



Implementation of a Web-Based Ticketing and Monitoring Information System for Occupational Safety and Health Incident Prevention at PT Takaki Indoroti Prima

Muhammad Fadly Yassin^{1*}, Dilvan Qiblatain², Indah Purwandani³

¹⁻³ Universitas Bina Sarana Informatika, Indonesia

email: muhammadfadlyyassin@gmail.com¹

Article Info :

Received:
25-09-2026
Revised:
05-10-2026
Accepted:
09-10-2026

Abstract

Occupational Safety and Health (OSH) management is essential for maintaining industrial operations, particularly where incident reporting and maintenance follow-up require timely coordination. At PT Takaki Indoroti Prima, reporting practices created limitations in data organization, status monitoring, traceability, and response time. This study aimed to design and implement a web-based ticketing and monitoring information system to support integrated reporting, tracking, and prevention of OSH-related incidents. The system was developed using the Waterfall model through requirements analysis, system design, implementation, testing, deployment, and maintenance. The application was built with Laravel and supported ticket creation, status monitoring, user management, machine and spare-part management, work-history recording, report printing, and dashboard-based information presentation. Functional validation was conducted using Black Box Testing on login, ticket input, user input, machine input, spare-part input, and ticket-management functions. The recorded test matrix comprised 20 scenarios, and all scenarios produced outputs consistent with the expected results, corresponding to a 100% pass rate for the documented functional tests. The implemented system improved reporting structure, strengthened traceability, facilitated monitoring of corrective actions, and provided more consistent operational documentation. These results indicate that the system can serve as a practical digital foundation for integrated maintenance and OSH reporting at PT Takaki Indoroti Prima.

Keywords: Information System, Monitoring, Occupational Safety and Health, Ticketing, Web-Based System.



©2026 Authors.. This work is licensed under a Creative Commons Attribution-Non Commercial 4.0 International License.
(<https://creativecommons.org/licenses/by-nc/4.0/>)

INTRODUCTION

The rapid advancement of digital information technologies has progressively transformed the manner in which industrial organizations collect, process, communicate, and utilize operational data, with web-based information systems becoming increasingly important for integrating dispersed activities into traceable and continuously accessible workflows. Contemporary occupational safety and health management is undergoing a similar transition, as organizations increasingly require digital infrastructures capable of converting fragmented safety observations, operational complaints, equipment problems, and near-miss information into structured records that can support timely intervention and organizational learning (Pratiwi et al., 2025). This transformation is particularly consequential in industrial environments because conventional reporting through electronic mail, handwritten documents, instant messaging, or verbal communication frequently leaves information distributed across disconnected channels and makes subsequent verification difficult. Empirical evidence concerning digital occupational safety management indicates that web-based reporting mechanisms can improve the organization and accessibility of safety-related information when they are embedded within operational management processes rather than treated merely as administrative databases (Ningtyas et al., 2024). At the same time, digital innovation in occupational health and safety increasingly links technological capability with operational efficiency, requiring systems that are sufficiently structured to support both routine work coordination and preventive safety management (Nugrahayani & Bunahri, 2025). The central technological challenge, consequently, is no longer simply the conversion of paper records into electronic form, but the construction of an information architecture through which each reported condition can be identified, assigned, monitored, verified, and retained as

an auditable record. Web-based occupational safety risk-assessment prototypes further demonstrate that centralized information environments can facilitate systematic risk documentation, although their practical value remains dependent on the extent to which reporting functions are connected to subsequent organizational action (Hardiani, 2025).

Within this broader digital transformation, ticketing systems provide a particularly relevant architecture because they convert individual reports and service requests into identifiable records whose status, responsibility, chronology, and resolution can be monitored throughout the handling process. Previous implementations have demonstrated that web-based ticketing can reduce the fragmentation associated with informal complaint management by establishing standardized mechanisms for recording, assigning, processing, and closing individual cases (Dirmanthara et al., 2022). Similar findings from organizational helpdesk environments indicate that ticket-based workflows can improve the coordination of maintenance and corrective activities because technicians and users operate through a shared record rather than relying on independently maintained communications (Herdiansyah et al., 2023). Nevertheless, the functional logic required for occupational safety reporting is more demanding than that required for an ordinary service desk, since delayed escalation, incomplete documentation, or the disappearance of a near-miss report may affect not only administrative efficiency but also the probability that hazardous conditions remain uncontrolled. Research on hazard-reporting applications supports this distinction by showing that systematic digital reporting can contribute to accident-prevention efforts when reported hazards are translated into identifiable corrective actions and subsequent monitoring (Salsabila & Wahyuningsih, 2023). Near-miss information is particularly valuable because patterns of minor unsafe events may reveal relationships among hazards that would remain invisible when incidents are recorded independently or only after an accident has occurred (Anggraini et al., 2024). The significance of such reporting extends beyond documentation itself, since prospective evidence indicates that an organization's adequacy of response to near-miss events is associated with subsequent occupational accident outcomes (Yamamoto et al., 2024).

Despite these developments, the existing literature remains divided between studies emphasizing general-purpose ticketing architectures and studies concentrating on occupational safety reporting, leaving comparatively limited examination of systems that integrate ticket lifecycle management directly with preventive OHS monitoring in a specific industrial operating environment. Several recent systems have successfully digitized occupational safety functions, including web-based K3 employee-information applications and contractor supervision mechanisms, yet their principal objectives do not fully address the combined requirements of ticket creation, responsibility allocation, progress tracking, historical traceability, and closure verification within a single preventive incident-management workflow (Melani et al., 2025). Research on web-based contractor reporting similarly demonstrates the value of digital monitoring for strengthening occupational safety management, but such applications are oriented toward contractor performance and supervision rather than a generalized mechanism through which employees can report and monitor operational or safety-related findings from initiation to resolution (Wardani, 2025). More technologically advanced approaches, including privacy-preserving near-miss systems based on blockchain and zero-knowledge proofs, have expanded the technical possibilities of secure safety reporting, although these architectures may introduce implementation complexity that is disproportionate to the operational requirements of organizations primarily seeking reliable internal reporting and follow-up mechanisms (Nyato et al., 2024). Existing e-ticketing research also shows that iterative software-development approaches can produce responsive ticket-management applications, yet the resulting systems are generally evaluated in terms of application functionality rather than their capacity to structure preventive occupational safety processes (Karimah et al., 2024). The unresolved issue therefore lies not in whether digital reporting is technically feasible, but in how a practically deployable web application can integrate reporting, status monitoring, responsibility, documentation, and closure into an auditable workflow suited to routine OHS incident prevention. This gap is especially visible where organizations continue to depend on several communication channels simultaneously, since technological fragmentation can reproduce procedural fragmentation even when individual channels are already digital.

The problem has direct scientific and practical relevance for PT Takaki Indoroti Prima, where non-centralized reporting of occupational safety findings and operational problems can generate inconsistent documentation, duplicate submissions, limited visibility of report status, unclear accountability, and delays in corrective action. A centralized ticket repository provides a potential

means of transforming these disconnected reports into structured operational records in which users can distinguish newly submitted, ongoing, completed, or rejected cases while authorized personnel retain responsibility for updating their progress. Such functionality requires an explicit system model capable of representing actors, activities, temporal interactions, and structural relationships, for which Unified Modeling Language offers an established framework for describing web-based information-system behavior and architecture (Hidayati et al., 2023). Database integrity is equally important because the usefulness of historical safety data depends on consistent relationships among users, ticket categories, reports, status changes, and resolution records, making entity-relationship modeling relevant to the organization of the underlying data structure (Pulungan et al., 2023). From an implementation perspective, Laravel provides a structured PHP-based framework through which application logic, routing, database operations, and interface components can be organized within a maintainable web architecture (Sinlae et al., 2024). A Progressive Web App-oriented understanding of contemporary website development also reinforces the expectation that browser-based systems should emphasize accessibility, usability, and compatibility across different user environments rather than functioning merely as static information interfaces (Apriyanti & Ramli, 2025). These technical choices become meaningful only when they correspond directly to identified reporting problems; otherwise, the application risks digitizing an inefficient procedure without improving its visibility, accountability, or preventive value.

This study is therefore positioned at the intersection of information-system engineering and preventive occupational safety management by treating ticketing not merely as an IT helpdesk mechanism but as an operational control instrument for recording, routing, monitoring, and documenting OHS-related findings. The system is deliberately designed around a relatively defined organizational workflow in which operators or employees create and monitor tickets, while technicians or administrators review reports, perform follow-up activities, update ticket status, and maintain the corresponding resolution history. For a project whose principal functional requirements can be identified before implementation, a sequential development process provides a coherent means of connecting requirements analysis, system design, coding, testing, and implementation, which is consistent with the structured characteristics of the Waterfall model (Wahyudi & Heksaputra, 2023). The proposed architecture also incorporates role-based interaction and standardized reporting categories so that access to information and operational actions corresponds to the responsibilities attached to each user group rather than remaining unrestricted or informally distributed. Functional validation is undertaken through Black Box Testing because the critical concern at the application level is whether specified inputs, validation rules, processing operations, and outputs behave according to the defined requirements without requiring assessment of the internal source-code structure during each functional test (Jibril et al., 2024). The system further incorporates web-based recording principles that have been applied to occupational safety management in technically intensive organizational contexts, where accessibility and structured digital documentation are important for maintaining continuity between reporting and managerial evaluation (Pratiwi et al., 2025). Unlike approaches primarily concerned with sophisticated analytical prediction or external cloud infrastructures, the proposed system emphasizes a deployable internal platform capable of establishing a reliable foundation for subsequent safety evaluation and technological extension.

