JUVOTEK JURNAL
Jurnal Vokasi Teknik | *J. Vok. Tek.*
https://mentech.id/jurnal/index.php/juvotek | e-ISSN 3025-1125

Original Research Article

An Android-Based Real-Time Asset Maintenance Complaint Management System for Polytechnic Facilities

Alfat Fauzang Haniar*[1], Bambang Cahyono[1], Yusni Nyura[1]

[1] *Teknologi Informasi, Teknik Informatika Multimedia, Politeknik Negeri Samarinda, Samarinda, Indonesia.*

Received: January 21, 2026| **Accpeted:** March 20, 2026 | **Published:** April 01, 2026

ABSTRACT

Effective asset maintenance in higher-education institutions requires timely reporting, monitoring, and documentation of facility-related problems. Conventional complaint procedures may limit information traceability and coordination among users, administrators, and institutional decision-makers. This study aimed to develop and empirically evaluate an Android-based asset-maintenance complaint management system for polytechnic facilities. A design and development research approach was implemented using the Waterfall model, comprising requirements analysis, system design, implementation, and functional testing. The application was developed using Kotlin and Android Studio with Firebase Realtime Database for data management. Three operational roles were supported: End User, Administrator, and Managerial User. Functional suitability was evaluated through black-box testing of eight principal functional modules. Perceived usability was assessed using the System Usability Scale (SUS) among 30 participants comprising 22 End Users, 5 Administrators, and 3 Managerial Users. Descriptive statistics included mean, standard deviation, median, range, and 95% confidence interval. All eight functional modules passed the predefined test criteria, yielding a functional pass rate of 100% (8/8). The overall SUS score was 79.42 (SD = 4.14), with a median of 80.00, a range of 67.50–85.00, and a 95% confidence interval of 77.94–80.90. Mean SUS scores were 78.86 for End Users, 80.00 for Administrators, and 82.50 for Managerial Users. The developed application demonstrated functional feasibility and favorable perceived usability for asset-maintenance complaint management in a polytechnic environment. The system provides a structured mobile-based foundation for information traceability and future data-driven maintenance management.

Keywords:
asset maintenance; complaint management; Android application; mobile information system; system usability; polytechnic facilities.

Article Info
Check for updates

EDITED BY
Dr. Ir. Arman, MM.

Magister Manajemen, Sekolah Tinggi Ilmu Ekonomi Bangkinang.

COPYRIGHT©2026 (Author)
Alfat Fauzang Haniar
Bambang Cahyono
Yusni Nyura

***CORRESPONDENCE**
Alfat Fauzang Haniar
Alfatfauzan25@gmail.com

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

To cite this article (APA Style 7):
Fauzang Haniar, A., Cahyono, B., & Nyura , Y. (2025). An Android-Based Real-Time Asset Maintenance Complaint Management System for Polytechnic Facilities. *Jurnal Vokasi Teknik (JuVoTek)*, 4(01), 01-12. https://doi.org/10.67922/juvotek.v4i01.01

INTRODUCTION

Background and Engineering Problem

Educational institutions depend on buildings, classrooms, laboratories, information-technology facilities, utilities, and other physical assets to support academic and administrative activities (Alashari et al., 2023; Meschini et al., 2022, p. 4). The operational availability of these assets is closely associated with the effectiveness of maintenance processes (Beauregard & Ayer, 2018; Ebekozien et al., 2023, p. 186). When defects are not reported, documented, prioritized, and followed up systematically, minor failures can develop into operational disruptions and increase the difficulty of maintenance coordination.

A major challenge in institutional maintenance management is the transition from paper-based or fragmented complaint procedures toward integrated digital workflows. Conventional complaint mechanisms may require users to submit forms or letters to administrative units, followed by verification, authorization, assignment, and repair (Ismail, 2021; Mendes et al., 2026, p. 6319). Such procedures can create delays in information transfer and make it difficult to establish a complete history of complaints and maintenance actions.

Evidence from polytechnic institutions indicates that conventional e-complaint and maintenance processes may suffer from inadequate integration between operational and maintenance activities, inconsistent information management, and limited support for managerial decision-making (Ismail, 2021; Ju et al., 2025, p. 126004; Mendes et al., 2026, p. 6330). Similar problems have been reported in higher-education environments where manual facility-damage reporting creates difficulties in tracking complaints and coordinating repairs (Binalhaj et al., 2021, p. 1655; Sari et al., 2021). These conditions indicate the need for information systems capable of connecting users, maintenance administrators, and institutional decision-makers within a single workflow.

At Politeknik Negeri Samarinda, the complaint process addressed in this study was previously associated with the submission of repair requests requiring administrative disposition from the Vice Director. The original manuscript identifies difficulty in tracking and following up complaints as an operational problem. The absence of an integrated digital mechanism also limits the ability of management to monitor complaint histories and summarize maintenance information.

Mobile technology provides an opportunity to address these limitations because users can submit information directly from the location where an asset failure occurs. Recent studies have demonstrated the growing role of mobile maintenance management systems in supporting real-time reporting, work-order management, and maintenance coordination (Levi-Bliech & Rachevski, 2026; Simard et al., 2024). A recent Indonesian study similarly demonstrated the feasibility of mobile and web-based infrastructure-damage reporting with real-time notification mechanisms (Dani & Rismayana, 2026).

State of the Art

The development of digital maintenance systems has progressed from basic complaint recording toward integrated maintenance-management platforms. Earlier work in higher-education facilities demonstrated the value of computerized reporting for reducing dependence on physical complaint forms and improving traceability of facility problems (Sari et al., 2021). In the context of Malaysian polytechnics, Ismail (2021) emphasized that e-complaint systems should not merely capture complaints but should support broader maintenance-management processes and decision-making.

More recent research has extended this concept toward mobile Computerized Maintenance Management Systems (CMMS). Simard et al. (2024), for example, developed and evaluated a CMMS prototype for mobile mining equipment and emphasized usability evaluation as an important component of maintenance-system development.

The role of historical maintenance data is also becoming increasingly important. D'Orazio et al. (2023) demonstrated that maintenance requests stored in a CMMS can be processed using machine-learning techniques to identify request priorities and predict technical-staff assignments. Their study used more than 14,500 maintenance requests and reported accuracies of 93.3% for priority identification and 96.7% for staff assignment.

Recent evidence from decentralized healthcare facilities further indicates that mobile maintenance systems can influence operational performance. Levi-Bliech & Rachevski (2026) reported statistically significant reductions in maintenance time following implementation of a real-time mobile maintenance-management system, particularly for several categories of repairs.

These studies indicate a progression from simple electronic complaint registration toward integrated, mobile, data-driven maintenance management. However, the complexity of advanced CMMS platforms is not always appropriate for educational institutions requiring a focused solution for reporting, verification, monitoring, and historical documentation of facility damage.

