Copyright © The Author(s) 2027
Jurnal Artificial, Vol 5 No 1, April 2027

Development and Functional Testing of a Web-Based Visitor Management System Using Rapid Application Development at PKBM INTAN South Tangerang

Khairudin [1*], Roeslan Djutalov [2], Agus Wiranto [3], Dian Yulianti [4], Nur Annisahaq [5]

[1, 2, 3, 4, 5] Universitas Pamulang, Indonesia

* dosen02591@unpam.ac.id

Abstract

Institutions providing server colocation and maintenance services require integrated visitor identification, coordination with designated personnel, and visit documentation. This study aimed to develop and functionally test a web-based visitor management system at PKBM INTAN, South Tangerang. Development followed Rapid Application Development through requirements planning, user design, construction, and cutover. User involvement comprised ten participants in requirements discussions, visit workflow discussions, and design reviews. The system was developed using PHP and Laravel to integrate regular registration, returning visitor registration, scheduled visits, photographic documentation, and visitor pass provision. Functional testing compared inputs, expected outcomes, actual outcomes, and conformity status across ten scenarios. All scenarios in the testing documentation were reported as conforming to their expected outcomes, covering visitor workflows, user authentication, employee management, visitor list presentation, schedule management, and profile updates. However, the photograph and visitor pass testing records did not detail the verification of photograph storage or completed pass printing. Error conditions and unauthorized access testing were also undocumented. The findings indicate that the system integrates visitor management functions within the scope of the reported scenarios. The study contributes an application of RAD to the development of a system accommodating three visitor workflows within a single application. Further evaluation is required to assess error handling, security, privacy, usability, and operational benefits.

Keywords: *Functional Testing, Rapid Application Development, Visit Management, Visitor Management System, Web Application*

Introduction

Visitor management is part of institutional operations that connects identity recording, internal coordination, and service delivery. Information about who is visiting, the purpose of the visit, the intended contact, and the time of arrival helps staff understand the activities taking place. These records can also be used to trace previous visits when necessary. Visitor management should therefore be understood as a process linking administrative information to staff actions, rather than merely completing an arrival form. A clear understanding of this relationship provides the basis for designing a system that addresses institutional needs (Halaho et al., 2025; Rayhan, 2026).

These management requirements vary according to the services provided and the purposes of visits. At institutions providing server colocation and maintenance services,

https://doi.org/10.54065/artificial.5.1.2027.1984

visits involve activities concerning clients' equipment or services. The visit process requires coordination with designated internal personnel so that staff understand the purpose of arrival and identify the person accompanying the visitor. Visitor registration forms part of a service sequence that includes checking information and providing visitor identification. A supporting system must accommodate these connections so that the recorded information can be used during subsequent service stages.

Web application development provides a means of connecting these processes through interfaces accessed using a browser (Aprisalni et al., 2025). Forms, scheduling information, visitor records, and administrative functions can be incorporated within the same application environment. This design enables users to interact with the available functions according to their needs and permissions. However, the availability of an application does not, by itself, demonstrate faster or more accurate service delivery. Assessing these benefits requires appropriate measurements, whereas the initial development stage must first establish that the functions reflect operational requirements (Ardhana et al., 2022).

Research on visitor management systems has demonstrated various approaches to supporting record keeping and service delivery. The development of a system encompassing visitor identity management and pass provision (Oktaviandri & Foong, 2019; Husodo et al., 2024). A visitor management system supported by mobile devices (Rivera, 2021). Both studies establish that managing visit information is an existing area of software application. Consequently, the contribution of subsequent research should be explained through the problem addressed, the design decisions made, and the available evaluation evidence, so that its position does not depend solely on the existence of a digital application.

In the context of this study, PKBM INTAN, South Tangerang, is described as providing server colocation and maintenance services. The visits addressed by the system involve clients undertaking activities related to servers, including equipment inspection. Accordingly, visitor management is discussed within this service context. This explanation is necessary to clarify the relationship among the institution's name, its server services, and its registration requirements. The scope of the study remains visit information management, while the technical activities of server inspection constitute the operational context underlying the need for the system.

The visit workflow involves coordination with the designated person in charge (PIC), completion of visit information, photographic documentation, staff validation, and visitor pass provision. Following these steps, visitors are accompanied by the PIC during server-related activities. This sequence shows that registration data are used by more than one operational actor. Information entered by visitors must be available to the personnel who check the visit and prepare the pass. The system design must therefore consider the relationships among activities, including the points at which information is transferred from visitors to internal personnel (Melendres & Aranda, 2024).

Variation in registration requirements is another consideration in development. Regular visitors enter information through a form, whereas previously registered visitors require retrieval of their earlier records. Scheduled visits require checking information prepared before arrival. These three conditions have related record-keeping purposes, but their starting points of interaction differ. A design accommodating these differences must

provide pathways that users can understand while maintaining connected records. Workflow integration is therefore a design issue that must be explained in relation to the requirements of the study site.

For returning visitor registration, retrieving previous information helps connect a new visit with existing records. Nevertheless, the system requires rules specifying the data used for searching and the actions taken when information is found or cannot be found. For scheduled visits, the relationship between identity and scheduling information also requires examination. These conditions show that development extends beyond providing form pages. Functions must be designed together with their conditions of use, while implementation conformity must be examined through scenarios representing user actions and system responses.

Specific operational requirements make user involvement an essential part of development. Users can explain the service sequence, the information required, and the difficulties encountered when using an interface design. The relationship between user involvement and system success and highlighted the need to consider the context of participation. In research reporting, this involvement should be described through the activities undertaken and the relationship between feedback and development decisions (Akello & Nabukenya, 2024; Cellens, 2023). Participant numbers indicate the scope of participation, but cannot directly establish satisfaction with or acceptance of the system.