Accordingly, this study aims to design and implement a web-based ticketing and monitoring information system for occupational safety and health incident prevention at PT Takaki Indoroti Prima, with particular emphasis on structured reporting, real-time status visibility, responsibility allocation, historical documentation, and systematic follow-up of operational and safety-related findings. The application is intended to replace fragmented reporting practices with a centralized workflow through which individual reports can be registered, categorized, assigned, processed, verified, and retained until resolution. At the theoretical level, the study extends the application of ticketing-system logic from conventional complaint and technical-service management toward preventive occupational safety governance by conceptualizing each safety finding as a traceable organizational action unit rather than an isolated notification. At the methodological level, it demonstrates how requirements identification, process modeling, database design, sequential software development, role-based implementation, and functional validation can be integrated into a coherent development framework for an organizational OHS information system. The study also establishes a structured digital dataset from which recurring equipment problems, report frequencies, processing statuses, and resolution times may subsequently be

evaluated, even though advanced predictive analytics are outside the scope of the initial implementation. Its contribution therefore resides in connecting software-engineering principles with the operational requirements of incident prevention while maintaining an architecture sufficiently practical for implementation within an internal industrial environment. Through this positioning, the research seeks to provide both an implementable information-system solution for PT Takaki Indoroti Prima and a methodological reference for organizations facing comparable problems of fragmented reporting, limited traceability, and delayed safety-related follow-up.

RESEARCH METHODS

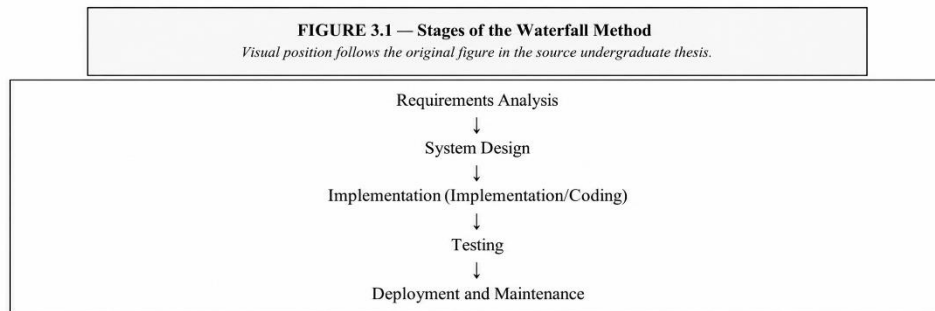


Figure 1. Waterfall Development Stages

Source: Wahyudi and Heksaputra (2023), adapted to the stages of this study

This study employed an empirical system-development approach using the Waterfall model to design and implement a web-based ticketing and monitoring information system for occupational safety and health incident prevention at PT Takaki Indoroti Prima, with development proceeding sequentially through requirements analysis, system design, implementation, testing, deployment, and maintenance (Wahyudi & Heksaputra, 2023). The requirements-analysis stage was grounded in operational observations, supporting documentation, relevant literature, and a direct interview conducted on 21 April 2025 with Rendy Junaedi, IT EDP at PT Takaki Indoroti Prima, to identify existing reporting problems, user roles, workflow constraints, and functional requirements. The resulting requirements were translated into a system architecture comprising authentication, ticket submission, ticket assignment and status management, work-history tracking, ticket summaries, user administration, report generation, and secure logout functions, while nonfunctional requirements covered security, availability, performance, usability, portability, and maintainability. System behavior and structural relationships were modeled using use case, activity, sequence, and class diagrams because UML provides a standardized mechanism for representing actor interactions, process flows, temporal exchanges, and object relationships within web-based information systems (Hidayati et al., 2023). Database structures were subsequently specified through ERD and LRS models, while implementation employed Laravel as the PHP application framework, HTML and CSS for interface presentation, JavaScript for client-side interaction, and MySQL for persistent data storage, with Laravel selected because its structured architecture supports modular and maintainable web application development (Sinlae et al., 2024). Deployment was performed within the intended organizational environment after the principal functions had been implemented, while application handover records, demonstration activities, presentation documentation, certificates, and the formal handover document were retained as implementation evidence and as a basis for subsequent maintenance, bug correction, and functional refinement.

System verification was conducted through functional Black Box Testing, in which each principal application feature was examined from the user perspective by supplying valid and invalid inputs and comparing the resulting outputs with predefined functional requirements without inspecting the internal program code (Jibril et al., 2024). The test procedure covered authentication, ticket creation, ticket-category selection, data validation, ticket assignment, status modification, work-history retrieval, ticket-summary presentation, user management, report generation, and logout, with each case classified as passed when the observed system response corresponded to the expected output and failed when functional deviation occurred. Validation focused on whether the implemented workflow could

transform previously fragmented safety and operational reports into traceable digital records whose submission, processing, responsibility, status, and resolution history could be monitored consistently, reflecting the preventive value associated with structured digital hazard-reporting applications (Salsabila & Wahyuningsih, 2023). Evaluation metrics consisted of functional success rate, input-validation accuracy, correctness of ticket-status transitions, completeness of ticket-history records, successful execution of role-based access functions, and consistency between recorded ticket information and the outputs displayed or generated by the system. System effectiveness was also examined through the availability of centralized ticket statistics, including ticket counts, processing status, and completion information, because web-based K3 reporting systems are intended to improve the accessibility and organization of occupational safety data for monitoring and evaluation purposes (Ningtyas et al., 2024). To support reproducibility, all testing scenarios were derived directly from the documented functional requirements, executed using the same defined user roles and system workflow, and recorded in a structured test matrix containing the tested function, input condition, expected output, observed output, and final pass-or-fail status.

RESULTS AND DISCUSSION

Functional Architecture and Role-Based Ticketing Workflow

The implemented ticketing system was structured around two principal actors, namely Technician/Admin and Operator/Employee, whose access rights were differentiated according to operational responsibility within the reporting and handling workflow. The Technician/Admin role was responsible for managing all tickets, adding users, maintaining work history, administering machine and spare-part data, printing reports, and terminating active sessions, while the Operator/Employee role primarily created tickets, monitored ticket summaries, printed ticket records, and logged out from the system. This separation established a functional boundary between report initiation and report processing, which is essential for maintaining accountability in a centralized ticketing environment. Role differentiation in web-based ticketing systems has been associated with clearer task allocation because individual users interact only with functions corresponding to their responsibilities (Bahrudin et al., 2020). Shared access was retained for authentication, selected monitoring functions, document generation, and logout because these activities were required by both operational groups. The two-actor model therefore provided a simple but controlled architecture for transforming occupational safety and maintenance findings into traceable digital records. Table 1 summarizes the original actor descriptions translated from the source system.

Table 1. Actor Description

Actor	Description
Technician/Admin	Login, Manage All Tickets, Add Users, Manage Work History, manage machine and spare-part data, Print Tickets/Reports, Logout.
Operator/Employee	Login, Input Tickets, View Ticket Summary, monitor tickets belonging to the user, Print Tickets, Logout.

Source: Research Results (2026)

The relationship between actors and system functions was represented through the ticketing system use case shown in Figure 1. The diagram places the Technician/Admin and Operator/Employee outside the system boundary and associates each actor with the functions that can be performed after authentication. Technician/Admin users interact with Manage All Tickets, Add User, Manage Work History, View Ticket Summary, Print Ticket, and Logout, while Operator/Employee users primarily interact with ticket submission, ticket monitoring, printing, and session termination. Authentication is positioned as a common function supporting access to the other use cases, reflecting its role as the gateway to role-based operations. UML modeling is useful in information-system development because it enables user responsibilities and functional interactions to be visualized before or alongside implementation (Hidayati et al., 2023). The diagram also demonstrates that the implemented application does not provide identical privileges to every user, which limits unnecessary modification of operational records. This role-oriented structure became the basis for the access-right configuration implemented in the web application.

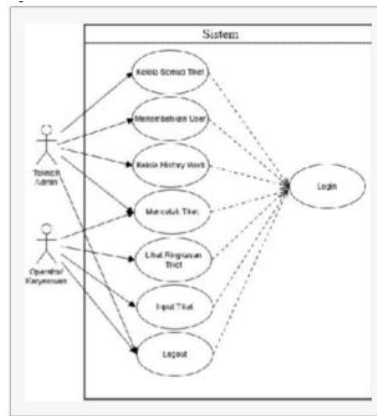


Figure 2. Ticketing System Use Case

The functional distribution can be examined more precisely through the user access-right matrix, which consolidates the functions dispersed across the actor descriptions and individual use cases. The source system documents ten principal functions: Login, Input Ticket, Manage All Tickets, View Ticket Summary, Work History, Add User, Manage Machines, Manage Spare Parts, Print/Export Ticket, and Logout. Technician/Admin users are associated with nine of these functions, whereas Operator/Employee users are associated with five, indicating a broader administrative responsibility for the technician role. Four functions are shared across both roles, namely Login, View Ticket Summary, Print/Export Ticket, and Logout, while ticket creation is assigned specifically to Operator/Employee users. The remaining administrative functions are restricted to Technician/Admin users because they involve processing reports, managing master data, or maintaining system accounts. Similar ticketing implementations emphasize that controlled access contributes to clearer responsibility and prevents operational functions from being performed by unauthorized users (Herdiansyah et al., 2023).