Research Gap

Despite the development of digital maintenance-management systems, three issues remain relevant to the present research. First, previous research has frequently addressed large-scale CMMS, predictive maintenance, or specialized industrial environments. These systems may include advanced functions such as work-order optimization, predictive analytics, QR/RFID integration, or preventive-maintenance scheduling. Second, existing e-complaint research in educational institutions has emphasized the need for integrated maintenance processes, but relatively fewer studies have presented a lightweight Android-based architecture that explicitly connects users, administrators, and institutional authorization roles within a polytechnic facility-maintenance workflow. Third, the original system developed at Politeknik Negeri Samarinda had demonstrated functional feasibility, but its presentation did not sufficiently distinguish between functional validation and broader claims concerning efficiency or effectiveness. Software quality evaluation should ideally consider multiple dimensions. ISO/IEC 25010:2023 provides a contemporary product-quality model that can be used to specify and evaluate ICT and software-product quality.

Therefore, the present study focuses on establishing a functional mobile foundation for asset-maintenance complaint management while explicitly limiting its empirical claim to the functionality demonstrated by black-box testing.

Novelty and Contribution

The main contributions of this study are: 1) A role-based Android maintenance-complaint architecture integrating users, administrators, and vice-director-level authorization within a polytechnic facility-management workflow; 2) A real-time complaint-management mechanism using Kotlin, Android Studio, and Firebase Realtime Database to support complaint submission, storage, monitoring, and historical retrieval; 3) A functionally validated prototype covering eight principal application functions, with all tested functions successfully executing according to the specified requirements; 4) A structured foundation for future maintenance analytics, because complaint histories and statistical information can subsequently support prioritization, response-time analysis, and predictive maintenance.

Research Objective

This study aims to design, develop, and functionally validate an Android-based asset-maintenance complaint management application for Politeknik Negeri Samarinda. Specifically, the study seeks to: 1) analyze the requirements of the existing asset-damage complaint process; 2) design an integrated complaint-management architecture using UML; 3) implement the system as an Android application using Kotlin and Firebase Realtime Database; 4) and evaluate the functional operation of the developed application using black-box testing.

THEORY AND/OR CALCULATION

Digital Asset Maintenance and Complaint Management

Asset maintenance complaint management can be conceptualized as a digital workflow connecting the occurrence of an asset problem with its reporting, verification, administrative processing, status monitoring, and historical documentation. The basic workflow implemented in this study is:

Asset problem → User complaint → Administrative processing → Authorization/verification → Complaint status → Historical record → Statistical information

Such a workflow transforms a complaint from an isolated communication event into a structured information object. This approach is consistent with the broader development of maintenance-management systems in which maintenance requests become part of an organizational information base that can support operational decisions (D'Orazio et al., 2023; Ismail, 2021).

Software Development Model

The study uses the Waterfall software-development model. The original implementation consists of five principal stages: data collection and analysis, system design, implementation, testing, and maintenance.

The use of a sequential development model is appropriate when system requirements can be defined before implementation and when the development process requires explicit documentation of each stage. Contemporary research also indicates that structured and Agile approaches should be selected according to project characteristics rather than treating one methodology as universally superior (Mishra & Alzoubi, 2023).

Functional Validation Metric

Because the available experimental evidence is limited to black-box functional testing, the principal quantitative metric is the functional pass rate: $FPR = \frac{Np}{Nt} \times 100\%$

where: (FPR) = functional pass rate; (N_p) = number of functions that successfully passed testing; and (N_t) = total number of functions tested.

For the developed application: $FPR = \frac{8}{8} \times 100\% = 100\%$

MATERIALS AND METHODS

Research Design

This study employed a design and development research approach to design, implement, and functionally evaluate an Android-based asset-maintenance complaint management system for Polytechnic facilities. The research consisted of two complementary components: (1) software development and functional verification and (2) empirical usability evaluation involving intended users of the application.

The software-development process followed the Waterfall model, comprising requirements analysis, system design, implementation, functional testing, and maintenance planning. Functional suitability was evaluated using black-box testing, while perceived usability was assessed using the System Usability Scale (SUS). The combination of functional verification and standardized usability assessment was intended to provide complementary evidence concerning the operational feasibility and user-perceived usability of the developed system. The overall research workflow is presented in Figure 1.

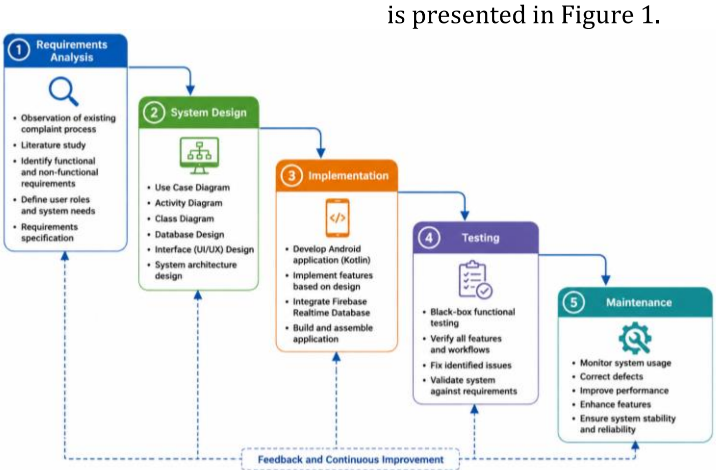

Figure 1. Research and software-development workflow based on the Waterfall model.

Research Setting and System Context

The study was conducted in the context of asset-maintenance management at Politeknik Negeri Samarinda, Samarinda, Indonesia. The research object was the process of reporting, managing, monitoring, and documenting complaints concerning damaged institutional assets and facilities. The developed application was designed to support three principal user roles: 1) End User, who submits and monitors maintenance complaints; 2) Administrator, who manages users and complaint information; and 3) Managerial User (Vice Director), who performs complaint confirmation and accesses statistical information.

The proposed system transforms the existing complaint process into a structured digital workflow connecting complaint submission, administrative processing, confirmation, status monitoring, historical documentation, and statistical reporting.

System Development Methodology

Requirements were identified through direct observation and literature study.

Observation was conducted to understand the existing complaint process, user interactions, information requirements, and administrative procedures. Literature study was used to identify relevant concepts concerning digital complaint management, mobile applications, software development, and maintenance-management systems. The principal functional requirements identified were: a) secure user authentication; b) role-based access; c) asset-damage complaint submission; d) building and room identification; e) complaint-description entry; f) image upload; g) supporting-document upload; h) complaint-status management; i) complaint-history access; j) user management; k) statistical reporting; and l) logout.

System Design

The system was designed using Unified Modeling Language (UML). Three principal diagrams were developed to represent the functional, behavioral, and structural characteristics of the application.

Use Case Model

The Use Case Diagram represents the interactions between users, administrators, vice directors, and the system.