Rapid Application Development (RAD) provides a development approach that incorporates requirements discussions and collaborative design reviews with users into application construction. The RAD implementation through an empirical review (Daraghmi & Daraghmi, 2022). While the previous studied examined its use in large and complex systems development (Saeeda et al., 2023). This literature provides a basis for considering how well the approach suits the project context. The use of RAD in research should be demonstrated through actual activities at each stage, so that naming the method is accompanied by an explanation of its implementation.

In this study, RAD was applied through requirements planning, user design, construction, and cutover. These stages provided a framework for connecting visit requirements with the application's design and functions. The documentation identifies ten participants involved in requirements discussions, visit workflow discussions, and design reviews. This information supports reporting user involvement within the documented scope. A more detailed assessment of the influence of participation would require traceability among feedback, prototype changes, and implementation decisions. Distinguishing these aspects helps maintain an accurate account of the development process. In addition to documenting development, evaluation is necessary to assess whether system functions produce appropriate responses. Mapped research on web application testing (Aydos et al., 2022). While systematically reviewed testing approaches. These studies establish that web application testing encompasses various objectives and techniques. The chosen evaluation must reflect the research objectives and the available evidence. In visitor management system development, functional testing can examine the relationship between user inputs and system outcomes under particular scenarios.

Accurate interpretation of testing depends on clear assessment criteria. The previous studied discussed the problem of determining whether observed system behavior corresponds to correct behavior (Wang et al., 2022; Vos et al., 2021). In reporting practice,

inputs, expected outcomes, and actual outcomes should be presented in a connected manner. Conformity status becomes meaningful when readers understand the actions performed and the outputs examined. The availability of an interface, the ability to perform an action, and successful storage of its results must therefore be distinguished when the evidence differs in its level of detail.

Testing normal use pathways should also be distinguished from testing error conditions. Login using valid credentials does not explain the response to an incorrect password, just as a successful data search does not explain what happens when records are unavailable. The previous studied demonstrated the importance of invalid-input testing for identifying weaknesses in web application validation (Fadlalla & Elshoush, 2023). Within a visitor management system, empty fields, incorrectly formatted email addresses, and missing records are relevant conditions to examine. Testing coverage determines the limits of the conclusions that can be drawn.

Camera use and visitor pass provision introduce further requirements for checking sequences of functions. A camera page demonstrates that a photo capture facility is available, but testing can extend to verifying that the file is stored and accessible to authorized personnel. Likewise, a printable pass page must be distinguished from evidence that printing has been completed. Device failures or camera permission issues may also affect the continuation of registration. Testing should therefore be designed according to the purpose of each function and the outcomes that actually need to be demonstrated.

Security is another aspect associated with the use of a system by multiple parties. Provided a conceptual basis for role-based access control. In a visitor management system, permissions must be allocated so that actions involving information correspond to users' responsibilities. Administrators, employees, and visitors have different interaction requirements. However, separate menus alone are insufficient to establish effective access restrictions. Examination must consider whether prohibited actions are actually rejected, including attempts to access functions or data beyond a user's authority.

The management of identity information and photographs also requires attention to privacy. Relevant requirements include collection purposes, authorized parties, retention periods, and deletion arrangements. Positioned security requirements engineering as an important component of software development (Zagita & Raharjo, 2023). In this study, explanations of protective mechanisms must refer to documented implementation and policies. The use of a particular framework cannot directly demonstrate that all protections have been implemented. Separating features, actual configurations, and evaluation results helps maintain accurate claims.

Based on this literature, the gap addressed by the study concerns documenting the relationship among operational requirements for server services, RAD implementation, integration of registration variants, and the conformity of the resulting functions. The study does not claim that visitor registration or workflow integration is an entirely new idea. Its focus is explaining how the study site's requirements were translated into a system and how the results were evaluated. This position provides a basis for an evidence-based contribution while allowing comparison with developments in other contexts.

This study aims to identify visitor management requirements at PKBM INTAN, South Tangerang, develop a web-based system using RAD, and evaluate functional conformity through documented testing scenarios. The research questions address how requirements

are translated into functions, how regular registration, returning visitor registration, and scheduled visits are integrated, and the extent to which testing results support implementation conformity. The contribution consists of development documentation connecting the operational context with system functions and the interpretation of testing results. Conclusions are restricted to the available evidence, while evaluations of security, usability, and operational benefits remain requirements for subsequent development. Within this scope, the development results can be discussed proportionately without turning documentation limitations into claims of success unsupported by measurement.

Method

Research Design and Context With Requirements Analysis

This study is a system development study situated at PKBM Insan Satya Guna (INTAN), South Tangerang. The research object is the process of registering, recording, and monitoring visitor information in a data center environment. The institution is described as providing server colocation and maintenance services to approximately 150 clients from various regions. This figure provides service context; it is neither the number of development participants nor the testing sample size. The intended system users are PICs or client representatives who make periodic visits for server inspection.

Before development, visitors were recorded manually using physical attendance records. The initial situation involved difficulties in organizing and monitoring information and the need for space to store these records. These conditions provided the basis for developing digital record keeping. The study did not quantify the extent of these difficulties, service time, or storage costs; this information was therefore used as the background to operational requirements.

