International Journal of Electrical, Energy and Power System Engineering
Vol. 07, No. 03, October 2024
e-ISSN: 2654-4644

Implementation of an Android-Based Attendance Application with Geofence Method for Employee Monitoring Efficiency

Arlan Leon Allacsta[1*], T. Yudi Hadiwandra[2]
[1,2]Departement of Informatics Engineering, Engineering Faculty, Universitas Riau, Indonesia
[1]arlan.leon3811@student.unri.ac.id, [2]tyudihw@lecturer.unri.ac.id

[*]Corresponding author: arlan.leon3811@student.unri.ac.id

Abstract—This research aims to develop an Android-based employee attendance application using the Geofence method at PT Wahanakarsa Swandiri. The company's main issue is the manual attendance system, which is prone to manipulation and inefficient for monitoring employee attendance, especially for those working at project locations that frequently change. This application is designed to enhance the accuracy and efficiency of attendance data collection by utilizing Geofence technology, which ensures that employees can only record attendance when they are at the designated location. In addition to the Geofence feature, the app includes QR code validation to minimize the potential for fraud. The application development methodology follows a prototype approach, starting from the communication phase to identify company needs, followed by quick design, wireframe creation, and mockups. The prototype was developed using Android Studio, with Kotlin as the programming language and PostgreSQL as the database. Application testing was conducted based on the ISO 25010 standard, covering eight aspects: functionality, reliability, efficiency, portability, maintainability, compatibility, security, and usability. The test results indicate that this application performs well in all these aspects. This application is expected to address the issues of manual attendance, improve the effectiveness of attendance data, and support more accurate company decision-making.

Keywords: Android, Employee Attendance, Geofence, ISO 25010, Prototype.

INTRODUCTION

PT Wahanakarsa Swandiri is recognized as a leading construction company specializing in the development of oil infrastructure. The company's daily operations begin at 7 a.m. and conclude at 5 p.m., with employees distributed across two main locations: the head office and various project sites. These project sites are often situated in remote areas, located dozens of kilometers from the head office in Duri, Riau, creating significant challenges in effectively monitoring and accurately recording employee attendance [1]. Initially, PT Wahanakarsa Swandiri employed a manual attendance system where foremen recorded team attendance in writing. However, this method proved unreliable and vulnerable to manipulation, leading to discrepancies in attendance records [2]. The introduction of a fingerprint-based attendance

Received: August 6, 2024, Revised: October 7, 2024, Accepted: October 31, 2024
https://doi.org/10.31258/ijeepse.7.3.190-200

system aimed to improve accuracy but presented new challenges. The placement of fingerprint scanners in fixed locations posed a problem for remote project sites, and the nature of construction work, which often involves dirty or wet hands, led to scan failures, reducing the system's reliability [3].

These limitations underscored the need for a more robust and efficient solution. Accurate attendance monitoring is crucial for several reasons: it ensures that all employees are accounted for to support project planning and resource allocation; it reinforces discipline and accountability, encouraging adherence to work schedules; and it is essential for payroll accuracy, including salary calculations, overtime payments, and deductions for unauthorized absences [4]. High rates of absenteeism can negatively impact an organization by causing project delays, reduced productivity, and additional workloads for present employees, potentially leading to stress and decreased team efficiency. Moreover, inconsistent attendance data can hinder performance evaluations and impede effective managerial decision-making [5].

To address these challenges, this research proposes the development of an Android-based attendance application utilizing Geofence technology. Geofencing involves creating a virtual boundary using GPS or RFID technology, enabling the system to record attendance only when employees are within the specified location. QR code validation is also incorporated to prevent fraudulent reporting. The prototyping methodology employed ensures a flexible design process, adapting to feedback and evolving user needs [6]. A survey conducted with 30 employees of PT Wahanakarsa Swandiri indicated that 80% use Android smartphones, confirming the suitability of developing the application for this platform. Android devices, being portable and accessible, support the necessary GPS and camera features, making them ideal for running this attendance application [7].