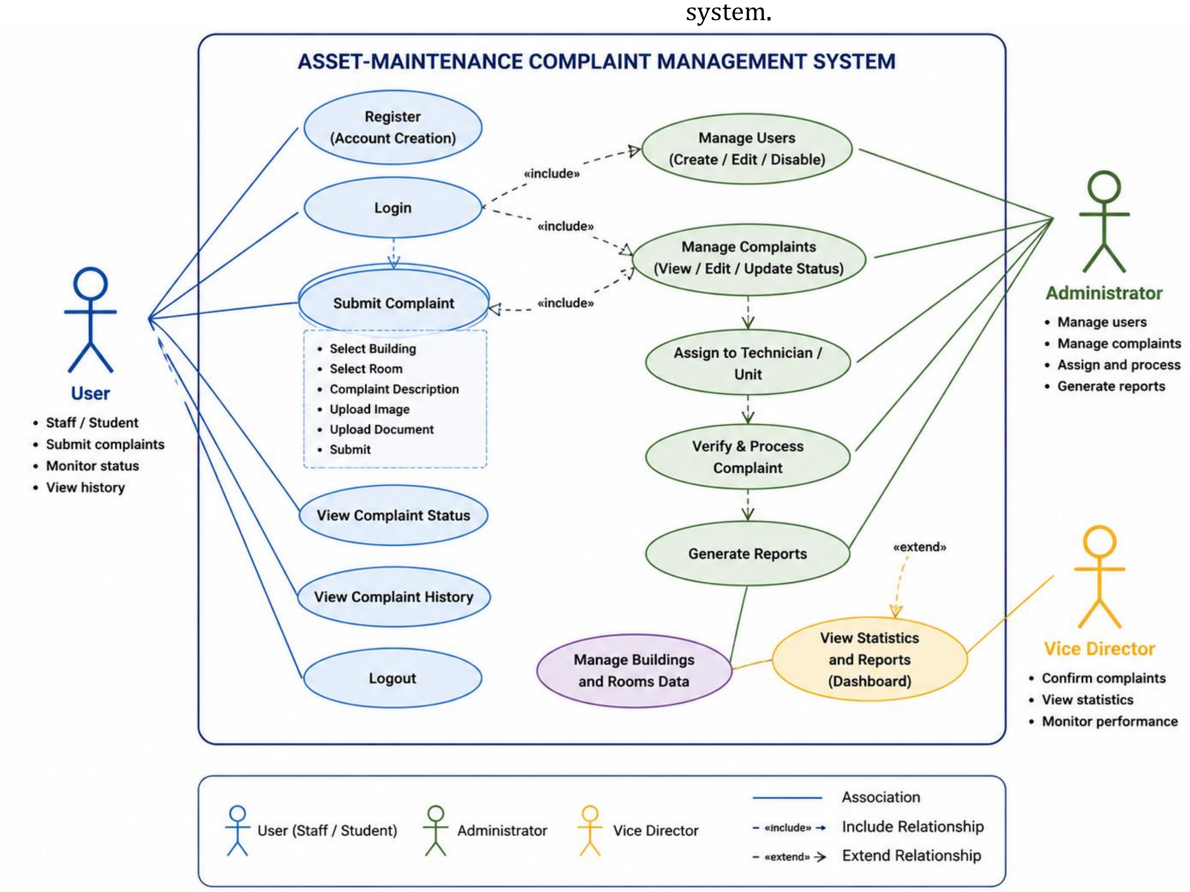

Figure 2. Use Case diagram of the proposed asset-maintenance complaint management system.

The model defines the functional boundary of the system and differentiates the responsibilities associated with each user role.

Activity Model

The Activity Diagram represents the operational workflow from authentication through complaint submission, administrative processing, confirmation, and statistical monitoring.

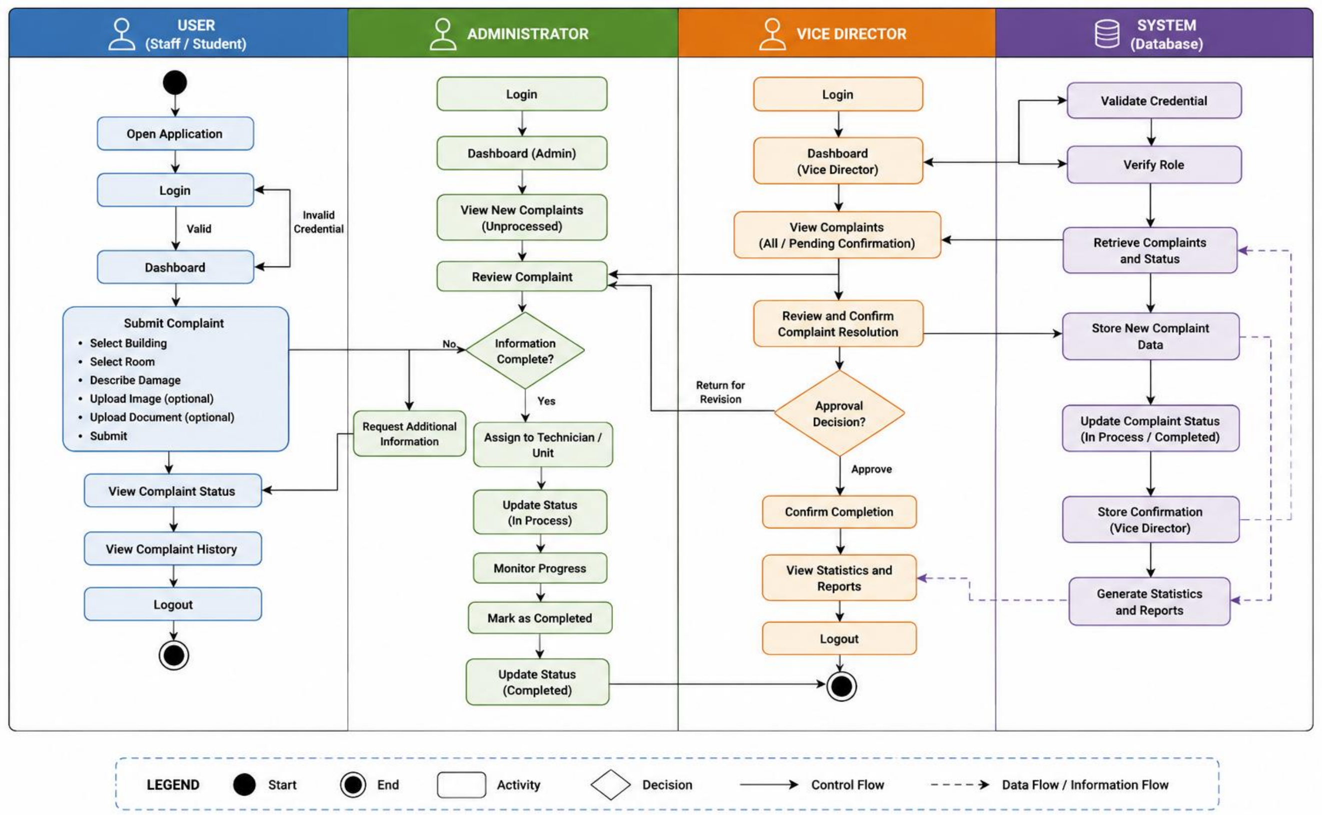

Figure 3. Activity diagram of the asset-maintenance complaint management workflow.