The actual workflow begins with visit confirmation with the PIC, followed by entering visitor data, taking a photograph, receptionist validation, printing a visitor pass, and accompaniment to the server room. The PIC is described as a client representative in the visit context and as a party involved in coordination and accompaniment in the procedural account. The documentation does not detail the distinction between the client's PIC and the internal accompanying PIC. Accordingly, the study does not define them as the same position or account type. Development concerns visit information, rather than evaluation of server inspection or physical security of the server room.

Information collection is described as observation and analysis of visitor administration processes, supplemented by requirements discussions and design reviews. Observation focused on registration, recording, storage, record monitoring, and the use of administrative information. Subsequent development distinguished regular visitors, returning visitors, and scheduled visitors. All three require personal details and visit information, although their registration starting points differ.

The managed data include identity details, visit purposes and information, photographs, validation status, and visitor passes. Supporting information includes employees, accounts, schedules, and profiles. Requirements analysis connected these data with login, visitor forms, visitor management, schedule management, employee management, visitor lists, and profile functions. The case analysis focused on the relationship between context and implementation; reporting considered clarity of scope and evidential limitations.

RAD Stages and User Involvement

Development followed requirements planning, user design, construction, and cutover. RAD provided a framework for translating requirements into application models and functions. The RAD literature highlights the importance of understanding development activities within their implementation context. This study used the framework to explain the process, without comparing development speed or costs with other methods.

During requirements planning, activities included reviewing manual record keeping, identifying visit information and the parties involved, and defining user permissions. Differences in visitor conditions were translated into three registration requirements. The outputs were functional requirements and access allocations that guided design.

During user design, requirements were translated into interaction workflows, system models, data relationships, and interfaces. UML was used to explain interactions and activities, while an ERD was used to represent data relationships. The design also included sequence diagrams, flowcharts, table relationships, and interfaces identified in the documentation. These models guided implementation; the login and visitor form activity diagrams illustrate the design. Ten participants took part in requirements discussions, workflow discussions, and design reviews. The discussion outcomes informed the development of login, forms, photographs, passes, visitor data, and scheduling functions. The documentation does not detail participants' roles, meeting frequency, feedback instruments, or changes in each version. Involvement is reported according to the recorded activities; participant numbers are not treated as tester numbers or satisfaction indicators. Distinguishing participation from evaluation is consistent the previous studied attention to user involvement (Mashkoor et al., 2023).

During construction, the application was developed using PHP and Laravel with a database for visitor and administrative information. Regular registration provides information entry, returning visitor registration uses email addresses to retrieve records, and scheduled visitor registration uses information prepared by employees. Photo facilities, passes, and administrative functions connect these pathways. Software versions and infrastructure details were not reported, so additional specifications are not assumed. During cutover, the application was implemented and examined using black-box scenarios. Actual outcomes were compared with expected outcomes. The documentation does not detail migration of manual records, training, the duration of operational use, or post-implementation monitoring. This stage therefore supports an account of implementation and functional testing without establishing successful long-term use.

System Modeling and User Permissions

The system has three roles: administrator, employee, and visitor. Administrators have access to system management, including master data, visitors, schedules, and user information. Employees view and confirm visitor data and prepare scheduled visits. Visitors interact with forms and photo facilities. Receptionists and PICs are operational actors identified in the procedures; the documentation is not used to introduce additional account roles.

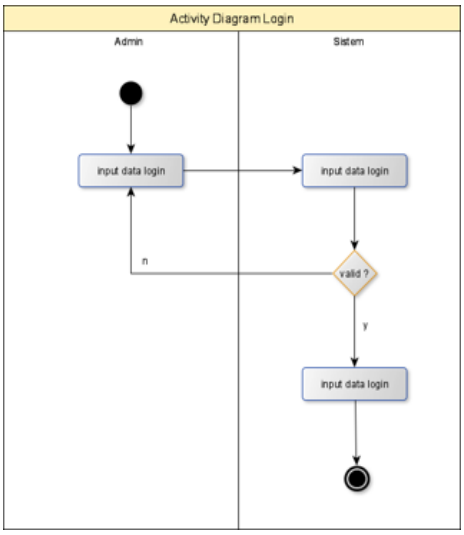


Figure 1. User Authentication Activity Diagram

Figure 1 illustrates the administrator and employee authentication workflow through checking account information against registered records. Figure 2 illustrates the visitor form completion workflow, encompassing personal details, photo capture, and data storage according to the system design.

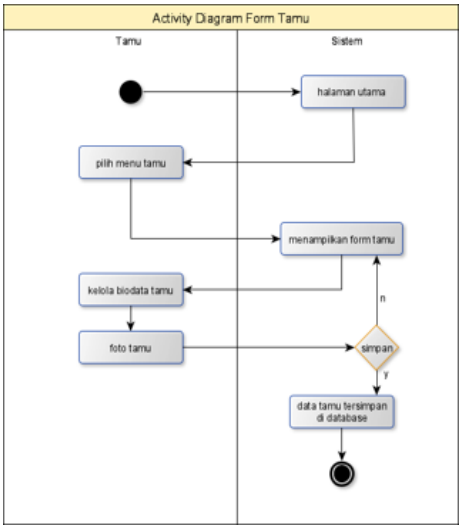


Figure 2. Visitor Form Completion Activity Diagram

The use cases identified in the documentation include Login, Manage Scheduled Visitors, Manage Visitors, Manage Employees, Visitor List, Manage Profile, and Visitor Form. Database modeling covers visitor records, employees, accounts, schedules, and visit activities. The documentation also identifies activity diagrams for employees, visitor lists, profiles, schedules, and visitor management. These models are described as construction references without reproducing models unavailable in the article draft.