Table 2. User Access Rights Matrix

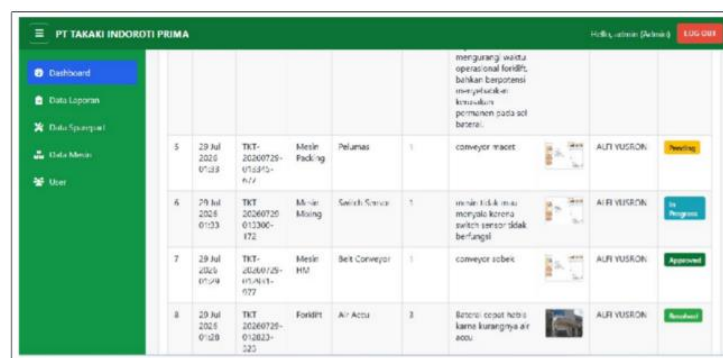
Function	Technician/ Admin	Operator/ Employee	Description
Login	Yes	Yes	Both actors must authenticate.
Input Ticket	–	Yes	Operator creates a report/finding ticket.
Manage All Tickets	Yes	–	Technician/admin processes and updates tickets.
View Ticket Summary	Yes	Yes	Summary is used to monitor ticket status.
Work History	Yes	–	Technician/admin traces previously processed work.
Add User	Yes	–	Account management is performed by technician/admin.
Manage Machines	Yes	–	Machine data serve as supporting master data for tickets.
Manage Spare Parts	Yes	–	Spare-part data support handling activities.
Print/Export Ticket	Yes	Yes	Ticket documents can be printed or downloaded as PDF.
Logout	Yes	Yes	Ends the user session securely.

Source: Research Results (2026).

The ticket-processing workflow begins when an Operator/Employee authenticates and creates a ticket containing information about an operational problem or occupational safety finding. Once submitted, the ticket is stored in the system database and becomes accessible to Technician/Admin users through the Manage All Tickets interface. The administrator can then review the report, update its status

according to the progress of corrective work, and record the resulting activity within the work-history component. This structure differs from informal reporting channels because each reported problem remains associated with a persistent digital record throughout the handling process. Web-based ticketing systems are particularly valuable when they retain continuity between problem submission, processing, monitoring, and final documentation rather than treating each stage as an independent communication event (Dirmanthara et al., 2022). The implemented workflow also connects machine and spare-part information with ticket processing, which allows operational resources used during corrective work to remain linked to the reported event. At completion, ticket information may be printed or exported as a PDF document for documentation and subsequent evaluation.

The implemented ticket lifecycle was defined through four principal processing statuses: Pending, Approved, In Progress, and Resolved. A newly submitted record begins in Pending status, after which Technician/Admin users can move it to Approved when the report has been accepted for handling. In Progress indicates that corrective work is actively being performed, while Resolved represents completion of the reported issue. Explicit status differentiation provides users with a visible representation of treatment progress and reduces uncertainty about whether a report has merely been submitted or is already being handled. This capability is relevant to occupational safety management because the value of reporting mechanisms depends not only on the number of submitted observations but also on the organizational response that follows each report (Yamamoto et al., 2024). The source testing scenarios confirmed that the system supported updates to each of these four statuses and also included an administrative ticket-deletion scenario.



ID	Date	Ticket ID	Machine	Location	Quantity	Description	Assigned To	Status
5	29 Jul 2024	TKT-20240729-0128-0129	Alat Packing	Pelamas	1	merusak waktu operasional mesin, butuh perbaikan mesin perbaikan permanen pada alat ini.	AURI YUSRON	Pending
6	29 Jul 2024	TKT-20240729-0129-0130-0131	Alat Masing	Sekolah Sinar	1	mesin rusak mau diganti karena switch sensor tidak berfungsi	AURI YUSRON	In Progress
7	29 Jul 2024	TKT-20240729-0130-0131-0132	Alat Belt Conveyor	HSU	1	conveyor sobek	AURI YUSRON	Approved
8	29 Jul 2024	TKT-20240729-0132-0133	Roller	Air Acti	3	Roller rusak perlu ganti karena keganggu alat	AURI YUSRON	Resolved

Figure 3. Tested Ticket Status Lifecycle

From an occupational safety information-management perspective, the combination of role-based access and explicit ticket status creates an auditable relationship among the reporting user, processing authority, operational progress, and final record. Reports that previously could have been dispersed through informal communication are instead converted into identifiable transactions whose lifecycle can be monitored through the same application. Digital hazard-reporting studies indicate that structured recording can improve the visibility of workplace safety information and support the identification of recurring patterns across incidents and near-miss observations (Anggraini et al., 2024). The implemented architecture does not yet perform predictive risk modeling, but the database created through routine ticket processing establishes a foundation for subsequent analysis of problem frequency, machine involvement, spare-part requirements, status distribution, and corrective response. Its immediate contribution lies in strengthening traceability and responsibility rather than introducing complex analytical automation. Web-based occupational safety applications similarly demonstrate that digitization is most useful when information can support both operational handling and subsequent managerial evaluation (Pratiwi et al., 2025). The architecture implemented at PT Takaki Indoroti Prima consequently provides an integrated mechanism through which reporting, access control, monitoring, corrective handling, and documentation can operate within a single information environment.

Process Modeling and System Interaction Architecture

The operational behavior of the ticketing system was modeled through activity and sequence diagrams to describe how user actions were translated into controlled application responses from

authentication to transaction completion. Activity diagrams represented the procedural flow of login, ticket management, user administration, work-history access, report generation, ticket summary retrieval, ticket creation, and logout, whereas sequence diagrams described the temporal interaction among the browser, middleware, controllers, models, and supporting services. This dual representation was necessary because process-oriented diagrams explain what users and the system do in sequence, while interaction-oriented diagrams reveal how the internal software components execute those processes. UML-based behavioral modeling provides a structured way to relate functional requirements to implementation logic and is particularly useful when a system contains multiple user roles and interconnected functions (Hidayati et al., 2023). In this implementation, each modeled process was aligned with the role separation established between Technician/Admin and Operator/Employee users. The resulting architecture therefore linked user requirements, software execution, and later functional verification within one coherent design structure.

Authentication constituted the first modeled process because every subsequent operation depended on a valid user session. Both Technician/Admin and Operator/Employee users accessed the same login page, entered registered credentials, and were redirected to the appropriate dashboard after successful verification. Invalid credentials generated an error response and prevented access to the ticketing environment, creating an explicit alternate flow that supported access control. The original login use case specified the requirements, goal, preconditions, postconditions, main flow, alternate flow, and actors involved, as preserved in Table 3. Black Box Testing principles emphasize that such predefined input–output expectations are essential because observed behavior can later be compared directly against the system specification (Jibril et al., 2024). Authentication therefore operated not only as an interface component but also as the initial control mechanism for role-based system access.

Table 3. Login Use Case Description

Element	Description
Requirements	The system provides a login interface that verifies the username and password of technicians and operators.
Goal	Provide access to technicians and operators according to their roles.
Pre-Conditions	Email/username and password have been registered; the user knows valid credentials.
Post-Conditions	If valid, the user enters the dashboard; if authentication fails, the system displays an error message.
Main Flow	Access the web application → system displays the login page → user enters username/password → system verifies credentials → if valid, user is redirected to the dashboard.
Alternate Flow	Incorrect credentials → system displays an error message and remains on the login page.
Actors	Technician/Admin and Operator/Employee.

Source: Research Results (2026).

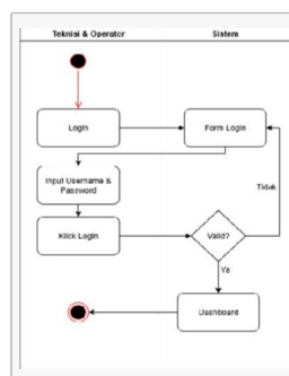


Figure 4. Activity Diagram for Technician and Operator Login

Ticket administration formed the principal corrective-work process because it connected reports submitted by employees with subsequent verification, handling, and status control by Technician/Admin users. After entering the ticket-management page, the administrator could view the available tickets, select a record, examine its details, update its status, and save the modification, while the system also supported ticket deletion after confirmation. This process converted each report into an operational record whose lifecycle could be managed within a single interface rather than through dispersed communication channels. Web-based ticketing architectures are effective when the same record accompanies the issue from submission through processing and closure because this improves traceability and reduces fragmented handling (Dirmanthara et al., 2022). The original use-case specification for this function is retained in Table 4, while Figures 5–7 preserve the source modeling sequence for ticket management, user administration, and work-history management. The placement of these diagrams within the discussion reflects their function as evidence of the modeled operational process rather than as supplementary material detached from the interpretation.