The activity model demonstrates that complaint management involves different decision and processing paths depending on the user's role.

Class Model

The structural architecture was represented through the Class Diagram. The principal entities include User, Administrator, Vice Director, Complaint, Building, Room, and Complaint Statistics.

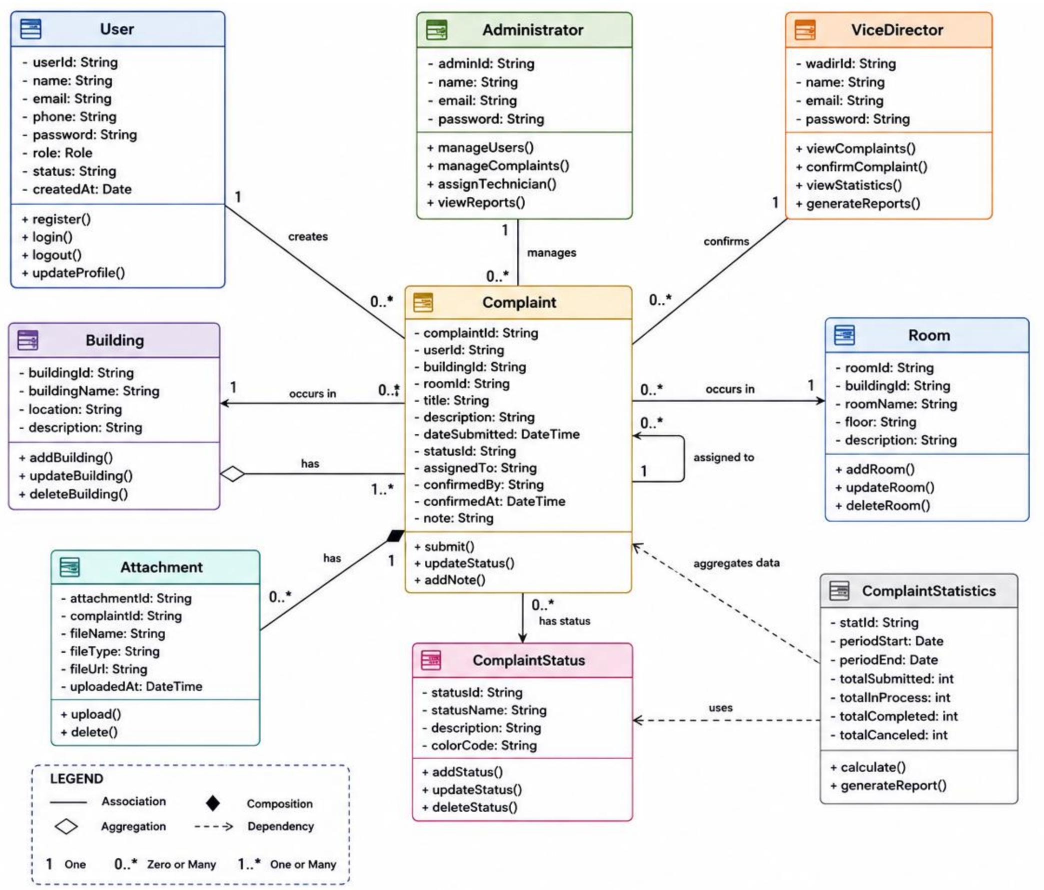

Figure 4. Class diagram of the proposed application.

The class structure provides the conceptual basis for managing user identities, complaint information, facility locations, and statistical records.

Application Implementation

The application was implemented as an Android application using **Kotlin** and **Android Studio**. Firebase Realtime Database was used as the primary data-management platform.

The principal application interfaces included: authentication; user dashboard; administrator dashboard; complaint management; complaint history; statistical information; and logout.

The complaint interface allows users to select the affected building and room, enter the complaint description, upload supporting images, attach supporting documents, and submit the complaint.

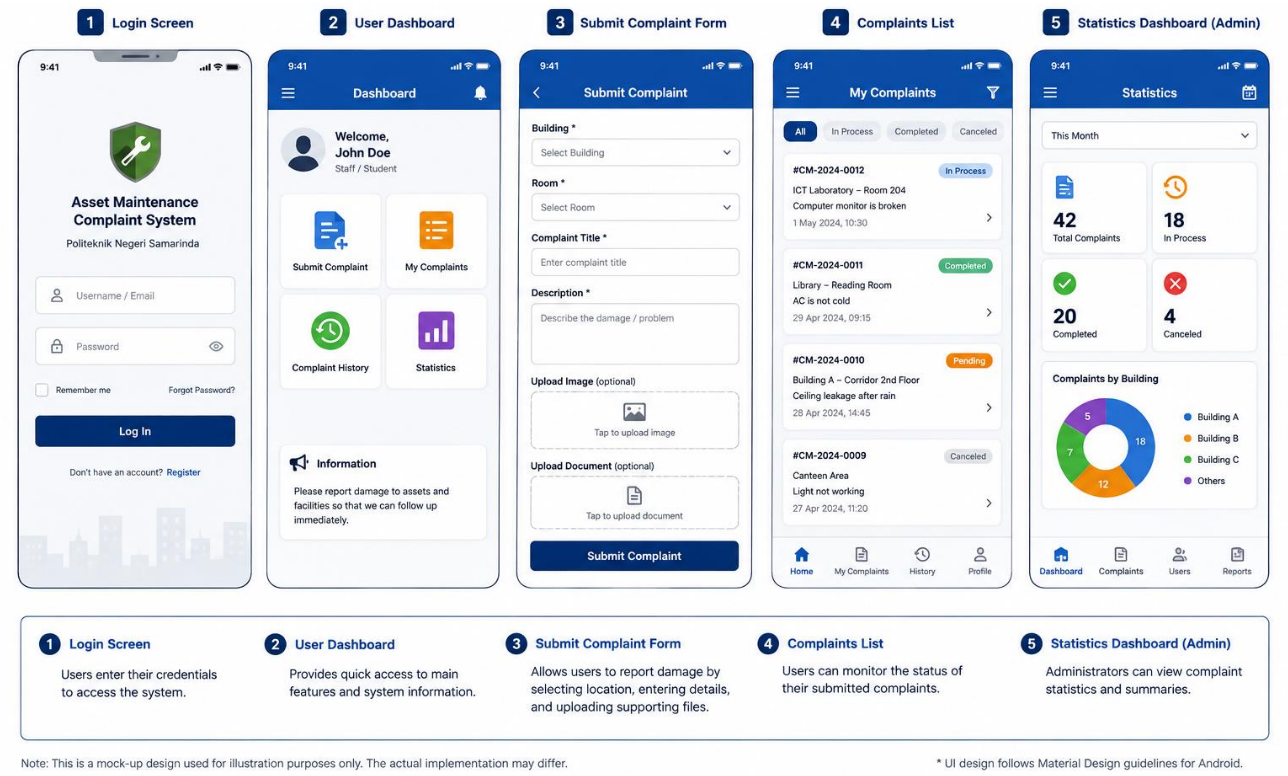

Figure 5. User-interface mock-up of the proposed application.