Permission allocation is described at the level of the reported functions. Role-based access control provides a reference for examining the relationship between users and permitted actions (Sandhu et al., 1996). However, the study does not include access-denial testing for each role. Permission design is therefore distinguished from evidence of its protective effectiveness.

Functional Testing and Results Analysis

Evaluation used black-box testing to examine system behavior through inputs and outputs, without source code inspection as part of the reported tests. The ten scenarios in the supplementary report cover three registration pathways, photographs, passes, login, employees, visitor lists, schedule management, and profiles. Web testing research indicates that evaluation should be aligned with its objectives and scope. The procedure consisted of defining scenarios and inputs, comparing responses with expected outcomes, and recording actual outcomes and status. A conforming status means that the behavior in the testing record was reported as matching the scenario criteria. Analysis was descriptive; statistical testing was not used because the documentation did not provide repeated measurements or quantitative evaluation variables.

The clarity of actual outcomes was examined to distinguish interface availability from successful completion of actions through storage or printing. Assessment criteria are required to determine whether observed behavior is correct. The ten scenarios were not used to calculate a success percentage for the entire system, because the scenarios do not constitute a random sample of all conditions of use.

Failed login, empty or invalid inputs, missing records, camera failure, and unauthorized access do not have documented outcomes. Duplicate records are another condition requiring consideration. All these conditions are identified as requirements for further testing. Dedicated usability, acceptance, performance, scalability, and security tests were not reported. Authentication and validation mechanisms were analyzed according to the documentation, without assuming particular hashing, encryption, photo restrictions, or retention policies.

Results

System Implementation and Administrative Functions

The development produced a web application using PHP and Laravel that connects visitor registration with administrative information management. The system provides the Check In, Customer, and Customer Terjadwal pathways. All three relate to personal details, webcam facilities, and pass pages, while internal functions manage the resulting records.

Table 1. Relationship Between Operational Requirements and System Functions

Operational requirement	System function	Scope
Regular registration	Check In form	Entry and processing of visit information
Returning visitors	Email address search	Presentation of retrieved information
Scheduled visits	Schedule verification and management	Information prepared before arrival
Visual documentation	Webcam facility	Photo capture interface
Visitor identification	Visitor pass page	Presentation of a printable pass
Internal administration	Employees, visitor list, and profiles	Management of supporting information

Table 1 maps operational requirements to system functions. The mapping explains the relationship between requirements and features without adding evaluation results that are not available.

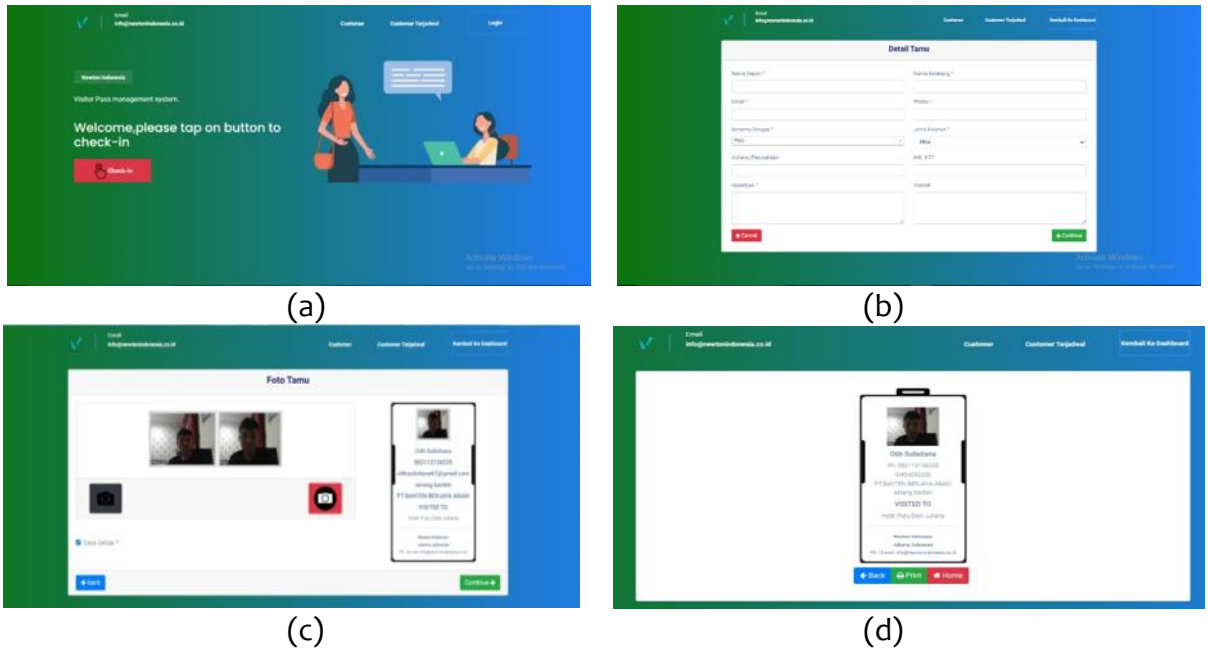
Employee management provides a form for adding records. The visitor list displays stored records. Schedule management allows information to be prepared before arrival,

while the profile provides user data updates and a password change function. List presentation supports record monitoring; it is not described as real-time tracking of visitors' locations.

Administrators manage functions according to their permissions, while employees participate in visit confirmation and preparation. The relationship between internal functions and registration allows data to be managed within one application. The study did not measure the service time, workload, or costs resulting from this relationship.