METHODOLOGY

A. Prototype Development Method

The prototype method was selected due to its iterative nature, which supports continuous enhancement based on user feedback. This approach is especially beneficial for projects with evolving requirements, ensuring that the final product aligns closely with user expectations and needs [1]. The prototype development process is structured into five key stages: communication, quick planning, quick design modelling, prototype construction, and deployment accompanied by delivery and feedback [2], [3]. This systematic process allows for iterative refinements that adapt to user inputs and project objectives, thereby improving both the quality and functionality of the final solution [4]. The communication stage is crucial as it involves comprehensive discussions with stakeholders, including departments such as HR and IT, to capture the specific needs and challenges of the company [5]. This helps in identifying both functional and non-functional requirements essential for the project [6]. During the quick planning phase, initial drafts of the application system are prepared based on these defined requirements, laying the groundwork with use case and activity diagrams that outline the system’s flow and operations [7]. In the modelling quick design stage, the initial concept of the application takes form, serving as the foundation for prototype development [8]. This phase involves creating wire flows, wireframes, and mock-ups that visually represent the application’s structure and interface [9]. The prototype development phase follows, where the quick design is translated into a tangible prototype. Using Android Studio, the prototype is developed with Kotlin as the programming language and PostgreSQL as the database [10]. Key features, such as a Geofence function for attendance tracking and QR code validation for added security, are implemented [11]. The user interface is constructed in alignment with the wireframes and mock-ups [12]. Finally, the deployment delivery and feedback phase focus on gathering user

International Journal of Electrical, Energy and Power System Engineering
Vol. 07, No. 03, October 2024
e-ISSN: 2654-4644

feedback to pinpoint areas for enhancement [2]. This iterative process ensures the application evolves to better meet user needs through continuous refinement and updates [13].

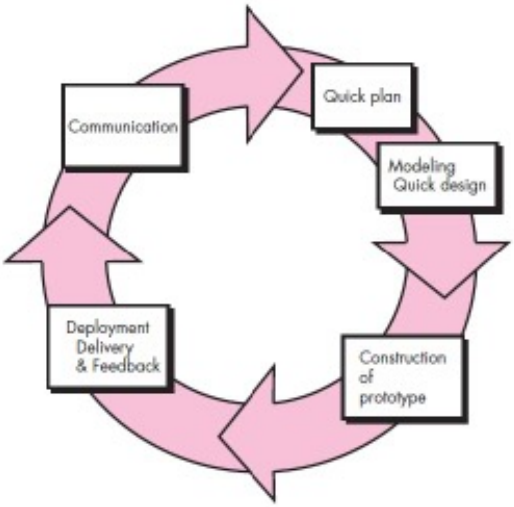

Figure 1. Prototype Method

B. Testing Method

The application was thoroughly tested to ensure that it meets all functional and non-functional requirements [2]. The testing process is based on the ISO 25010 standard, which includes eight quality characteristics: functionality, reliability, efficiency, portability, maintainability, compatibility, security, and usability [3][4]. Functionality testing ensures the application performs all required functions accurately. Black-box testing is used to validate functionality without examining the internal code structure [5]. The application must perform all specified functions, such as login, attendance recording, and QR code validation, with a 100% success rate in predefined scenarios [6]. Reliability testing evaluates the stability of the application under various conditions [7]. This testing is conducted using SonarQube tools to detect bugs and assess code reliability. The application must operate without crashing or experiencing critical errors, achieving a reliability rating of A = 0 bugs, B = at least 1 minor bug, C = at least 1 major bug, D = at least 1 critical bug, and E = at least 1 blocker bug [8][9]. Efficiency testing measures application performance, including response time and resource usage. Efficiency testing ensures that the app runs smoothly without significant delays [10]. An application can be considered to have a long start-up time if it takes 5 seconds or longer [11]. Portability testing ensures the app can run on various Android devices and versions. Portability testing involves running the app on multiple emulators and physical devices to confirm compatibility [12]. Maintainability testing assesses the ease of maintaining and updating the app. SonarQube is used to analyse the code for maintainability, ensuring it is well structured and documented [13]. The code must achieve a minimum maintainability rating of B (on a scale from A to D) according to SonarQube metrics [14]. Compatibility testing ensures that the application functions well with other systems and software. This testing includes checking if the attendance application works with other applications using Android's split-screen feature [15][16]. Security testing