Table 4. Manage All Tickets Use Case Description

Element	Description
Requirements	The system provides a feature for managing all tickets to view, update status, provide follow-up actions, and manage ticket data.
Goal	Facilitate Technician/Admin users in managing and monitoring all incoming tickets.
Pre-Conditions	Technician/Admin is registered, has logged in, and ticket data are available.
Post-Conditions	Ticket data are displayed; status can be updated; change history is stored.
Main Flow	Select Manage All Tickets → ticket list is displayed → select a ticket → ticket details are displayed → perform an update → system saves the change → success notification is displayed.
Alternate Flow	Ticket data are not found or the modification input is invalid → system displays an error message.
Actors	Technician/Admin.

Source: Research Results (2026).

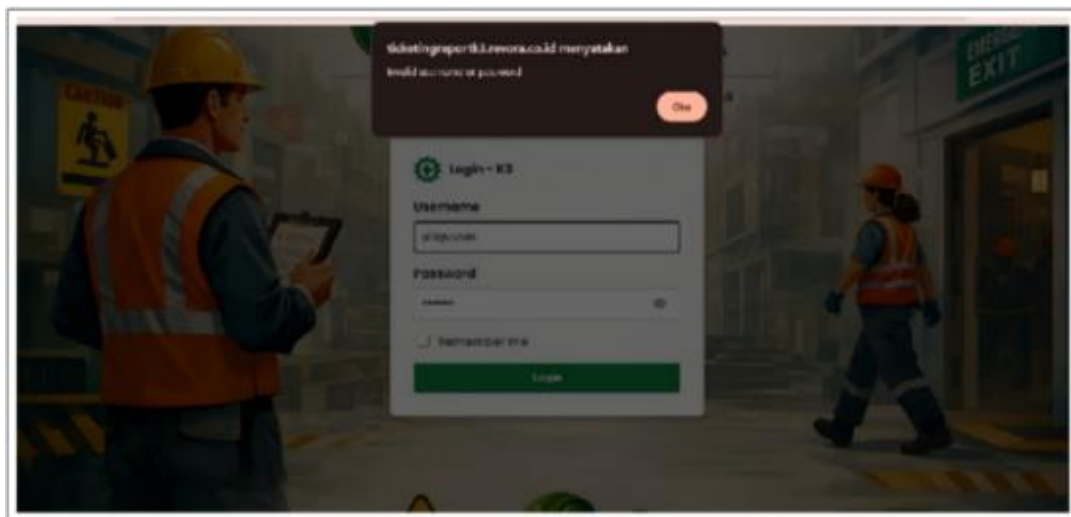


Figure 5. Activity Diagram for Technician Ticket Management

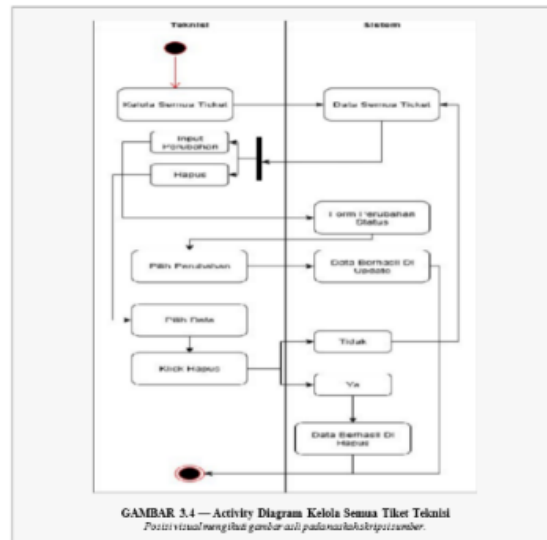


Figure 6. Activity Diagram for Adding Users

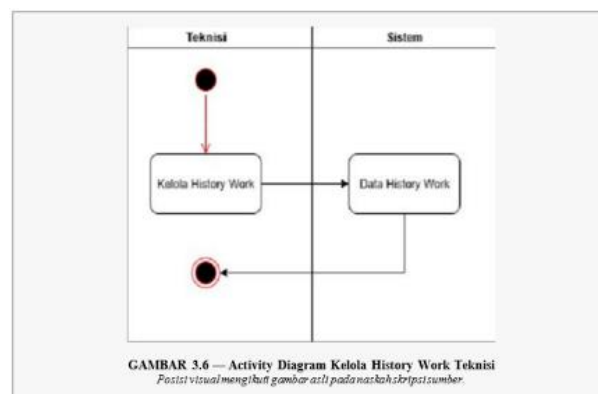


Figure 7. Activity Diagram for Technician Work History Management

The remaining activity models expanded the workflow to ticket printing, ticket-summary retrieval, ticket creation, and session termination. The ticket-printing process selected an existing record and generated a printable or PDF output, while the summary process retrieved user-associated ticket data and displayed the current handling status. The Operator/Employee ticket-creation flow required the entry of complaint details, machine or item information, required spare parts, quantity, and an optional image before submission, after which the application validated and stored the report. These processes demonstrate that the system was designed around modular activities that remain connected through common authentication, ticket data, and access-control mechanisms. Web-based information systems benefit from this modular organization because distinct operational functions can be maintained separately while still sharing a unified database and user-management structure (Nouvel et al., 2021).

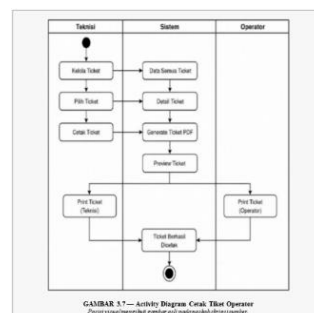


Figure 8. Activity Diagram for Operator Ticket Printing

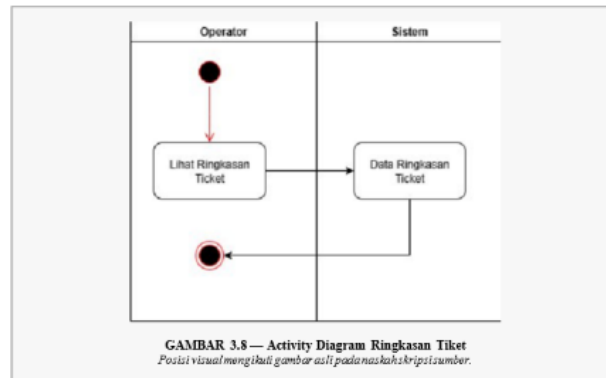


Figure 9. Activity Diagram for Operator Ticket Summary

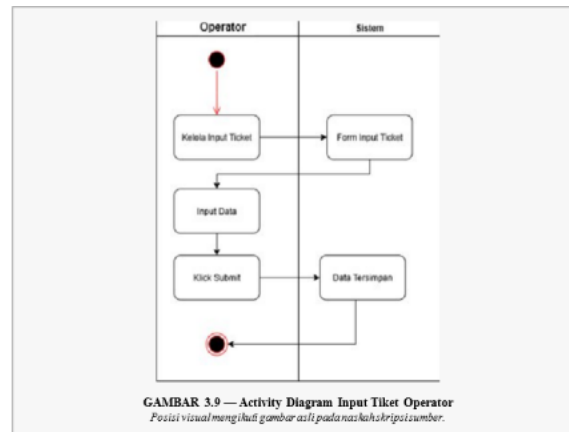


Figure 10. Activity Diagram for Operator Ticket Input

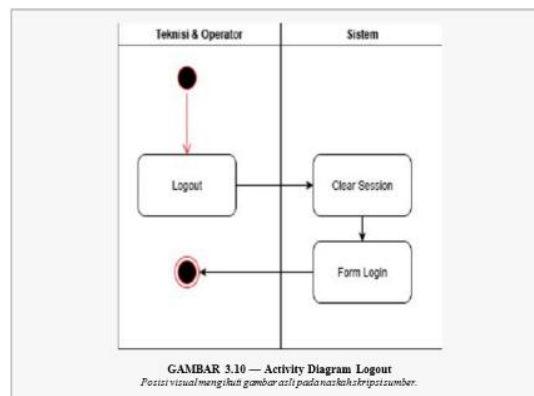


Figure 11. Activity Diagram for Logout

At the software-interaction level, the sequence models showed that user requests passed through several application components before reaching the database. During authentication, the browser forwarded credentials to AuthController, UserModel retrieved user information, the hashing component verified the password, and JwtService generated an authentication token when validation succeeded. Ticket creation followed a similar layered structure in which JwtMiddleware verified the token before TicketController retrieved active machine and spare-part records, validated the submitted information, stored an image when present, generated a ticket code, assigned the initial Pending status, and saved the transaction through TicketModel. This layered architecture separates authentication, business logic, and data persistence and is consistent with modular Laravel-based web development (Sinlae et al., 2024).