Three Registration Workflows and Visitor Documentation

Regular registration begins by selecting the Check In menu on the main page. Visitors then enter information on the details page and proceed to the webcam photo capture page. A visitor pass is displayed at the next stage. The regular registration interfaces are presented in Figure 3(a)–(d).



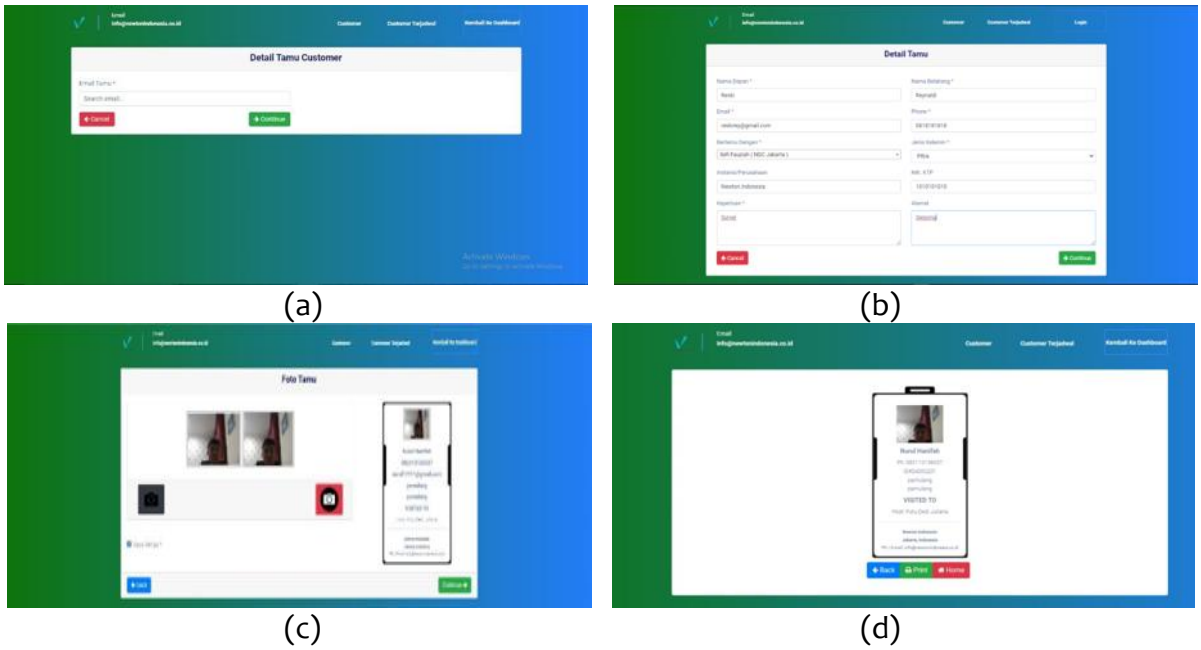
(a) (b) (c) (d)

Figure 3. Regular Registration Interfaces (Check In): (a) Visitor Main Page; (b) Visitor Details Page; (c) Webcam Photo Capture Page; and (d) Visitor Pass Display

Figure 3(a) shows the main page for selecting a registration pathway, while Figure 3(b) shows the visitor details form. Figure 3(c) provides the webcam photo capture interface, and Figure 3(d) displays the visitor pass. The four panels explain the interface sequence; successful photo storage and completed pass printing are examined separately using the testing documentation.

Returning visitors register through the Customer menu by entering a previously registered email address. When a record is found, earlier information is displayed on the details page to continue entering personal details and taking a photograph. The email entry, visitor details, photo capture, and visitor pass interfaces are presented in Figure 4(a)–(d).

ISSN 2808-7038



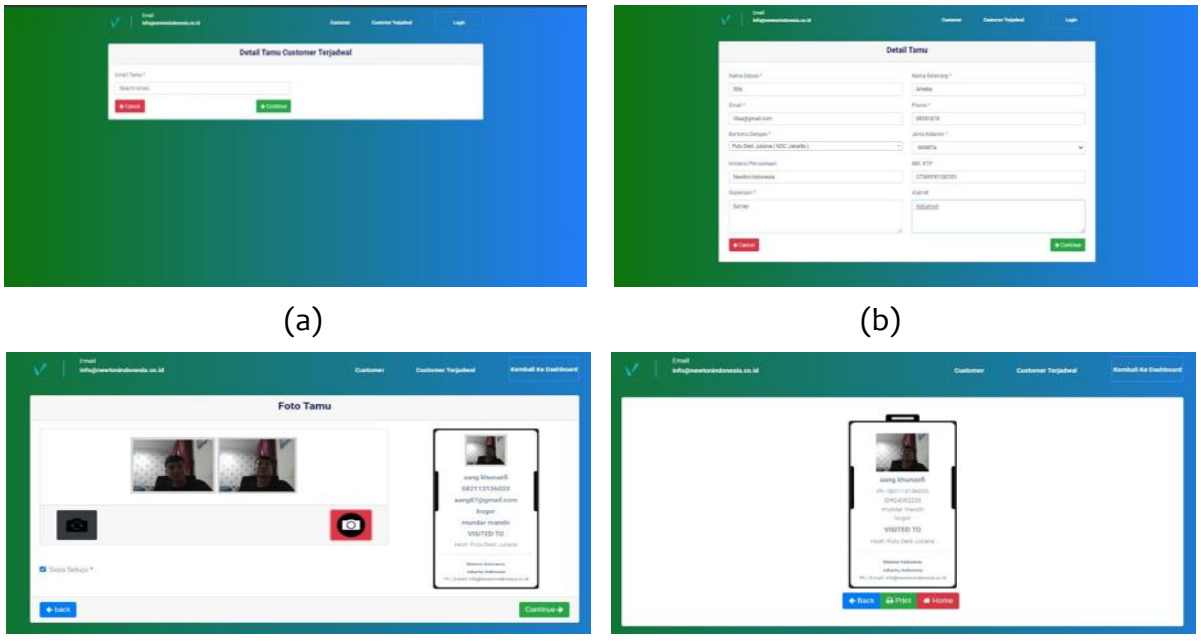
(a) (b)