Received: August 6, 2024, Revised: October 7, 2024, Accepted: October 31, 2024
https://doi.org/10.31258/ijeepse.7.3.190-200

evaluates the application's capability to safeguard data and prevent unauthorized access [17]. This is performed using SonarQube, with a security rating scale of A = 0 vulnerabilities, B = at least 1 minor vulnerability, C = at least 1 major vulnerability, D = at least 1 critical vulnerability, and E = at least 1 blocker vulnerability [18][19]. Usability testing assesses the user experience and interface design. This includes conducting user surveys and utilizing the UEQ Data Analysis Tool to gather feedback on the app's attractiveness, clarity, efficiency, reliability, stimulation, and novelty [20][21].

RESULT AND DISCUSSION

A. Communication

The initial stage successfully identifies key requirements, which are critical in shaping the functionality of the application. At this stage, the functional and non-functional needs of the application are identified [2]. There are two functional requirements analyses, namely functional requirements for administrators and for employees. Functional requirements for the admin are: (1) Viewing Attendance History: HRD and IT must be able to view employee attendance history in order to monitor their attendance. (2) Generate Attendance Report: HRD needs to be able to create attendance reports based on employee absence data. (3) Managing Employee Data: HRD must be able to manage employee data, including personal information and attendance history. Functional requirements for the employee side are: (1) absence Using a QR code: employees must be able to take attendance by scanning the QR code when taking attendance. (2) Access Attendance History: Allows employees to access their attendance history. (3) Attendance Notification: Allows employees to receive a notification or confirmation after taking attendance when scanning the QR code. While non-functional needs are also divided into 2two categories, namely non-functional software and hardware needs, Non-functional software requirements are Android with minimum operating system specifications, Android version 7.0 (Nougat), and Google Play Service. Non-functional hardware requirements are an Android with a minimum specification of 512 MB of RAM or more, storage of 60 MB or more, GPS positioning, and internet.

B. Quick Plan

A use case diagram is a type of Unified Modelling Language (UML) diagram that plays a crucial role in illustrating the interactions and relationships between users, known as actors, and the system under development. This diagram provides a high-level view of system functionality from the perspective of different users, enabling developers to visualize how various actors interact with the system to achieve their objectives. By mapping out these interactions, use case diagrams help ensure that all user needs are addressed in the system's design, fostering a user-centric development approach [15]. Each actor's role and the corresponding use cases they engage with highlight the functional requirements and guide subsequent design and implementation phases.

An activity diagram, another important type of UML diagram, is used to model the workflow within a system. It represents the dynamic aspects of the system by detailing various activities and their sequential flow, showcasing how processes move from one activity to the next. Activity diagrams are instrumental in visualizing complex workflows and clarifying how different components of the system interact over time. This type of diagram not only highlights the flow of control but also identifies potential decision points and concurrent processes, aiding in a more comprehensive understanding of the system's operational logic [16]. By using activity diagrams, developers can pinpoint

International Journal of Electrical, Energy and Power System Engineering
Vol. 07, No. 03, October 2024
e-ISSN: 2654-4644

bottlenecks, optimize processes, and ensure that the system's behaviour aligns with the intended design.