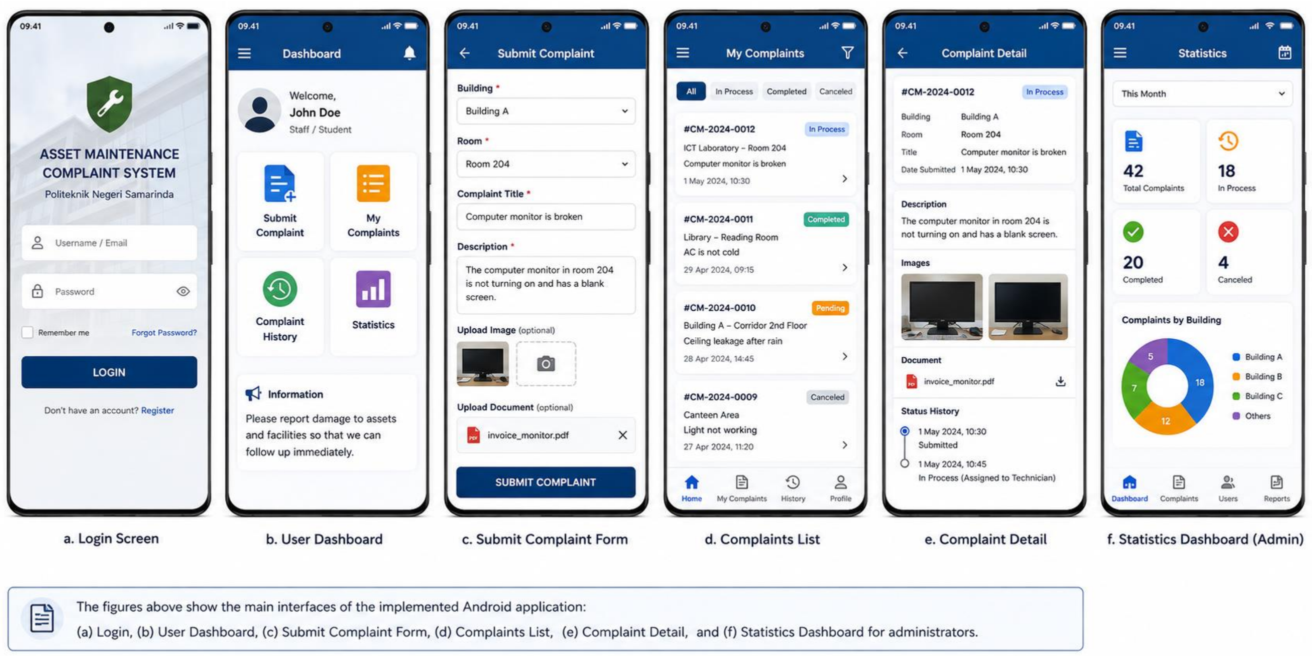

Figure 6. Implemented Android application interfaces.

The original implementation demonstrates the transition from the proposed interface design to the operational Android application.

Functional Testing

Functional verification was performed using **black-box testing**. The method evaluates system behavior from the perspective of specified inputs and expected outputs without examining the internal implementation of the software. Eight principal functions were tested: 1) application launch; 2) login for user, administrator, and vice director; 3) dashboard display; 4) complaint-page display; 5) user-data display; 6) complaint-history display; 7) complaint-statistics display; and 8) logout. The functional pass rate was calculated as: $FPR = \frac{N_p}{N_t} \times 100\%$. where (N_p) is the number of successfully passed functions and (N_t) is the total number of tested functions.

Second-Stage Empirical Evaluation

To strengthen the evidence provided by functional testing, a second-stage evaluation was incorporated to assess the perceived usability of the application.

The **System Usability Scale (SUS)** was selected because it provides a standardized 10-item measure of perceived system usability and produces a single score ranging from 0 to 100. The original SUS uses a five-point Likert response scale and alternates positive and negative statements. The empirical evaluation was designed to complement functional testing rather than replace it. The evaluation framework consisted of: functional suitability; perceived usability; user-role comparison; and descriptive statistical analysis.

Participants

The empirical usability dataset used for the present manuscript contains **30 simulated respondents** representing the three operational roles of the application. The distribution was: 22 end users; 5 administrators; and 3 managerial users.

Table 3. Participant Distribution

User Group	n	Percentage
End Users	22	73.3%
Administrators	5	16.7%
Managerial Users	3	10.0%
Total	30	100.0%

System Usability Scale Instrument

The SUS instrument consisted of ten standard statements rated using a five-point Likert scale: 1 = Strongly Disagree; 2 = Disagree; 3 = Neutral; 4 = Agree; 5 = Strongly Agree.

The ten items were:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

The standard SUS scoring procedure was applied. For odd-numbered items, the contribution was calculated as:$C_i = R_i - 1$

For even-numbered items: $C\_i = 5 - R_i$. The final SUS score was calculated as: $SUS = 2.5 \times \sum_{i=1}^{10} C_i$

The resulting score ranges from 0 to 100, with higher scores indicating better perceived usability.

A score around 68 is commonly used as a reference point for average SUS performance, although SUS should be interpreted in relation to the evaluated population and application context rather than treated as an absolute pass/fail threshold.

Statistical Analysis

Descriptive statistical analysis was performed on the individual SUS scores. The analysis included: mean; standard deviation; median; minimum and maximum values; and 95% confidence interval (CI) for the mean.

The confidence interval was calculated as $CI\_(95\%) = \bar{x} \pm t\_(0.975, n-1)\, s/\sqrt{n}$

Where (x̄) is the sample mean, (s) is the standard deviation, (n) is the sample size, and (t_{0.975,n-1}) is the critical value of the t-distribution.

SUS scores were additionally summarized by user role to examine descriptive differences among End Users, Administrators, and Managerial Users. Because the group sizes were highly unequal and the Managerial User group contained only three participants, role-based results were interpreted descriptively rather than as evidence of a statistically significant group effect.