(c) (d)

Figure 4. Returning Visitor Registration Interfaces (Customer): (a) Email Entry Page; (b) Visitor Details Page; (c) Webcam Photo Capture Page; and (d) Visitor Pass Display

Information retrieval connects subsequent visits with previous records. This function provides a mechanism for reusing information, but reductions in the amount of input or registration duration were not measured. The available testing results concern email addresses that were found, rather than unsuccessful searches.

In the Customer Terjadwal menu, employees prepare visitor information before arrival. Visitors then enter their email addresses to check previously registered records. The system displays visitor details and provides photo capture steps and a pass display. These interfaces are presented in Figure 5(a)–(d).



(a) (b)

(c) (d)

Figure 5. Scheduled Visitor Registration Interfaces : (a) Email Entry Page; (b) Visitor Details Page; (c) Webcam Photo Capture Page; and (d) Visitor Pass Display

Scheduled visitor verification is distinguished from internal schedule management. Verification connects arriving visitors with existing records, whereas schedule management stores information before the visit. This distinction explains why the two functions have separate scenarios in the testing results.

Testing Results for Core and Administrative Functions

Table 2 retains all ten scenarios, inputs, expected outcomes, actual outcomes, and statuses from the supplementary report. The terminology is translated consistently so that readers can distinguish the source records from the interpretation of results.

Table 2. Black-Box Testing Results

Scenario	Input	Expected outcome	Actual outcome	Status
Regular visitor registration (Check In)	Visitor data in the visitor form	Visitor data are validated and registration continues	The system matches the entered data against database records and redirects to the visitor page	Conforming
Returning visitors (Customer)	Previously registered visitor email address	Previous visitor data are displayed and the visit process can continue	If the email address is found in the database, previous records are displayed on the visitor details page to continue entering personal details and taking a photograph	Conforming
Scheduled visits	Previously registered visitor data/ email address	Scheduled visitor data are verified and the visit process continues	Scheduled visitor data can be verified and displayed on the visitor details page	Conforming
Photo capture	Visitor data and webcam	The system captures a photograph of the visitor's face	The system provides a webcam page for photo capture in the Check In, Customer, and Customer Terjadwal processes	Conforming
Visitor pass creation	Previously entered visitor data	The system generates a visitor pass	The system displays a printable visitor pass page for the administrator/receptionist	Conforming
Login	Email address, password, and role	The system matches the data against the database and redirects the user to the dashboard	The system matches the access details and redirects the user to the dashboard according to access permissions	Conforming
Employee data management	Employee menu → Add	Employee data are stored and displayed on the employee page	The add-employee fields are displayed and the record is successfully added	Conforming
Visitor list display	Visitor menu	The system displays previously entered visitor data	The system displays the visitor list according to visit records	Conforming
Scheduled visitor management	Scheduled Visitor menu	Data are stored in the database and the system redirects to the scheduled visitor menu	Data are successfully entered into the database and the system redirects to the scheduled visitor menu	Conforming
User profile	Profile menu → Edit Profile	Updated profile data are stored	Updated data are successfully stored in the database	Conforming

All scenarios were reported as conforming in the documentation. Returning visitor registration displayed information when an email address was found, while scheduled

visits displayed previously registered details. Administrative functions were reported to add employees, display lists, save schedules, and update profiles. For regular registration, the actual outcome describes matching data against the database and redirecting users to the visitor page. This account is retained. The documentation does not explain the matching rules for new visitors, so interpretation is not extended to a particular validation mechanism. For photo capture, the actual outcome describes providing a webcam page in the three pathways. For pass creation, the actual outcome describes a page that administrators or receptionists can print. The conforming status is retained as a reported result, but the descriptions do not detail successful photo storage, reopening files, or completed printing. Interface availability and successful completion of the action sequence are separated in the interpretation. Login testing used valid access details, so the result concerns authorized access. No results are available for incorrect credentials or unauthorized access. The absence of records for these conditions is not assigned a pass or fail status.

Security and Privacy Within the Scope of Evidence

The mechanisms described include authentication using email addresses, passwords, and roles; allocation of administrator, employee, and visitor access; form validation; and password changes through profiles. Photographs form part of registration and are used on visitor passes. This information identifies functions associated with access and identity processing. Password protection algorithms, field-level validation rules, photo file access restrictions, retention periods, and periodic deletion are not documented. Accordingly, the results do not establish technical mechanisms or formal privacy policies for these aspects. Login that matches account information is not used to assume any particular form of password storage. Error-condition, usability, acceptance, performance, scalability, and dedicated security testing are not available. The study's evidence covers implementation and functional conformity within the reported scenarios. This scope limits interpretation in the discussion and conclusions.

Discussion

Operational Conformity and Positioning Relative to Previous Research

Integration of the three pathways shows how variations in visitor conditions were translated into functions. Regular visitors begin by entering information, returning visitors retrieve previous information, and scheduled visitors use records prepared by employees. All three connect identification with administration within one application. This relationship is relevant to repeated visits by client representatives for server inspection. Registration provides records for service delivery and coordination, while the visitor list allows internal personnel to review information.