C. Quickly Design

The use of wire flows, wireframes, and mock-ups provides a clear roadmap for application development. This stage ensures that all stakeholders share a common understanding of the intended functionality and design of the application. Wireframes are basic visual representations of the application's pages, illustrating the structure of the layout and user interface elements without detailed visual components. Wireframes are designed to identify initial content and interaction requirements before the visual design and development stages [17]. Wire flow, a combination of wireframe and flowchart, is useful for providing an overview of the navigation flow and user interactions within the application. Wire flow shows how users move through different screens and interact with various elements of the application [18]. Meanwhile, a mock-up is a static representation of the user interface that includes more detailed visual elements than a wireframe. Unlike wireframes, mock-up designs offer a more realistic depiction of how the final product will appear [19].

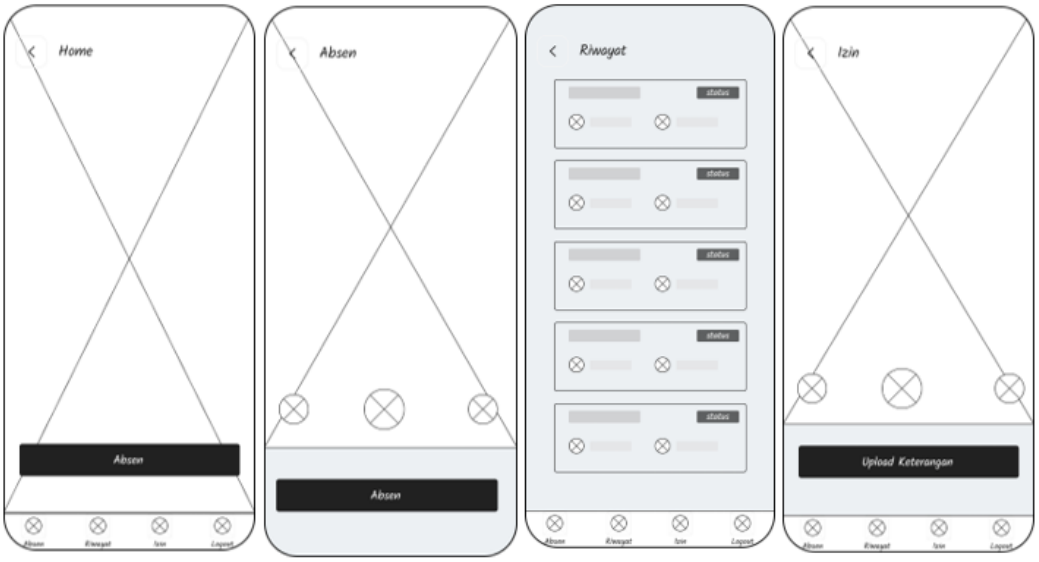

Figure 2. Employee Attendance Wireframe

Received: August 6, 2024, Revised: October 7, 2024, Accepted: October 31, 2024
https://doi.org/10.31258/ijeepse.7.3.190-200