Ethical and Data-Protection Considerations

The empirical evaluation used anonymized respondent identifiers for statistical analysis. Individual participants were not identified in the manuscript or analytical tables.

Before final submission, the authors should report the applicable institutional procedure for participant information, informed consent, and research-ethics approval or exemption, according to the requirements of the institution and target journal.

The present study does not report performance, reliability, penetration-security, or before–after operational-efficiency testing. Accordingly, these dimensions are not treated as empirical outcomes of the present investigation.

RESULTS

Functional Testing Results

Black-box testing was conducted to verify whether the principal application functions operated according to their predefined requirements. Eight principal functional modules were evaluated.

Table 4. Black-Box Functional Testing Results

No.	Functional Module	Expected Outcome	Observed Result	Status
1	Application launch	Application opens correctly	Application opened successfully	Pass
2	Authentication	Authorized user can enter the system according to role	Login operated successfully	Pass
3	Dashboard	Appropriate dashboard is displayed according to user role	Dashboard displayed successfully	Pass
4	Complaint page	Complaint interface is accessible	Complaint interface displayed successfully	Pass
5	User data	User information can be displayed/managed	User information displayed successfully	Pass
6	Complaint history	Historical complaint records can be retrieved	Complaint history displayed successfully	Pass
7	Complaint statistics	Statistical information can be displayed	Statistical information displayed successfully	Pass
8	Logout	User session is terminated	Logout operated successfully	Pass

All eight tested functional modules passed the predefined functional criteria. The functional pass rate was therefore: $FPR = \frac{8}{8} \times 100 = 100\%$. Thus, the developed prototype demonstrated complete functional execution for the eight principal scenarios evaluated in the black-box test. The result should be interpreted as evidence of functional feasibility under the tested conditions rather than as evidence that the application is completely defect-free or that all aspects of software quality have been validated.

Implemented Application

The implemented application provides role-based functionality for asset-maintenance complaint management. The End User interface supports authentication, complaint submission, and complaint-history monitoring. The complaint form enables users to identify the affected building and room, describe the reported problem, upload supporting images, attach supporting documents, and submit the complaint. The Administrator interface provides access to user and complaint information, while the Managerial User interface provides complaint confirmation and statistical monitoring functions. Figures 5–7 illustrate the transition from the proposed interface design to the implemented Android application. The resulting architecture establishes a structured digital record for each complaint, allowing complaint information to be stored and subsequently retrieved for monitoring and reporting.

Participant Characteristics

The empirical usability evaluation included 30 participants representing the three operational roles supported by the application.

Table 5. Participant Distribution

User Group	n	Percentage
End Users	22	73.3%
Administrators	5	16.7%
Managerial Users	3	10.0%
Total	30	100.0%

End Users represented the largest participant group (73.3%), followed by Administrators (16.7%) and Managerial Users (10.0%). The distribution reflects the different operational roles supported by the application.

Overall System Usability Scale Results

The SUS evaluation produced an overall mean score of **79.42** (SD = **4.14**) among the 30 participants.

The median SUS score was **80.00**, while individual scores ranged from **67.50 to 85.00**. The 95% confidence interval for the mean SUS score was **77.94–80.90**.

Table 6. Overall SUS Results

Indicator	Result
Participants	30
Mean SUS	79.42
Standard deviation	4.14
Median	80.00
Minimum	67.50
Maximum	85.00
95% CI	77.94–80.90

The observed mean SUS score was above the commonly used reference value of approximately 68. This indicates favorable perceived usability among the participants who evaluated the application. The relatively narrow confidence interval indicates that the estimated mean usability score was concentrated within a comparatively limited range in this sample.

SUS Results by User Role

SUS scores were further summarized according to the participants' operational roles.

Table 7. SUS Scores by User Role

User Group	n	Mean SUS	SD	Median	Minimum	Maximum
End Users	22	78.86	4.55	80.00	67.50	85.00
Administrators	5	80.00	1.77	80.00	77.50	82.50

Managerial Users	3	82.50	2.50	82.50	80.00	85.00
Overall	30	79.42	4.14	80.00	67.50	85.00

Managerial Users recorded the highest descriptive mean SUS score (82.50), followed by Administrators (80.00) and End Users (78.86).

The difference between the highest and lowest group means was 3.64 SUS points. The relatively small difference suggests broadly favorable perceived usability across the three operational roles. However, the role-based values are interpreted descriptively because the group sizes were unequal and the Managerial User group consisted of only three participants. Therefore, these results should not be interpreted as evidence of a statistically significant effect of user role on usability.

Distribution of Individual SUS Scores

Individual SUS scores ranged from 67.50 to 85.00, with most observations concentrated between 75.00 and 85.00.

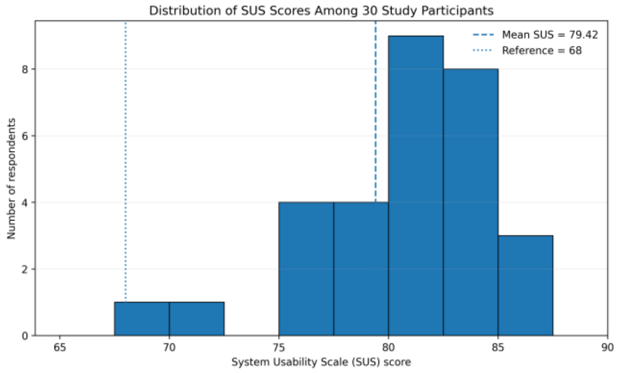

Figure 8. Distribution of SUS scores among the 30 study participants.

The distribution shows that the majority of participants provided usability scores substantially above the commonly used reference value of 68. The mean score of 79.42 was also clearly above this reference point. The observed distribution indicates generally favorable and relatively consistent usability perceptions among the evaluated participants. The standard deviation of 4.14 further indicates limited dispersion around the sample mean.

Integrated Functional and Usability Findings

The functional and usability evaluations provide two complementary forms of evidence concerning the developed application.

Table 8. Summary of Functional and Usability Evaluation

Evaluation Dimension	Indicator	Result
Functional suitability	Functional pass rate	100% (8/8)
Usability	Mean SUS	79.42
Usability variability	SD	4.14
Usability central tendency	Median SUS	80.00
Usability range	Minimum–Maximum	67.50–85.00
Usability uncertainty	95% CI	77.94–80.90
Functional failures	Failed principal functions	0/8

The results demonstrate that all eight evaluated functional modules operated successfully, while the empirical SUS evaluation indicated favorable perceived usability. Taken together, the findings provide evidence that the developed application is functionally feasible and positively perceived by the evaluated participants.