The transition from physical records to a database provides a means of digital organization. Nevertheless, the study did not measure whether this change reduced storage space, service time, errors, or overall workload. The previous studied demonstrated that identity and visit information management have been developed digitally (Oktaviandri & Foong, 2019; Rivera, 2021). This study follows that direction and applies it to the requirements of the study site. Integration of the three pathways is positioned as a contextual design decision. No direct comparison was conducted to establish superiority over other systems.

The implementation contribution can be examined through the relationship among requirements, models, interfaces, and testing results. The application extends beyond a digital form because it provides record retrieval and schedule preparation. Nevertheless, feature completeness is insufficient to establish effectiveness. Benefit claims require observation of actual use and measurable indicators.

RAD Implementation and the Meaning of user Involvement

The RAD stages connected requirements with design and implementation. Discussions with ten participants demonstrate participation in requirements definition and design review. These activities are relevant to understanding the service sequence and the information used by staff. The RAD literature provides a basis for examining the approach within its development context. In this case, the evidence explains the process followed but does not provide duration or cost comparisons. The findings therefore demonstrate RAD implementation without establishing its superiority over Waterfall, prototyping, or other approaches.

The positioned user involvement as a relationship that should be understood through its forms of participation. Ten participants in discussion activities do not automatically constitute ten testers. The absence of a satisfaction instrument also limits assessment of system acceptance. Reporting separates development participation from evaluation of use. The traceability of prototype changes requires further strengthening. Records connecting feedback, decisions, and interface changes would show how participation influenced the product. Such documentation would distinguish initial designs from revised outcomes without relying on participant numbers as the sole evidence of involvement.

Testing Interpretation with Security and Privacy Considerations

The ten scenarios reported as conforming support the functions under the documented conditions. Their scope encompasses registration pathways and administrative functions. However, testing of all possible inputs or combinations of user actions is not available. These statuses cannot be treated as a guarantee of comprehensive reliability. The previous studied discussed the need for criteria to determine correct behavior (Kamaraj et al., 2023). In this case, the level of detail in actual outcomes requires attention: a webcam page demonstrates interface availability, whereas evidence of storage requires checking the file and its connection to a visitor record. A pass page demonstrates data presentation, whereas completed printing requires a separate action outcome. This distinction preserves the source statuses while avoiding overinterpretation.

Invalid-input testing is important for examining validation weaknesses. Empty fields, incorrect email formats, missing records, duplicates, and camera failure affect the continuation of registration. Unauthorized access testing is also required to assess role restrictions. These tests should record inputs, expected responses, actual outcomes, and statuses at a level of detail that can be examined.

Role allocation provides a basis for regulating permissions, consistent with the concept described by (Sandhu et al., 1996). However, the availability of menus corresponding to roles does not prove that direct requests for prohibited functions or files are rejected. Effectiveness should be evaluated for both permitted and prohibited actions. Administrators, employees, and visitors require checks appropriate to their responsibilities. The emphasized security requirements as part of development. Identity

details and photographs require provisions governing collection purposes, access, storage, and deletion. Privacy research also provides context for the need to incorporate protection into software development (Herwanto et al., 2024). In this study, detailed configurations and provisions are unavailable; this limits the conclusions without proving that protection is absent. The use of Laravel is not treated as a substitute for implementation evidence.

Contributions and Limitations with Implications for Further Development

The study provides development documentation connecting visitor requirements for server services with web application functions. The three pathways accommodate registration variants within one environment, accompanied by identification and administrative functions. This contribution is contextual and should be assessed through the relationship between requirements and the available implementation. In practice, the function mapping can inform reviews of procedures and improvement priorities. Schedule management relates to visit preparation, record retrieval relates to repeat arrivals, and lists relate to record review. Staff usability and service impacts require evaluation of actual use.

Limitations include participation details, prototype changes, testing coverage, and verification of photographs and passes. Dedicated acceptance, usability, performance, scalability, and security evaluations are unavailable. The case also concerns a single institution; application at other sites requires adjustments to workflows and permissions. Additional evaluation should be selected according to the claims to be demonstrated, rather than expanding claims without measurement. The next priorities are completing core function testing through storage and printing, testing error conditions, and examining role-based access. Security documentation and data provisions should be linked to actual implementation. Subsequent evaluation can measure registration time, errors, user experience, or system load according to its objectives. Mobile access development, including an Android application, may be considered based on user requirements and accompanied by functional and nonfunctional testing.

Conclusion

This study developed a web-based visitor management system at PKBM Insan Satya Guna (INTAN), South Tangerang, using PHP and Laravel with the Rapid Application Development approach. The system accommodates regular registration, returning visitor registration, and scheduled visits, and provides photographic documentation, visitor pass pages, and supporting administrative functions. Development involved ten participants in requirements discussions, visit workflow discussions, and design reviews. The ten scenarios in the testing documentation were reported as conforming. These results support functional conformity within the tested conditions. However, the photo capture and visitor pass documentation does not detail verification of photo storage or completed printing. Invalid inputs, failed login, missing records, camera failure, and unauthorized access also lack documented testing. Accordingly, the findings cannot establish comprehensive reliability, security, user satisfaction, or improvements in efficiency.

The study contributes an application of RAD to integrate three visit workflows according to the site's operational requirements. Further development should strengthen documentation of prototype changes, expand testing, and explain security mechanisms and personal data management. Evaluation of actual use is required to assess system usability and operational benefits.