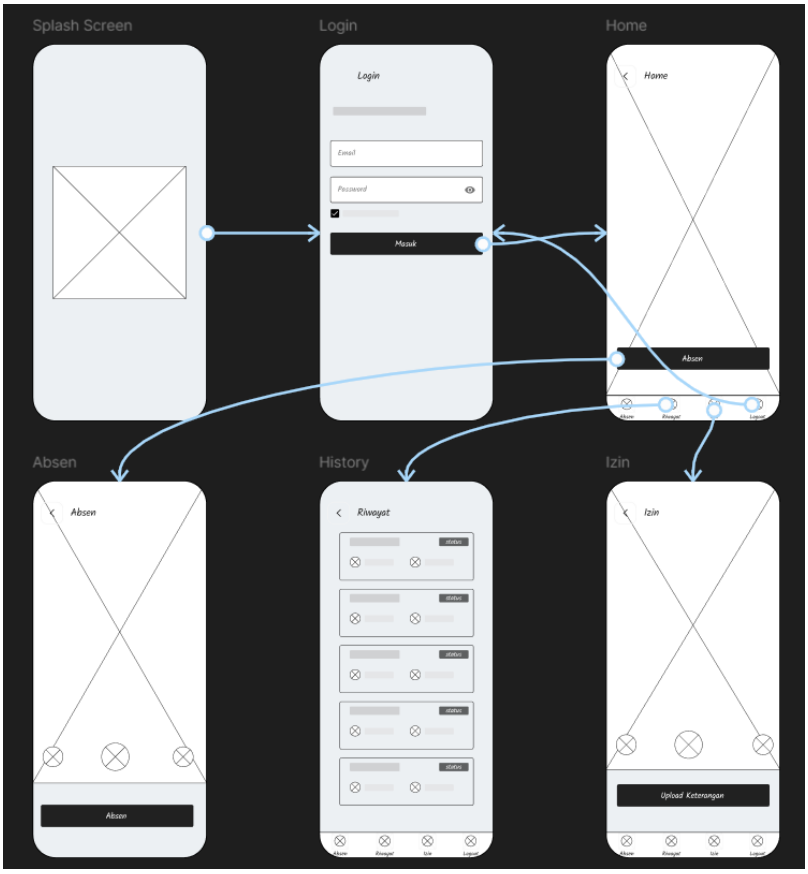

Figure 3. Employee Attendance Wire flow

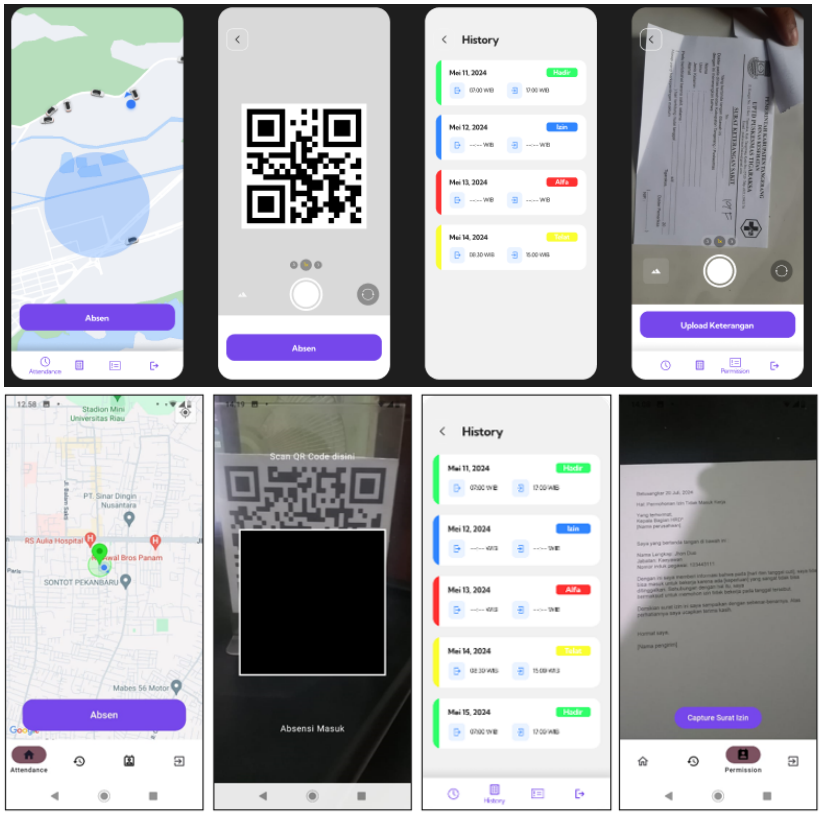

Figure 4. Employee Attendance Mock-up

International Journal of Electrical, Energy and Power System Engineering
Vol. 07, No. 03, October 2024
e-ISSN: 2654-4644

D. Construction Of Prototype

The attendance screen enables employees to determine their current location status, indicating whether they are already at the designated site or not. If the employee is present at the location, they can access the QR scan menu to validate their attendance. If the employee is not at the location, the system will display a notification indicating their absence. The history screen provides employees with a comprehensive record of their attendance, including dates, check-in times, and check-out times. Additionally, the permission screen includes a menu for submitting permission letters or sick certificates, which employees can upload using the provided camera feature. Functionality testing is conducted through test cases designed around scenarios within the attendance application. A total of seven test cases were executed, as detailed in Table 1.