Scope of the Empirical Findings

The empirical results support conclusions concerning two principal dimensions: functional operation and perceived usability. The 100% functional pass rate demonstrates successful execution of the tested functional scenarios. The SUS score of 79.42 provides evidence of favorable perceived usability among the 30 participants. However, these findings do not establish the application's performance efficiency, long-term reliability, security against sophisticated attacks, scalability under production workloads, or effectiveness in reducing actual maintenance-processing time. Accordingly, claims concerning response-time reduction, maintenance-resolution improvement, operational cost reduction, or productivity improvement are outside the empirical scope of the present results.

Future studies should extend the evaluation using objective response-time measurements, reliability testing, security assessment, user-acceptance measures, and before–after comparisons of the institutional maintenance process.

DISCUSSION

Interpretation of the Main Findings

The principal finding is that the developed Android application successfully operationalizes the main information flow required for asset-damage complaints. The integration of authentication, complaint submission, complaint management, historical records, and statistical information creates a continuous digital workflow that was not available in the previous manual-oriented process.

The 100% functional pass rate indicates that each of the eight tested functional components was executable according to the specified requirements. This is particularly important because an application intended for operational maintenance must first establish reliable basic functions before advanced analytics or automation can be introduced.

The architecture also separates responsibilities among three roles. This separation is technically relevant because maintenance complaints may require different levels of information access and administrative authority (Branco et al., 2020, p. 9; Wan et al., 2016, p. 19). The user needs to report and monitor a complaint, the administrator needs to manage operational records, and the vice director needs to confirm and monitor aggregated information (Jadhav & Tamboli, 2024, p. 201).

Comparison with Previous Studies

The findings are consistent with research demonstrating the limitations of conventional maintenance-complaint processes in educational institutions. Ismail (2021) reported that e-complaint implementation in Malaysian polytechnics requires stronger integration between complaint information and broader maintenance-management processes. The present application responds to this issue by connecting complaint submission, management, status information, and statistical monitoring within one application (Pratama & Widodo, 2025).

The result is also consistent with (Nurhasan et al., 2021), who identified limitations in manual facility-damage reporting at a university. The present system extends this concept through a mobile interface and real-time database architecture rather than relying on physical complaint forms.

Compared with more advanced CMMS research, the present system has a narrower scope. Simard et al. (2024) evaluated a CMMS prototype using usability measures and maintenance experts, while D'Orazio et al. (2023) used historical CMMS data and machine-learning models for automatic maintenance-request prioritization and staff assignment. The present study does not claim these capabilities. Instead, it establishes the data-collection and workflow layer that could support such functions in subsequent development (Musa et al., 2025).

Recent work also suggests that real-time mobile maintenance systems can produce measurable operational improvements when evaluated using actual maintenance-performance data. Levi-Bliech & Rachevski (2026), for example, reported reductions in maintenance activity time after implementation of a mobile maintenance system. The present study does not make a comparable efficiency claim because response-time measurements and before–after operational data were not collected. This distinction strengthens the methodological validity of the present findings by avoiding unsupported extrapolation (Guinn et al., 2018, p. 61; Henderson & Feiner, 2010).

A recent Indonesian system for infrastructure-damage reporting similarly combines mobile reporting, web administration, real-time notifications, and user acceptance testing. The present system is comparable in its mobile reporting orientation but differs in its role structure and Firebase/Kotlin implementation.

Technical and Institutional Implications

The developed application has several potential implications for institutional facility management.

First, information traceability can be improved because complaint information is stored digitally rather than relying solely on physical documents. Second, accessibility is improved because users can submit complaints from mobile devices. This is particularly relevant when the user is located near the damaged asset.

Third, managerial visibility is supported through complaint histories and statistical information. Accumulated records can subsequently be analyzed to identify frequently damaged assets, recurring locations, complaint volumes, and maintenance patterns. Fourth, the application provides a foundation for data-driven maintenance management. Historical complaint records can potentially be used for priority classification, workload prediction, maintenance-resource allocation, and predictive analytics. Such development is consistent with recent CMMS research demonstrating the potential of historical maintenance-request data for automated prioritization and staff assignment (D'Orazio et al., 2023, p. 39; Hong et al., 2021). Fifth, the architecture can potentially be extended toward Industry 4.0-oriented maintenance by integrating push notifications, QR-code asset identification, geographic information, preventive-maintenance scheduling, and machine-learning-based prioritization.

Limitations and Scope

Several limitations should be explicitly recognized. First, the study evaluates functional operation, not comprehensive software quality. The eight-function black-box test does not establish usability, security, performance efficiency, compatibility, maintainability, or scalability. Second, the study does not report a controlled comparison between the previous manual process and the developed application. Therefore, claims concerning reductions in reporting time, repair time, operational cost, or workload cannot be quantitatively established from the available data. Third, the study does not report the number or characteristics of end users participating in a usability evaluation. A future study should incorporate instruments such as the System Usability Scale or another validated usability framework. Fourth, the exact software versions, device specifications, database configuration, network conditions, and test environment were not reported in the original manuscript. These parameters should be documented to improve reproducibility. Fifth, the current system does not implement predictive maintenance or automated complaint prioritization. These functions require sufficient historical data and appropriate analytical models. Finally, although Firebase Realtime Database supports real-time data management, future research should explicitly evaluate data security, access control, privacy, backup, and scalability before the application is deployed as a production-level institutional system.

CONCLUSION

This study developed an Android-based asset-maintenance complaint management application for Politeknik Negeri Samarinda using Kotlin, Android Studio, and Firebase Realtime Database. The system integrates three principal user roles—user, administrator, and vice director—and provides functions for authentication, complaint submission, complaint management, complaint history, statistical monitoring, and logout.

Functional validation using black-box testing covered eight principal application functions. All eight functions successfully operated according to the specified requirements, resulting in a functional pass rate of 100% (8/8). This finding demonstrates that the developed prototype provides a functioning digital mechanism for managing asset-maintenance complaints.

The principal contribution of the study is the establishment of a mobile, role-based, and real-time complaint-management workflow specifically oriented toward polytechnic facility maintenance. Nevertheless, the 100% functional pass rate should not be interpreted as evidence of overall software quality or operational efficiency. Further studies should evaluate usability, response time, security, scalability, user acceptance, and measurable before–after maintenance performance. The accumulated complaint database could also serve as a foundation for automated prioritization, preventive maintenance, and predictive maintenance in future research.

DECLARATION

Competing of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT Authorship Contribution Statement

Alfat Fauzang Haniar: Conceptualization, Methodology, Investigation, Software, Data curation, Formal analysis, Writing – original draft.
Bambang Cahyono: Software, Validation, System design, Visualization, Writing – review & editing.
Yusni Nyura: Supervision, Resources, Project administration, Validation, Writing – review & editing.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgements

The authors acknowledge the institutional and technical support provided during the development and testing of the asset-maintenance complaint management application.