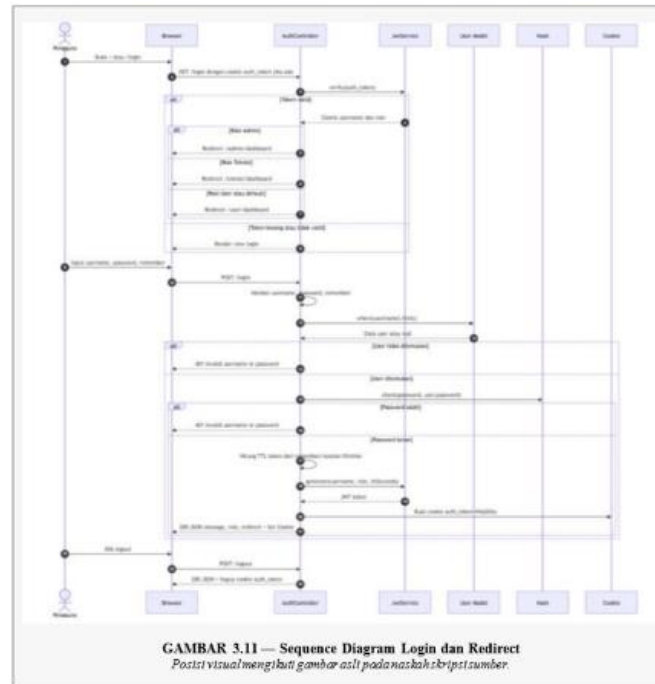


Figure 12. Sequence Diagram for Login and Role-Based Redirect

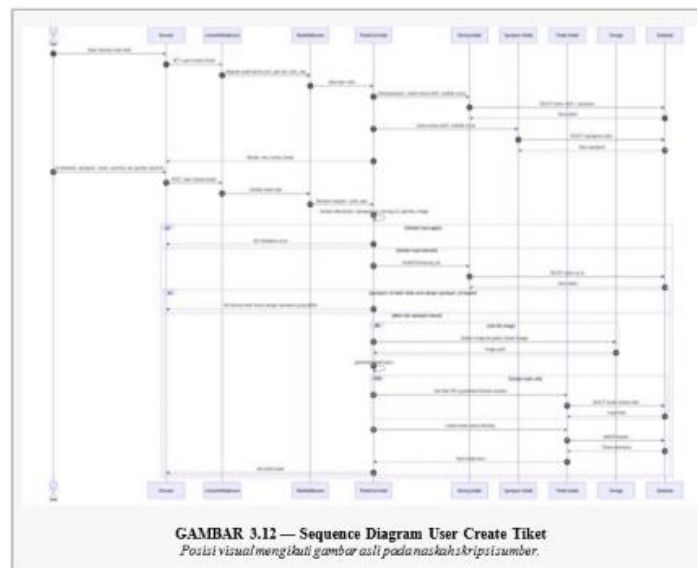


Figure 13. Sequence Diagram for User Ticket Creation

Technician/Admin interactions extended the same architectural pattern to work-history management, machine and spare-part administration, ticket updates, and PDF export. When a ticket was modified, the system verified authentication and role data, retrieved the relevant record, validated the requested change, updated the ticket status, and adjusted spare-part stock when the handling process required component usage. Administrative operations on machine and spare-part records were likewise controlled through JwtMiddleware and RoleMiddleware, while export requests were processed through PdfController after validating filters such as date, ticket code, user name, and role. Figures 14–16 preserve the source interaction models for work-history management, administrative data management, and PDF export, but the analytical interpretation continues beyond the figures so that the visual material does not terminate the subsection. The interaction design demonstrates that access control, transaction processing, inventory adjustment, historical recording, and reporting were technically linked rather than implemented as isolated functions. Web-based OHS information systems gain practical value when reporting and follow-up remain integrated with supporting operational data and documentation

processes (Pratiwi et al., 2025). This architecture provided the technical bridge between process modeling and the database and interface implementation examined in the following section.

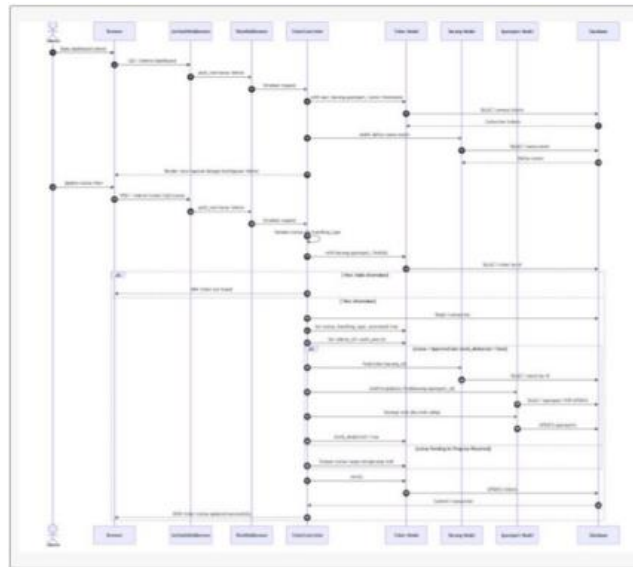


Figure 14. Sequence Diagram for Technician Work History Management

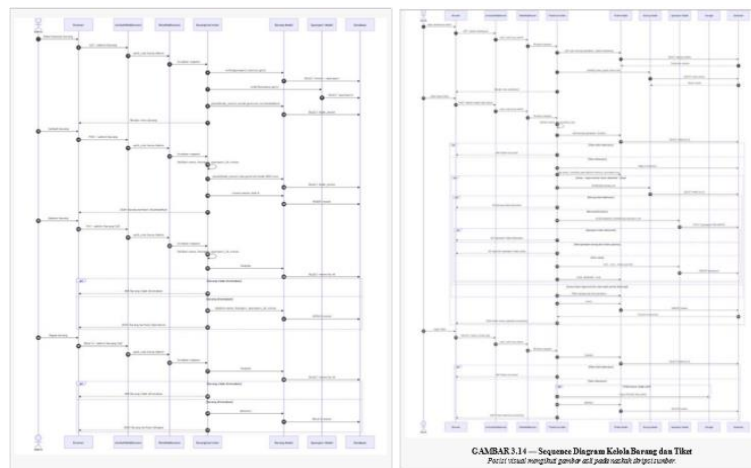


Figure 15. Sequence Diagram for Admin Management of Machines, Spare Parts, and Tickets

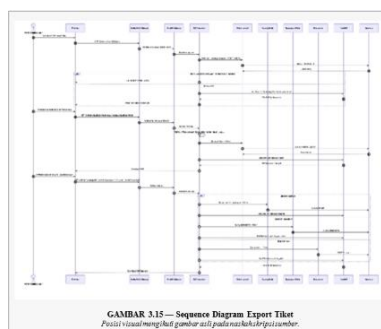


Figure 16. Sequence Diagram for Ticket Export to PDF

The overall process model shows that the implemented application did not treat ticket submission, authorization, data processing, and reporting as independent events. Instead, each user action was routed through a defined control path that preserved user identity, role privileges, ticket status, supporting resource information, and output generation within the same system architecture. This integration is important for occupational safety and maintenance reporting because fragmented

technical processes can reproduce the same traceability problems found in manual communication practices. The modeled relationships therefore provide evidence that the application architecture was aligned with the operational objective of maintaining continuous control over each reported issue from entry to documented output. The sequence between user interface, middleware, controller, model, and database also supports maintainability because each layer has a distinguishable responsibility within the transaction flow. Such modularity is consistent with contemporary web application development, where system robustness depends on clear separation of presentation, control, authentication, and persistence functions (Huzzaemah, 2024).

Database Architecture and Web Interface Implementation

The implemented ticketing system used a structured data architecture in which user accounts, ticket transactions, machine records, spare-part information, activity histories, and reporting components were organized as interrelated entities rather than independent datasets. The class model identified User, Role, and Permission as the principal access-control entities, while Ticket, Category, Priority, and Status formed the core transactional structure of the reporting process. Supporting classes such as Attachment, Comment, TicketHistory, and Assignment extended the ticket entity by preserving contextual information, communication records, historical changes, and task allocation. The model also included Incident and CorrectiveAction to represent occupational safety events and corresponding interventions, while Report, ReportFile, Setting, ActivityLog, and AuditTrail supported reporting, configuration, and traceability. Structuring system objects according to clearly differentiated responsibilities is consistent with object-oriented information-system design because explicit class relationships reduce ambiguity between application functions and persistent data structures (Pulungan et al., 2023).

Table 5. Summary of Main System Classes

Class Group	Classes/Entities	Function
Access	User, Role, Permission	Manage user accounts and access rights.
Audit	ActivityLog, AuditTrail	Record important activities and system usage traces.
Ticketing	Ticket, Category, Priority, Status	Store tickets and their classification attributes.
Collaboration	Attachment, Comment, TicketHistory, Assignment	Support attachments, comments, change history, and task assignment.
Incident	Incident, CorrectiveAction	Record incidents and corrective/preventive actions.
Reporting	Report, ReportFile	Generate and store reports.
Configuration	Setting	Store system configuration.

Source: Research Results (2026).