Acknowledgment

-

References

Akello, C. K., & Nabukenya, J. (2024). Users involvement in the electronic health information systems development process in Uganda: What is missing in relation to requirements gathering and analysis. *Oxford Open Digital Health*, 2, oqae020. https://doi.org/10.1093/oodh/oqae020

Aprisalni, R., Komala, I. R., & Ruhiat, A. (2025). Design of the Ceria Space profile information system using the website-based Rapid Application Development method. *Journal Artificial: Informatics and Information Systems*, 3(2), 232–252. https://doi.org/10.54065/artificial.868

Ardhana, V. Y. P., Sapi'i, M., Hasbullah, H., & Sampetoding, E. A. M. (2022). Web-based library information system using Rapid Application Development (RAD) method at Qamarul Huda University. *The IJICS (International Journal of Informatics and Computer Science)*, 6(1), 43–50. https://doi.org/10.30865/ijics.v6i1.4031

Aydos, M., Aldan, Ç., Coşkun, E., & Soydan, A. (2022). Security testing of web applications: A systematic mapping of the literature. *Journal of King Saud University - Computer and Information Sciences*, 34(9), 6775–6792. https://doi.org/10.1016/j.jksuci.2021.09.018

Callens, C. (2023). User involvement as a catalyst for collaborative public service innovation. *Journal of Public Administration Research and Theory*, 33(2), 329–341. https://doi.org/10.1093/jopart/muac030

Daraghmi, Y. A., & Daraghmi, E. Y. (2022). RAPD: Rapid and participatory application development of usable systems during COVID19 crisis. *IEEE Access*, 10, 93601–93614. https://doi.org/10.1109/ACCESS.2022.3203582

Fadlalla, F. F., & Elshoush, H. T. (2023). Input validation vulnerabilities in web applications: Systematic review, classification, and analysis of the current state-of-the-art. *IEEE Access*, 11, 40128–40161. https://doi.org/10.1109/ACCESS.2023.3266385

Haloho, J., Kiswanto, R. H., & Lahallo, J. (2025). Web-based sales management information system for PT. Sinarta Karya Papua using Rapid Application Development. *Jurnal Ilmiah Sistem Informasi*, 4(2), 475–486. https://doi.org/10.51903/5asdpn24

Herwanto, G. B., Ekaputra, F. J., Quirchmayr, G., & Tjoa, A. M. (2024). Toward a holistic privacy requirements engineering process: Insights from a systematic literature review. *IEEE Access*, 12, 47518–47542. https://doi.org/10.1109/ACCESS.2024.3380888

Husodo, C., Nugroho, A. C., & Widjaja, A. A. (2024). Implementation of web based accounting information system using Rapid Application Development (RAD). *SISFORMA*, *11*(2), 100–110. https://doi.org/10.24167/sisforma.v11i2.4579

Kamaraj, K., Lanitha, B., Karthic, S., Prakash, P. N. S., & Mahaveerakannan, R. (2023). A hybridized artificial neural network for automated software test oracle. *Computer Systems Science and Engineering*, 45(2), 1837–1850. https://doi.org/10.32604/csse.2023.029703

Mashkoor, A., Egyed, A., Wille, R., & Stock, S. (2023). Model-driven engineering of safety and security software systems: A systematic mapping study and future research directions. *Journal of Software: Evolution and Process*, 35(7), e2457. https://doi.org/10.1002/smr.2457

Melendres, U. M., & Aranda, K. M. (2024). Development and evaluation of a web-based resident information management system. *Journal of Computer, Software, and Program*, *1*(1), 14–22. https://doi.org/10.69739/jcsp.v1i1.50

Oktaviandri, M., & Foong, K. K. (2019). Design and development of visitor management system. *Mekatronika: Journal of Intelligent Manufacturing & Mechatronics*, 1(1), 73–79. https://doi.org/10.15282/mekatronika.v1i1.152

Rayhan, M. A. (2026). Implementation of Naive Bayes and Support Vector Machine for SMS Spam Classification Using the SMS Spam Collection Dataset. *Journal Artificial: Informatika Dan Sistem Informasi*, *4*(2), 197–212. https://doi.org/10.54065/artificial.1249

Rivera, J. N. D. (2021). VMS Support: A mobile-based support to computerized visitor management system. *Software Impacts*, *8*, Article 100056. https://doi.org/10.1016/j.simpa.2021.100056

Saeeda, H., Ahmad, M. O., & Gustavsson, T. (2023). Identifying and categorizing challenges in large-scale agile software development projects: Insights from two Swedish companies. *ACM SIGAPP Applied Computing Review*, 23(2), 23–43. https://doi.org/10.1145/3610409.3610411

Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-based access control models. *Computer*, 29(2), 38–47. https://doi.org/10.1109/2.485845

Vos, T. E. J., Aho, P., Pastor Ricos, F., Rodriguez-Valdes, O., & Mulders, A. (2021). Testar – Scriptless testing through graphical user interface. *Software Testing, Verification and Reliability*, 31(3), e1771. https://doi.org/10.1002/stvr.1771

Wang, Y., Mäntylä, M. V., Liu, Z., Markkula, J., & Raulamo-Jurvanen, P. (2022). Improving test automation maturity: A multivocal literature review. Software Testing, Verification and Reliability. https://doi.org/10.1002/stvr.1804

Zagita, T. C., & Raharjo, T. (2023). Information security integration with agile software development: Systematic literature review and expert judgement. *The Indonesian Journal of Computer Science*, 12(6). https://doi.org/10.33022/ijcs.v12i6.3593