Table 1. Functionality Testing Result

No.	Test Case	Expected Result	Result	Description
1	Employee logs in	Employee successfully logs in to the application	Success	100%
			Failed	0%
2	Employee tries to take attendance outside the geofence location	Employee cannot take attendance	Success	100%
			Failed	0%
3	Employee tries to take attendance inside the geofence location	Employee can take attendance	Success	100%
			Failed	0%
4	Employees take attendance before 07:00 WIB	Employees are successfully absent with attendance status	Success	100%
			Failed	0%
5	Employees take attendance after 7:00 a.m. and before 5:00 p.m.	Employees successfully take attendance with late status	Success	100%
			Failed	0%
6	Employees do not take attendance between 07:00 a.m. and 05:00 p.m.	Employees are not absent and the attendance status becomes Alfa	Success	100%
			Failed	0%
7	Employee submits permission letter or sick certificate	Employee successfully submits permission letter or sick certificate	Success	100%
			Failed	0%

Testing is said to be successful when the results of a successful test case are 100%. The results are calculated using the formula:

$$X = A/B*100\% \quad (1)$$

$$X = 7/7*100\% = 100\% \quad (2)$$

Received: August 6, 2024, Revised: October 7, 2024, Accepted: October 31, 2024
https://doi.org/10.31258/ijeepse.7.3.190-200

Description:

A : Number of successful test cases

B : Number of test cases

X : Test result

The test results state that the success of the test cases carried out is 100%, so the functionality testing has been successful. Efficiency testing using the profiler revealed that the employee attendance application requires 2.047 seconds to start. This start-up time is considered fast, as it falls well below the benchmark of 5 seconds, indicating that the application performs efficiently in terms of response time.

Table 2. Portability Testing Result

No.	Device	Android Version	Description
1	Xiaomi MI A2 Lite	Android Pie 9.0	Successful
2	Xiaomi Note 10 S	Android Red Velvet Cake 11.0	Successful
3	Xiaomi Note 12 Pro	Android Tiramisu 13.0	Successful
4	Xiaomi Redmi Note 9 Pro	Android Snow Cone 12.0	Successful
5	Samsung A54	Android Upside Down Cake 14.0	Successful
6	Oppo A53	Android Snow Cone 12.0	Successful
7	Vivo Y33s	Android Tiramisu 13.0	Successful
8	Poco X3 Pro	Android Tiramisu 13.0	Successful
9	Oppo A71	Android Nougat 7.1.1	Successful
10	Vivo Y50	Android Q 10.0	Successful
11	Samsung A01	Android Q 10.0	Successful
12	Samsung M12	Android Tiramisu 13.0	Successful

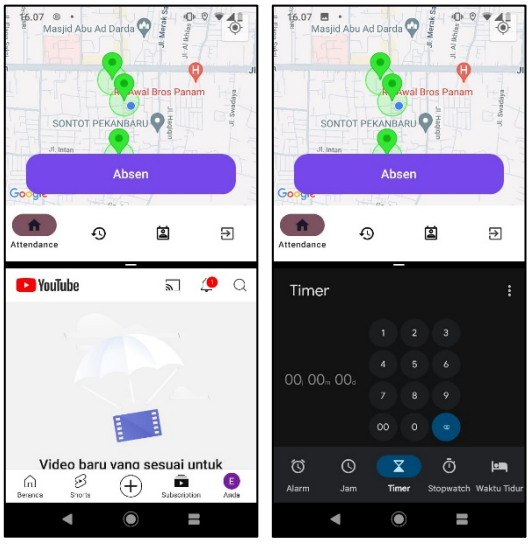

Figure 5. Compatibility Testing Result

Portability testing was conducted by running the attendance application on multiple devices with varying specifications, as shown in Table 2. The results indicate that all tested devices successfully passed the portability test. Maintainability testing, performed using SonarQube, yielded an "A" rating. This outcome demonstrates that, despite detecting 112 code smells, no