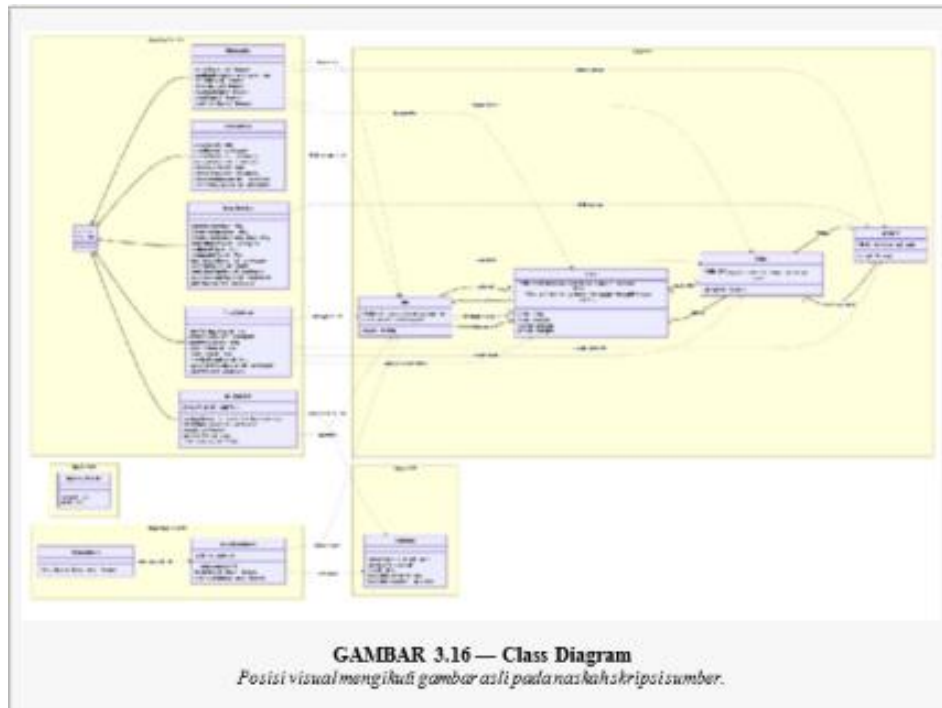


Figure 17. Ticketing and OHS Monitoring System Class Diagram

The relational database architecture concentrated operational activity around the Tickets entity, which connected submitted reports with users, machines or items, spare-part requirements, processing status, and transaction chronology. The Entity Relationship Diagram identified Users, Tickets, Barang, and Sparepart as the principal entities, while the Logical Record Structure translated those relationships into tables containing primary keys, foreign keys, and application attributes. ERD-based modeling is appropriate for transactional web applications because explicit relationships between entities support referential consistency and reduce unnecessary duplication before physical database implementation (Pulungan et al., 2023). The source design assigns user_id to associate each ticket with the submitting user and barang_id to connect the report with the relevant machine or item. Spare-part quantity, status, timestamps, and processing indicators extend each ticket record beyond a simple textual complaint and allow the transaction to reflect operational handling requirements.

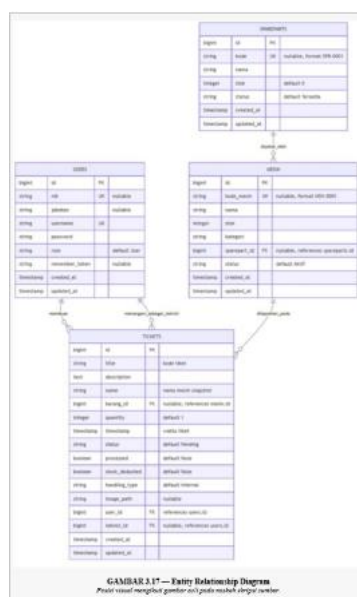


Figure 18. Entity Relationship Diagram

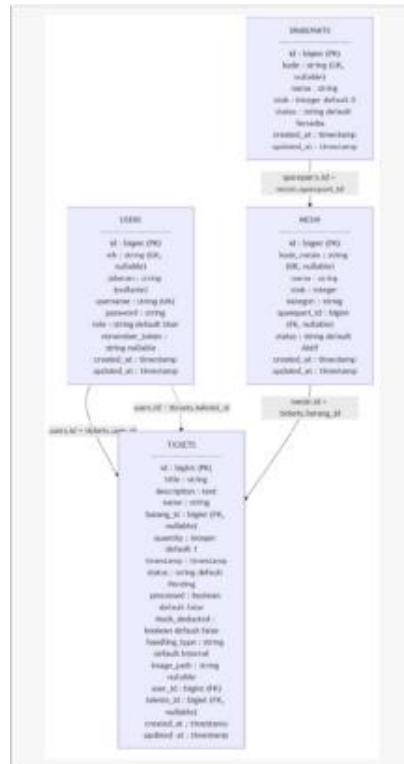


Figure 19. Logical Record Structure

At the physical schema level, the Users and Tickets tables contained the principal fields required to connect authenticated users with submitted operational reports. The Users table included the identifier, employee number, position, username, password, role, session token, and timestamp fields, while uniqueness constraints were assigned to selected identification fields. The Tickets table was more extensive because it recorded the report title, description, creator identity, machine relationship, spare-part quantity, timestamp, ticket status, processing indicators, handling type, image path, user relationship, and audit timestamps. Table 6 and Table 7 reproduce these specifications directly from the source design without altering the declared field types, lengths, or relational functions. This separation between account data and transactional data supports clearer database responsibilities because authentication records remain structurally distinct from incident or maintenance-report records. Laravel-based application development benefits from this organization because data models can map persistent entities to authentication and business logic in a systematic manner (Sinlae et al., 2024). The database structure therefore provides the technical basis for connecting role-based access, ticket creation, and subsequent corrective handling.

Table 6. User File Specification

Name	Data Type	Length	Description
id	BIGINT UNSIGNED	20	Primary Key, AUTO_INCREMENT, NOT NULL
nik	VARCHAR	255	NULL, UNIQUE
jabatan	VARCHAR	255	NULL; e.g., admin, operator, staff
username	VARCHAR	255	NOT NULL, UNIQUE
password	VARCHAR	255	NOT NULL, employee NIK in the source design
role	VARCHAR	255	NOT NULL, admin/user
remember_token	VARCHAR	100	NULL, session reminder
created_at	TIMESTAMP	—	Time when user data were created
updated_at	TIMESTAMP	—	Time when user data were updated

Source: Research Results (2026).

Table 7. Ticket File Specification

Name	Data Type	Length	Description
id	BIGINT UNSIGNED	20	Primary Key, AUTO_INCREMENT, NOT NULL
title	VARCHAR	255	NOT NULL, report title
description	TEXT	–	NOT NULL, problem description
name	VARCHAR	255	NOT NULL, ticket creator name
barang_id	BIGINT UNSIGNED	20	Foreign Key to item/machine data
quantity	INT	11	NOT NULL, spare-part quantity
timestamp	TIMESTAMP	–	NOT NULL, ticket creation time
status	VARCHAR	255	NOT NULL, ticket status
processed	TINYINT	1	Indicator that the ticket has been processed
stock_deducted	TINYINT	1	Indicator of stock deduction
handling_type	VARCHAR	255	Internal ticket-handling type
image_path	VARCHAR	255	NULL, image file location
user_id	BIGINT UNSIGNED	20	Foreign Key to user
created_at	TIMESTAMP	–	Time when the ticket was created
updated_at	TIMESTAMP	–	Time when the ticket was last updated

Source: Research Results (2026).

Machine and spare-part records were maintained in separate master tables so that operational resources could be referenced consistently without repeatedly embedding the same descriptive values within individual ticket records. The Barang table stored machine identification, machine name, stock, category, spare-part linkage, activation status, and audit timestamps, whereas the Sparepart table contained component code, component name, available stock, activation status, and timestamps. This separation allowed machine and component information to be updated independently while remaining available during ticket creation and corrective maintenance processing. Tables 8 and 9 preserve the original field definitions and show that the source database used integer stock variables, unique machine or spare-part codes where specified, and a relational linkage between machine and spare-part data. Such master-data separation improves consistency because recurring resources are represented once rather than being re-entered independently for each transaction. Web-based organizational systems benefit from structured master data because standardized records reduce uncontrolled variation during repeated data-entry activities (Melani et al., 2025). The resulting schema also supports the system workflow in which spare-part stock can be adjusted when ticket handling requires a component.

Table 8. Machine/Item File Specification

Name	Data Type	Length	Description
id	BIGINT UNSIGNED	20	Primary Key, UNIQUE, AUTO_INCREMENT
kode_mesin	VARCHAR	225	UNIQUE, NULL, machine code
nama	VARCHAR	225	NOT NULL, machine name
stok	INT	11	NOT NULL, stock quantity
kategori	VARCHAR	225	NOT NULL, machine category
sparepart_id	BIGINT UNSIGNED	20	Foreign Key to spare part
status	VARCHAR	225	Active/Inactive status
created_at	TIMESTAMP	–	Time when data were created
updated_at	TIMESTAMP	–	Time when data were updated

Source: Research Results (2026).

Table 9. Spare-Part File Specification

Name	Data Type	Length	Description
id	BIGINT UNSIGNED	20	Primary Key, AUTO_INCREMENT
kode	VARCHAR	255	NULL, UNIQUE
nama	VARCHAR	255	NOT NULL, spare-part name
stok	INT	11	NOT NULL, available spare-part quantity
status	VARCHAR	255	Active/Inactive status
created_at	TIMESTAMP	–	Time when data were created
updated_at	TIMESTAMP	–	Time when data were updated

Source: Research Results (2026).

The database architecture was translated into eight principal interface functions covering login, Technician/Admin dashboard access, report history, user management, machine management, spare-part management, employee ticket creation, and Operator/Employee ticket monitoring. These interfaces were not treated as independent visual screens but as role-specific access points to the database entities and application functions established in the system architecture. The login interface relied on Users data for authentication, the technician dashboard used Tickets and Users, report history retrieved Tickets, and the administrative pages corresponded directly with Users, Barang, and Sparepart. The employee reporting interface combined Tickets, Barang, and Sparepart so that a submitted report could contain operational context rather than only free-text information. Browser-based systems are more effective when interface functions correspond clearly to user responsibilities and underlying data structures because this reduces navigational ambiguity and supports repeated operational use (Apriyanti & Ramli, 2025).

