational process requirements that accompany the software and cannot be solved only by adding technical features.

User adoption depends on whether the system is understandable and dependable. A user may abandon a submission if the form is too difficult, the upload fails without explanation, or the application does not make clear whether an offline report has synchronized. Administrators may revert to informal communication if the dashboard does not reflect their actual workflow or if notifications do not contain enough context. Training and support should therefore explain the reporting steps, the meaning of report statuses, the limits of AI classifications, and the process for handling failed synchronization. The project implementation description includes user training, configuration, testing, security checks, and performance monitoring as activities associated with making the software available to its intended users.

EVALUATION FRAMEWORK AND EVIDENCE REQUIREMENTS

A clear evaluation framework separates functionality from effectiveness. Functional evaluation asks whether a feature is available and behaves as expected: a user can submit a report, attach supported evidence, obtain a location, save an offline draft, synchronize it later, view a status, and trigger the appropriate administrative workflow. Security evaluation asks whether access restrictions are enforced and whether protected records remain unavailable to unauthorized users. Reliability evaluation examines failure recovery, duplicate prevention, and whether report information remains

consistent after interrupted operations. Usability evaluation considers whether the intended users can complete common tasks and interpret system feedback without avoidable confusion.

The AI component requires its own evidence. A meaningful test should use a documented set of reports for which the expected category has been established independently. The results should identify the number of technical-fault and suspected-vandalism examples, explain how the data were divided for training and testing where applicable, and report the correct and incorrect classifications for each class. Accuracy alone may hide poor performance when one class is more common than the other, so precision, recall, and the F1-score can help describe the balance between missed incidents and incorrect alerts. These metrics are evaluation requirements rather than results reported by the current project document.

Operational effectiveness should be assessed separately from model performance. Relevant measures may include the time between report preparation and confirmed receipt, the time from receipt to first administrative review, the proportion of records with usable location information, the share of reports that reach a documented status, and user feedback about ease of reporting. A comparison with the prior reporting process would need a defined baseline, the same observation period where feasible, and clear definitions for each measure. Cost-effectiveness would require documented implementation and operating costs as well as a defensible method for estimating benefits. Since those numerical data are not supplied in the project report, this paper records them as items for evaluation rather than presenting unsupported measured gains.

CONCLUSION

This paper presented the AI-Driven Multimedia Fault Reporting and Anti-Vandalism System for Malawi's public infrastructure as a software-based approach to reporting and managing faults and suspected vandalism. The system combines web and mobile access, photo and audio uploads, location mapping, AI-assisted classification, offline reporting and synchronization, report tracking, notifications, and emergency response support. Together, these functions provide a centralized workflow for submitting incident evidence, reviewing reports, prioritizing cases, and recording follow-up.

The project documents functional, security, performance, and recovery testing as tested, while usability testing remains incomplete. Since the report

does not include numerical AI accuracy or response-time measurements, those outcomes should not be assumed; additional evaluation is needed to quantify effectiveness and usability in real operating conditions. The project nevertheless provides a practical software foundation for strengthening communication between citizens and responsible institutions, without requiring dedicated cameras, sensors, or other IoT equipment.

REFERENCES

- [1] Mwafulirwa, W. (2021). Computerised and not corrupt government? Assessing the effectiveness of e-governance in Malawi public service. Available at: <https://www.researchgate.net/publication/354554647>
- [2] Makoza, F. (2013). The level of e-government implementation: Case of Malawi. Available at: <https://www.researchgate.net/publication/298704854>
- [3] Malanga, D. (2016). E-Government adoption, implementation, benefits and challenges: The Malawian experience. Available at: <https://www.researchgate.net/publication/312527820>
- [4] Msosa, M., Hoko, Z., & Maliano, S. (2023). Records management system at Northern Region Water Board, Malawi. <https://doi.org/10.1080/13614576.2023.2175709>
- [5] Mtingwi, F., & Van Belle, J. P. (2012). E-government and m-government readiness in Malawi. Available at: <https://www.researchgate.net/publication/312488450>
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Cambridge, MA: MIT Press. Available at: <https://www.deeplearningbook.org>
- [7] Chollet, F. (2021). Deep Learning with Python (2nd ed.). Manning Publications. Available at: <https://www.manning.com/books/deep-learning-with-python-second-edition>
- [8] Rosebrock, A. (2019). Deep Learning for Computer Vision with Python. PyImageSearch. Available at: <https://pyimagesearch.com>
- [9] International Association of Chiefs of Police. (2012). NIBRS incident reporting framework. Available at: <https://nij.ojp.gov/library/publications/nibrs-technical-specifications>
- [10] International Telecommunication Union. (2021). Digital Transformation for Sustainable Development. Geneva: ITU. Available at: <https://www.itu.int>