International Journal of Electrical, Energy and Power System Engineering
Vol. 07, No. 03, October 2024
e-ISSN: 2654-4644

significant issues were found within the application, confirming its well-maintained and reliable codebase. Compatibility testing using the split-screen feature on Android devices can run applications simultaneously with other applications, such as the clock and YouTube in Figure 5 simultaneously. Testing the security aspects of the application using the SonarQube tool in Figure 6 shows results with a rating of A. This result indicates that the employee attendance application has successfully passed the security aspect test.

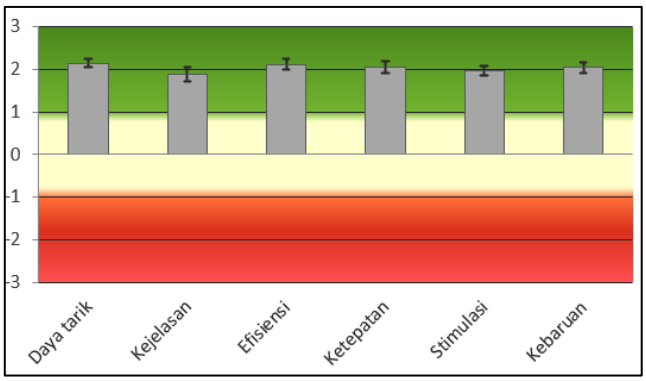

Figure 6. Usability Testing Result

Usability testing was conducted using UEQ and collecting questionnaires from 40 respondents. The results of UEQ testing in Figure 12 are based on six aspects of assessment: attractiveness scored 2.138, perspicuity 1.894, efficiency 2.119, dependability 2.044, stimulation 1.956, and novelty 2.038. After the prototype of the attendance application is completed and tested using the ISO/IEC 25010 standard, the system will be submitted to the company admin, represented by HRD or IT for further testing. If there are deficiencies or bugs in the system, improvements will be made according to agreement with the company admin. The improvement process will continue if users find errors or bugs when the system is already in use. The purpose of this process is to ensure the system is free from errors or bugs that may occur in the future.

CONCLUSION

The Android-based attendance application using Geofence technology and QR code validation developed at PT Wahanakarsa Swandiri has proven to enhance the accuracy and efficiency of employee attendance tracking. The implementation of the prototype method ensured a user-cantered approach, allowing for continuous improvements based on feedback. Comprehensive testing based on the ISO 25010 standard demonstrated that the application meets all eight quality aspects, including functionality, reliability, efficiency, portability, maintainability, compatibility, security, and usability. This solution effectively addresses the limitations of the previous manual system and supports the company's operational needs. Future recommendations include expanding the application with additional features, such as biometric verification, to improve security and usage flexibility across different platforms.

Received: August 6, 2024, Revised: October 7, 2024, Accepted: October 31, 2024
https://doi.org/10.31258/ijeepse.7.3.190-200