Table 10. Relationship between Interfaces, Actors, and Data

Interface	Main Actor	Data Used	Main Function
Login	Technician & Operator	Users	Username/password authentication and access determination.
Technician Dashboard	Technician/Admin	Tickets, Users	Summary and access to system-management functions.
Report Data	Technician/Admin	Tickets	Display history of processed tickets.
User Data	Technician/Admin	Users	Display and add users.
Machine Data	Technician/Admin	Barang	Manage the machine/item list.
Spare-Part Data	Technician/Admin	Sparepart	Manage spare-part lists and stock.
Create Report	Operator/Employee	Tickets, Barang, Sparepart	Create a ticket containing a complaint, machine, spare-part requirement, quantity, and image.
Operator Dashboard	Operator/Employee	Tickets	Display tickets created by the corresponding user.

Source: Research Results (2026).

The combined class, relational, and interface structures demonstrate that Tickets functioned as the central transactional element linking users, machines, spare parts, status information, and reporting outputs. The source architecture associated each ticket with user_id and barang_id, while quantity and spare-part information supported corrective handling and work-history documentation. Export functionality subsequently retrieved ticket data using selected filters and generated PDF output after access validation through middleware and controller logic. This arrangement indicates that the presentation layer, role-based access rules, database entities, and document-generation process were integrated into one continuous information flow rather than implemented as disconnected modules. Such integration is relevant to occupational safety information systems because the value of digital reporting depends on maintaining the relationship between a reported condition, the responsible user,

the affected operational object, and the documented follow-up (Hardiani, 2025). The database architecture therefore supported both immediate transaction processing and later retrieval of historical OHS and maintenance information.

Functional Validation and Implementation Performance

Functional validation was conducted through Black Box Testing to determine whether the implemented application produced the expected responses under both valid and invalid input conditions. The testing scope covered six functional areas documented in the source system: login, ticket input, user input, machine input, spare-part input, and management of all tickets. Across these areas, the recorded test matrix contained 20 individual scenarios, comprising three login scenarios, three ticket-input scenarios, three user-input scenarios, three machine-input scenarios, three spare-part scenarios, and five ticket-management scenarios. Each scenario compared the submitted condition with an explicitly defined expected output and classified the result as valid when the observed behavior corresponded to that expectation. This approach is consistent with Black Box Testing because functional correctness is assessed from input–output behavior without requiring examination of the internal application code (Jibril et al., 2024). The resulting validation structure also reflects the importance of systematic functional verification in web-based ticketing applications before deployment to operational users (Karimah et al., 2024).

Table 11. Login Page Testing

No.	Test Scenario	Test Case	Expected Result	Test Result	Conclusion
1	Correct username and password	Correct Username & Password	Successfully enters the system	As Expected	Valid
2	Entering incorrect login data	Correct Username & Incorrect Password	Message “Invalid username or password” appears	As Expected	Valid
3	Username and password are not entered	Username & Password fields are empty	Message “please fill out this field” appears	As Expected	Valid

Source: Research Results (2026).

Table 12. Ticket Input Testing

No.	Test Scenario	Test Case	Expected Result	Test Result	Conclusion
1	Ticket input form is completed correctly	Complete ticket data	Ticket is successfully created and appears in the list	As Expected	Valid
2	Ticket input form is incomplete	Some data are not entered	Ticket is not created and validation is displayed	As Expected	Valid
3	Ticket input form is empty	Empty data	Ticket is not created and validation is displayed	As Expected	Valid

Source: Research Results (2026).

The authentication results demonstrate that the application distinguished between three materially different conditions: valid credentials, partially invalid credentials, and empty credentials. All three cases produced the expected system response, indicating that the login function operated consistently across successful authentication and error-handling conditions. Ticket-input testing showed the same pattern because complete data generated a new ticket, whereas incomplete or empty records were rejected and returned a validation response. This behavior is operationally important because a ticketing system used for occupational safety and maintenance reporting requires sufficiently complete information before a record can enter the handling workflow. Digital reporting systems are

more useful when structured validation supports data completeness and reduces the risk that inadequately specified reports become part of the operational record (Ningtyas et al., 2024). The recorded results therefore indicate that the system implemented preventive input control at two critical points: initial access and report creation.

Validation was subsequently extended to the administrative master-data functions for users and machines, both of which were tested using complete, incomplete, and empty input conditions. The user-management tests verified that a complete record could be created and stored, whereas incomplete or empty forms did not generate new user records. Machine-input testing followed the same three-condition structure, with valid machine data accepted and invalid or absent values rejected through application validation. The consistency of these outcomes is relevant because user and machine records constitute master data that are referenced by subsequent ticket transactions. Errors at this level could propagate into reporting, role assignment, or machine identification if incorrect records were accepted into the database. Structured information management is closely associated with operational efficiency when access data and supporting management information are kept consistent within the same digital environment (Nugrahayani & Bunahri, 2025).

Table 13. User Input Testing

No.	Test Scenario	Test Case	Expected Result	Test Result	Conclusion
1	User input form is completed correctly	Complete user data	Data are successfully created and appear in the user list	As Expected	Valid
2	User input form is incomplete	Some data are not entered	User is not created and validation is displayed	As Expected	Valid
3	User input form is empty	Empty data	User is not created and validation is displayed	As Expected	Valid

Source: Research Results (2026).

Table 14. Machine Input Testing

No.	Test Scenario	Test Case	Expected Result	Test Result	Conclusion
1	Machine input form is completed correctly	Complete machine data	Machine data are successfully created and appear in the list	As Expected	Valid
2	Machine input form is incomplete	Some data are not entered	Data are not created and validation is displayed	As Expected	Valid
3	Machine input form is empty	Empty data	Data are not created and validation is displayed	As Expected	Valid

Source: Research Results (2026).

Spare-part input was evaluated using the same validation logic applied to users and machines, allowing direct comparison of system behavior across multiple administrative forms. Complete spare-part records were accepted and stored, while incomplete and empty records were rejected with the expected validation behavior. More critically, the Manage All Tickets function was examined through five scenarios covering status transitions to Pending, Approved, In Progress, and Resolved, together with ticket deletion. Every recorded scenario generated the expected output and received a Valid conclusion, indicating that both status control and administrative deletion operated according to the functional specification. This capability is significant because explicit status progression gives safety-related reports an observable processing state rather than leaving users uncertain about whether action has occurred. Previous work on digital hazard-reporting systems similarly emphasizes that the

effectiveness of reporting depends on connecting submitted information with visible follow-up and monitoring processes (Wardani, 2025).

Table 15. Spare-Part Input Testing

No.	Test Scenario	Test Case	Expected Result	Test Result	Conclusion
1	Spare-part input form is completed correctly	Complete spare-part data	Spare-part data are successfully created and appear in the list	As Expected	Valid
2	Spare-part input form is incomplete	Some data are not entered	Data are not created and validation is displayed	As Expected	Valid
3	Spare-part input form is empty	Empty data	Data are not created and validation is displayed	As Expected	Valid

Source: Research Results (2026).

Table 16. Manage All Tickets Testing

No.	Test Scenario	Test Case	Expected Result	Test Result	Conclusion
1	Change ticket status	Status becomes Pending	Ticket status is successfully changed	As Expected	Valid
2	Change ticket status	Status becomes Approved	Ticket status is successfully changed	As Expected	Valid
3	Change ticket status	Status becomes In Progress	Ticket status is successfully changed	As Expected	Valid
4	Change ticket status	Status becomes Resolved	Ticket status is successfully changed	As Expected	Valid
5	Delete ticket	Delete one ticket	Ticket is successfully deleted	As Expected	Valid

Source: Research Results (2026).

Across the six functional groups, the source data document 20 successful test cases out of 20 scenarios, corresponding to a functional pass rate of 100% for the scenarios actually tested. This figure should be interpreted specifically as conformance with the documented Black Box test cases rather than as evidence that the application is free from every possible software defect or performance limitation. The result indicates that the implemented functions performed consistently for authentication, form validation, master-data creation, ticket-status changes, and ticket deletion under the conditions recorded in the test matrix. Such functional conformity is important for OHS-oriented systems because reporting reliability depends on the application preserving submitted information and processing states without unexpected functional failure. More technologically advanced safety-reporting architectures have explored privacy-preserving and secure reporting mechanisms, illustrating the broader importance of trustworthy digital infrastructure for occupational risk information (Nyato et al., 2024). Within the present implementation, however, the evidence supports functional correctness of the defined operational workflow rather than advanced security, scalability, or predictive-performance claims.

The implementation also changed the practical organization of reporting by replacing dispersed manual maintenance and OHS reporting activities with an integrated web-based workflow in which operators submit reports and Technician/Admin users manage status, work history, machine information, spare-part records, and printable documentation. The Waterfall development structure supported a sequential transition from requirements analysis and system design to implementation, testing, deployment, and maintenance, while the Laravel-based web architecture provided the framework for the resulting operational application. The source material nevertheless identifies four important limitations: the initial system remains web-based without a native mobile application, notifications are not yet integrated with external services, data storage remains local or internal, and

advanced performance-analysis modules are not included. These limitations mean that the present results demonstrate functional feasibility and workflow integration rather than comprehensive enterprise-scale maturity. Future development could extend external notifications, distributed or cloud-based storage, advanced analytics, and more sophisticated monitoring capabilities while preserving the validated ticketing workflow already implemented. The current version therefore provides a functional operational foundation for structured OHS incident-prevention reporting at PT Takaki Indoroti Prima, while leaving clearly identifiable technical directions for subsequent development.