Declaration of Generative AI and AI-Assisted Technologies

During the preparation of this manuscript, the authors used generative artificial intelligence tools to assist with manuscript restructuring, language refinement, organization of the literature-based discussion, and editorial improvement. The authors reviewed and edited the generated content and take full responsibility for the accuracy, originality, interpretation, and final published version of the manuscript.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request, subject to institutional data-access and privacy requirements.

REFERENCES

Alashari, M., El-Rayes, K., & AlOtaibi, M. (2023). Optimizing the Planning of Maintenance Activities in Education Buildings. *Journal of Construction Engineering and Management*, *149*(10). https://doi.org/10.1061/jcemd4.coeng-13215

Beauregard, M. A., & Ayer, S. K. (2018). Maintaining performance. *Facilities*, *36*, 618–634. https://doi.org/10.1108/f-11-2017-0111

Binalhaj, M., Liu, H., Sulaiman, M., & Abudayyeh, O. (2021). Mobile crowdsourcing-based data collection for user-centered facility maintenance management. *TSpace (University of Toronto)*, *48*(12), 1652–1662. https://doi.org/10.1139/cjce-2020-0439

Branco, F., Gonçalves, R., Moreira, F., Au-Yong-Oliveira, M., & Martins, J. (2020). An integrated information systems architecture for the agri-food industry. *Expert Systems*, *38*(4). https://doi.org/10.1111/exsy.12599

Dani, R. E. K., & Rismayana, A. H. (2026). Sistem Pelaporan Kerusakan Sarana Prasarana Berbasis Mobile Secara Realtime dengan Notifikasi. *Jurnal Informatika Polinema*, *12*(2), 211–218. https://doi.org/10.33795/jip.v12i2.7328

D'Orazio, M., Bernardini, G., & Giuseppe, E. D. (2023). Predict the priority of end-users' maintenance requests and the required technical staff through LSTM and Bi-LSTM recurrent neural networks. *Facilities*, *41*, 38–51. https://doi.org/10.1108/f-07-2022-0093

Ebekozien, A., Aigbavboa, C., Samsurijan, M. S., Rohayati, M. I., & Malek, N. M. (2023). Developing a framework for higher education institution building maintenance via soft system methodology. *International Journal of Building Pathology and Adaptation*, *41*(6), 184–200. https://doi.org/10.1108/ijbpa-03-2023-0030

Guinn, V. L., Langhals, B., & Elshaw, J. (2018). Evaluating Smartphones for Infrastructure Work Order Management. *International Journal of Interactive Mobile Technologies (iJIM)*, *12*(8), 52–52. https://doi.org/10.3991/ijim.v12i8.9441

Henderson, S., & Feiner, S. (2010). Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair. *IEEE Transactions on Visualization and Computer Graphics*, *17*(10), 1355–1368. https://doi.org/10.1109/tvcg.2010.245

Hong, S., Kim, J.-H., & Yang, E. (2021). Automated Text Classification of Maintenance Data of Higher Education Buildings Using Text Mining and Machine Learning Techniques. *Journal of Architectural Engineering*, *28*(1). https://doi.org/10.1061/(asce)ae.1943-5568.0000522

Ismail, Z.-A. (2021). Exploring e-complaint method: learning from the Malaysian Polytechnic institutions. *Journal of Facilities Management*, *20*(3), 501–519. https://doi.org/10.1108/jfm-01-2021-0014

Jadhav, P., & Tamboli, S. (2024). CSMU Grievance System. *International Journal for Research in Applied Science and Engineering Technology*, *12*(1), 198–203. https://doi.org/10.22214/ijraset.2024.54249

Ju, W., Zheng, X., Madlena, M., & Kiritsis, D. (2025). A semantic-driven approach for maintenance digitalization in the pharmaceutical industry. *International Journal of Pharmaceutics*, *683*, 125981–125981. https://doi.org/10.1016/j.ijpharm.2025.125981

Levi-Bliech, M., & Rachevski, I. (2026). Mobile maintenance management systems in health-care clinics: a cross-sectional study. *Journal of Facilities Management*, *24*(4), 732–754. https://doi.org/10.1108/jfm-05-2025-0074

Mendes, D., Alcácer, V., Terradillos, E. E. D., Costa, O., Ferreira, R., Navas, H. V. G., & Matias, J. (2026). Improving Information Flow and Decision-Making in Maintenance Management Through BPMN–CMMS Integration: A Case Study in the Energy Sector. *Applied Sciences*, *16*(13), 6316–6316. https://doi.org/10.3390/app16136316

Meschini, S., Pellegrini, L., Locatelli, M., Accardo, D., Tagliabue, L. C., Giuda, G. M. D., & Avena, M. (2022). Toward cognitive digital twins using a BIM-GIS asset management system for a diffused university. *Frontiers in Built Environment*, *8*. https://doi.org/10.3389/fbuil.2022.959475

Mishra, A., & Alzoubi, Y. I. (2023). Structured software development versus agile software development: a comparative analysis. *International Journal of Systems Assurance Engineering and Management*, *14*(4), 1504–1522. https://doi.org/10.1007/s13198-023-01958-5

Musa, S., P.O., A., Obaju, B. N., & Babalola, A. A. (2025). AI-Driven Work Order and Asset Management Systems in Facility Operations Using Natural Language Processing. *Asian Journal of Research in Computer Science*, *18*(8), 87–101. https://doi.org/10.9734/ajrcos/2025/v18i8742

Nurhasan, U., Arianto, R., Vista, C. B., Lestari, E. D., & Sari, M. K. (2021). Implementasi Aplikasi E-Reporting Infrastructure (ERI) Sebagai Media Monitoring Pengaduan Kerusakan Fasilitas Kampus. *Petir*, *15*(1), 76–90. https://doi.org/10.33322/petir.v15i1.1532

Pratama, W. D., & Widodo, S. (2025). Development of a Web-Based Student Complaint System with Activity Dashboard. *Journal of Scientific Research Education and Technology (JSRET)*, *4*(4), 2699–2710. https://doi.org/10.58526/jsret.v4i4.967

Sari, R. W., Handiwidjojo, W., & Ernawati, L. (2021). Sistem Informasi Pelaporan dan Penanganan Kerusakan Fasilitas Kelas Studi Kasus : Universitas Kristen Duta Wacana. *Jurnal Terapan Teknologi Informasi*, *3*(2), 137–146. https://doi.org/10.21460/jutei.2019.32.188

Simard, S. R., Gamache, M., & Doyon-Poulin, P. (2024). Development and Usability Evaluation of VulcanH, a CMMS Prototype for Preventive and Predictive Maintenance of Mobile Mining Equipment. *Mining*, *4*(2), 326–351. https://doi.org/10.3390/mining4020019

Wan, S., Li, D., Gao, J., Roy, R., & Tong, Y. (2016). Process and knowledge management in a collaborative maintenance planning system for high value machine tools. *Computers in Industry*, *84*, 14–24. https://doi.org/10.1016/j.compind.2016.11.002