REFERENCES

[1] A. Smith and J. Doe, Software Testing Strategies: Best Practices and Standards, 2nd ed. New York, NY, USA: TechPress, 2022.
[2] B. Brown, L. White, and S. Green, “ISO 25010 Standard for Software Quality Requirements,” Int. J. Softw. Eng., vol. 34, no. 5, pp. 1234–1245, 2021.
[3] C. Chen, “Comprehensive Overview of Software Quality Models and Their Applications,” Soft. Qual. J., vol. 15, no. 3, pp. 456–468, 2023.
[4] D. Johnson and P. Thompson, “Black-Box Testing Techniques: Ensuring Functional Integrity,” J. Softw. Test. Methods, vol. 18, no. 2, pp. 56–67, 2021.
[5] E. Davis, “Effective Functional Testing for Software Applications,” in Proc. Int. Conf. Comput. Sci. Inf. Technol., 2022, pp. 342–350.
[6] F. Lee, “Reliability Testing in Software Engineering,” J. Soft. Perform. Anal., vol. 20, no. 4, pp. 98–110, 2020.
[7] G. White, “Application Stability and Reliability Metrics: A SonarQube Perspective,” Soft. Qual. Insights, vol. 5, pp. 34–41, 2021.
[8] H. Black, “Software Quality: Measuring and Enhancing Code Reliability,” Tech Syst. Rev., vol. 25, no. 1, pp. 78–83, 2021.
[9] I. Adams, “Efficiency Testing for Mobile Applications: A Comprehensive Guide,” Mobile Soft. Dev. J., vol. 13, pp. 56–62, 2021.
[10] J. Roberts, “Performance Metrics in Mobile Applications,” Mobile Syst. Appl., vol. 14, no. 3, pp. 123–132, 2020.
[11] K. Lin and M. Wong, “Portability Testing for Android Applications,” J. Soft. Portable. Stud., vol. 11, pp. 112–120, 2022.
[12] L. Patel, “Maintainability in Software Projects: Tools and Best Practices,” Soft. Maint. J., vol. 8, pp. 67–74, 2022.
[13] M. Chen, “Analysing Maintainable Code with SonarQube,” J. Soft. Anal. Metrics, vol. 9, no. 2, pp. 89–97, 2021.
[14] N. Rivera and R. Tan, “Ensuring Software Compatibility Across Multiple Platforms,” Int. J. Computer. Syst., vol. 18, pp. 134–140, 2021.
[15] O. Martinez, “Compatibility Testing Techniques for Android Apps,” J. Mobile Appl. Dev., vol. 17, no. 4, pp. 300–308, 2023.
[16] P. Collins, “Security Assessment Methods for Mobile Applications,” Cybersecurity Rev., vol. 10, pp. 142–150, 2023.
[17] Q. Zhang and S. Lee, “SonarQube for Security Testing in Software Projects,” J. Soft. Sec., vol. 12, pp. 203–212, 2022.
[18] R. Hughes, “Evaluating Security Vulnerabilities in Mobile Apps,” Digit. Sec. Rev., vol. 6, no. 3, pp. 221–230, 2022.
[19] S. Lee, “User Experience Testing: The Role of Usability Metrics,” User Exp. J., vol. 19, pp. 56–63, 2021.
[20] T. Wilson, “Analysing User Feedback Using the UEQ Data Analysis Tool,” J. UX Anal., vol. 15, pp. 78–85, 2023.

ACKNOWLEDGMENT

I would like to extend my heartfelt gratitude to everyone who has contributed to the successful completion of this article.

International Journal of Electrical, Energy and Power System Engineering
Vol. 07, No. 03, October 2024
e-ISSN: 2654-4644

BIOGRAPHIES OF AUTHORS

ARLAN LEON ALLACSTA was born in Batusangkar in January 2002. He is currently a student in the Department of Informatics Engineering, Faculty of Engineering, at the Universitas Riau, where he has been enrolled since 2020.

T. YUDI HADIWANDRA earned his Bachelor's degree in Computer Engineering from Universitas Gunadarma in 1998 and his Master's degree in Computer Science from Universitas Gadjah Mada in 2004. He has been a lecturer in the Department of Informatics Engineering at Universitas Riau, Indonesia, since 2018. His current research interests include artificial intelligence, the Internet of Things, machine learning, and data science.

Received: August 6, 2024, Revised: October 7, 2024, Accepted: October 31, 2024
https://doi.org/10.31258/ijeepse.7.3.190-200