CONCLUSION

The implementation of the web-based ticketing and monitoring information system successfully transformed previously manual maintenance and occupational safety reporting activities at PT Takaki Indoroti Prima into a more integrated, traceable, and systematically documented digital workflow. The Waterfall model provided a structured development sequence encompassing requirements identification, system design, implementation, functional testing, deployment, and maintenance, allowing each stage to be completed in accordance with predefined system needs. The resulting application facilitated ticket submission, monitoring of handling progress, management of work history, machine and spare-part administration, and report generation within a centralized web environment. Laravel 13 supported the technical implementation through an organized web-application structure that connected authentication, role-based access, ticket processing, database interaction, and reporting functions. Functional validation through Black Box Testing on login, ticket input, user input, machine input, spare-part input, and ticket-management functions showed that the documented scenarios produced outputs consistent with the expected results and were classified as valid. These findings indicate that the system provides a functional foundation for improving operational efficiency, work-monitoring transparency, documentation consistency, and accountability in maintenance and OHS-related reporting activities.

Further development should focus on extending the system beyond its current implementation limits while preserving the validated workflow and database structure established in this study. Wider operational deployment and testing are required to evaluate system performance under more diverse users, workloads, and real working conditions, particularly with respect to response time, reliability, usability, and scalability. A mobile-oriented version could improve field accessibility by enabling technicians and operators to submit, monitor, and update tickets directly from operational locations without relying solely on desktop or browser-based access. Integration with inventory and corporate asset-management databases would strengthen the relationship between maintenance records, machine history, spare-part consumption, and corrective actions, creating a more comprehensive organizational information base. Periodic user training and technical assistance should also accompany broader implementation so that role-based functions, reporting procedures, and monitoring features are used consistently across departments. Future versions should additionally incorporate external notification services, distributed or cloud-based storage, and advanced performance analytics to support faster escalation, richer management evaluation, and more proactive occupational safety and maintenance decision-making.

REFERENCES

- Anggraini, D. P., Ambarwati, R., Dedy, D., & Toirxonovna, A. M. (2024). Workplace hazard identification through near-miss reports: A social network analysis. *Indonesian Journal of Information Systems*, 7(1), 70–83. <https://doi.org/10.24002/ijis.v7i1.8891>
- Apriyanti, A., & Ramli, M. (2025). Pengembangan website Program Studi Teknik Informatika berbasis Progressive Web App (PWA). *Jurnal Ilmu Pengetahuan dan Teknologi*, 9(1), 50–57. <https://doi.org/10.31543/jii.v9i1.383>
- Bahrudin, R. M., Ridwan, M., & Darmojo, H. S. (2020). Penerapan helpdesk ticketing system dalam penanganan keluhan penggunaan sistem informasi berbasis web. *Jurnal Teknik Informatika UNIS*, 7(1), 71–82. <https://doi.org/10.33592/jutis.Vol7.Iss1.147>
- Nugrahayani, T., & Bunahri, R. R. (2025). Pengaruh pendidikan dan pelatihan teknologi informasi, keselamatan dan kesehatan kerja (K3), dan inovasi sistem manajemen terhadap efisiensi operasional. *JDMP (Jurnal Dinamika Manajemen Pendidikan)*, 9(2), 178–191. <https://doi.org/10.26740/jdmp.v9n2.p178-191>

- Dirmanthara, D. F., Nugraha, E. S., Prasetya, T., Ali, I., & Kaslani. (2022). Rancang bangun sistem informasi ticketing berbasis website pada STF Muhammadiyah Cirebon. *REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer*, 6(4), 674–685. <https://doi.org/10.33395/remik.v6i4.11780>
- Hardiani, P. W. B. (2025). Web-based information prototype system for occupational safety and health risk assessment in universities. *Indonesian Journal of Adult and Community Education*, 7(1). <https://ejournal.upi.edu/index.php/IJACE/article/view/88928>
- Herdiansyah, C. D. C., Safaruddin, S., & Ismi, A. (2023). Pengembangan sistem informasi helpdesk ticketing sistem berbasis web pada PT Semen Baturaja (Persero) Tbk. *Jurnal Multidisipliner Bharasumba*, 2(2), 132–147. <https://doi.org/10.62668/bharasumba.v2i02.564>
- Hidayati, A. T., Widyantoro, A. E., & Ramadhani, H. J. (2023). Perancangan sistem informasi wirausaha mahasiswa (Siwirma) berbasis web dengan Unified Modelling Language (UML). *Jurnal Penelitian Rumpun Ilmu Teknik*, 2(4), 86–107. <https://doi.org/10.55606/juprit.v2i4.2906>
- Huzzaemah, E. (2024). Rancang bangun sistem informasi berbasis website menggunakan Laravel 11 di Dinas Komunikasi dan Informatika Kabupaten Purwakarta. *Jurnal Rekayasa Informatika*, 1(2), 44–50. <https://ejournal.pei.ac.id/index.php/JRIT/article/view/213>
- Jibril, M., Zulrahmadi, & Amin, M. (2024). Pengujian sistem informasi e-modul pada SMPN 1 Tempuling menggunakan black box testing. *Jurnal Perangkat Lunak*, 6(2), 327–332. <https://doi.org/10.32520/jupel.v6i2.3326>
- Karimah, M., Eriana, E. S., & Haerudin, H. (2024). Implementasi metode Extreme Programming pada pembuatan e-ticketing berbasis web. *Spectrum: Multidisciplinary Journal*, 1(2), 117–125. <https://journals.sanusantara.com/index.php/spectrum/article/view/83>
- Melani, L. L., Ilhadi, V., & Muthmainnah. (2025). Perancangan dan implementasi sistem informasi absensi karyawan K3 PLN berbasis website. *Sisfo: Jurnal Ilmiah Sistem Informasi*, 9(2), 12–27. <https://doi.org/10.29103/sisfo.v9i2.24996>
- Ningtyas, P. N., Kurniasih, D., & Arninputranto, W. (2024). Perancangan sistem informasi pelaporan statistik K3 berbasis website. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 9(1), 44–51. <https://ejournal.bsi.ac.id/index.php/ijcit/article/view/16411>
- Nouvel, A., Sutrisno, & Indriani, R. (2021). Informasi surat masuk dan surat keluar berbasis web pada suatu instansi. *Indonesian Journal on Software Engineering*, 7(1), 55–62. <https://ejournal.bsi.ac.id/mo/index.php/ijse/article/view/10181>
- Nyato, E. J., Kimito, E., Yang, J., Lee, D., & Lee, D. (2024). Blockchain-integrated zero-knowledge proof system for privacy-preserving near-miss reporting in construction projects. *Automation in Construction*, 168, 105825. <https://doi.org/10.1016/j.autcon.2024.105825>
- Pratiwi, E. L., Sabrina, E., & Ramadhani, M. K. (2025). Implementation of web technology in occupational health and safety management in the electrical distribution sector. *KnE Social Sciences*, 376–386. <https://doi.org/10.18502/kss.v10i11.18756>
- Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2023). Analisis teknik Entity-Relationship Diagram dalam perancangan database. *Jurnal Ekonomi Manajemen dan Bisnis (JEMB)*, 1(2), 143–147. <https://doi.org/10.47233/jemb.v1i2.533>
- Salsabila, A. M., & Wahyuningsih, A. S. (2023). Efektivitas penerapan aplikasi pelaporan bahaya sebagai upaya pencegahan kecelakaan kerja pada PT X. *Syntax Literate: Jurnal Ilmiah Indonesia*, 8(5). <https://doi.org/10.36418/syntax-literate.v8i5.11810>
- Sinlae, F., Irwanda, E., Maulana, Z., & Syahputra, V. E. (2024). Penggunaan framework Laravel dalam membangun aplikasi website berbasis PHP. *Jurnal Siber Multi Disiplin*, 2(2), 119–132. <https://doi.org/10.38035/jsmd.v2i2.186>
- Wahyudi, A. F. S., & Heksaputra, D. (2023). Pengembangan aplikasi Penilaian Outcome-Based Education (OBE) berbasis website dengan metode Waterfall. *INSERT: Information System and Emerging Technology Journal*, 4(2), 86–94. <https://doi.org/10.23887/insert.v4i2.6528>
- Wardani, A. P. A. M. K. (2025). Inovasi sistem pelaporan dan pengawasan kinerja kontraktor berbasis web dalam mendukung sistem manajemen K3 di perusahaan pupuk. *Conference on Safety Engineering and Its Application*, 8(1). <https://journal.ppns.ac.id/index.php/seminarK3PPNS/article/view/2840>
- Yamamoto, A., Nagata, T., Odagami, K., Adi, N. P., Nagata, M., & Mori, K. (2024). Relationship between a company's adequate response to near-misses and occupational accidents: A 1-year

prospective cohort study. *Journal of Occupational Health*, 66(1), uiae053.
<https://doi.org/10.1093/joccuh/uiae053>