ISSN 2356-5462
http://socj.telkomuniversity.ac.id/ijoict/

Intl. Journal on ICT
Vol. 10, No. 1, Jun 2024. pp. 41-52
doi: doi.org/10.21108/ijoict.v10i1.919

The Redevelopment of Proxsis Workspace with Responsive Design and Multiplatform Approaches using Flutter Framework

Shinta Yulia Puspitasari[1]*, Iqbal Abdul Ra'uf[2], Rio Nurtantyana[3], Cahyo Tri Satrio[4]

[1,2,3]*School of Computing, Telkom University*
Bandung, Indonesia
[4]*Techno Infinity*
Bandung, Indonesia

[*] shintayulia@telkomuniversity.ac.id

Abstract

Most of previous studies implemented the responsive design approach for the web-based application platform only since it had several difficulties to apply in the mobile-based application platform. In addition, mobile applications required different codebases since there were several platforms like Android and iOS. However, this study tried to redevelop the Proxsis Workspace website into a mobile application with responsive design and multiplatform approaches using Flutter Framework to explore its potentials and the difficulties of these two approaches for mobile development. We used the Proxis Workspace website as a case study since the system developed well and is important for the company, but the proposed two approaches have not yet been implemented. In addition, we provide detailed improvements, and also perform several types of software testing of the redevelopment app, namely usability evaluation, responsive design testing, and multiplatform compatibility assessment. Eight participants were participated in this study to measure the improvement of the redevelopment application. The results revealed that the redevelopment version of the Proxsis Workspace could implement the responsive design and multiplatform approaches well. Furthermore, the software testing found that the redevelopment version passed the responsive design and multiplatform testing. In addition, there was a significant difference and enhancement of the usability score from 52.50 with the marginal category to 72.81 with the acceptable category.

Keywords: Responsive design, Multiplatform, Flutter Framework, Usability, Redevelopment

I. Introduction

Proxsis Workspace is a human resource information system (HRIS) developed by Techno Infinity company. In the original version, the Proxsis Workspace is based on the website system that employees usually access via desktop computers. The system had three main features, such as a dashboard to see the detailed progress, attendance to check-in/out, and a profile to see the detailed information.

However, the previous studies mentioned the importance of the flexibility and compatibility of the system, which should be accessed not only on the desktop computer but also from any device, such as mobile devices [1]. Furthermore, after the authors observed, we found that the current situation was not aligned with the new

Received on 07 Mar 2024. Revised on 24 Apr 2024. Accepted and Published on 30 Jun 2024.

business process in the company, which can accommodate the employees to work from home since the COVID-19 pandemic situations. Hence, many employees used their mobile devices rather than desktop computers in their homes to check-in/out for attendance on the Proxsis Workspace website. However, there was another issue since the employees used different devices like mobile phones with different screens, in which the Proxsis Workspace was not properly rendered.

On the other hand, the previous studies mentioned that usability testing was a method to measure a system [2, 3]. It is because usability testing is the heuristic approach to evaluate the system with a questionnaire based on the users' perspectives, for example, the Usability System Scale (SUS) [2]. Hence, the authors conducted preliminary usability testing with SUS to measure and understand the quality of the original version of the Proxsis Workspace. The results showed that the original version reached an SUS score of 54.375 points, which was labeled poor usability. There were several reasons, such as the user interface (UI) not being rendered properly on the mobile devices (e.g., block elements were overlapping, there were inconsistencies in the UI); the navigation was difficult to click; not compatible with the different web browsers or operating system; and it is hard to input the form in the mobile devices. However, a previous study mentioned that the system must reach the SUS score of 68 points to satisfy the users [3].

Furthermore, a previous study suggested that an application could accommodate users widely with mobile devices even if an application is based on the website [4]. Several studies suggested using a responsive design approach with a mobile-first strategy to enhance the usability aspects [5-7]. This approach could properly analyze the development to reach the requirement for mobile devices [5]. Hence, in this study, the authors used a responsive design approach to enhance the original version of Proxsis Workspace.

In addition, several studies also compared the mobile application development approaches between the responsive web application and the multiplatform approach [8, 9]. However, most studies suggested using a multiplatform application approach to accommodate diverse users with different devices because it is more useful and widely. This phenomenon was supported by Flutter Framework by Google, which could help the developers build the application with a single code base to accomplish multiplatform application [10]. Several studies confirmed that using Flutter Framework could tackle devices compatibility, since an application developed with Flutter can be used in different operating systems and screen sizes [11].

Before distributing to users, a development application needs to be passed the testing process. Furthermore, there were several mobile application development testing methods like standard ISO 25010 with several aspects for software testing [12]. However, in this study, we focus only on usability and compatibility aspects, which matched our proposed approaches, such as the responsive design and multiplatform. Hence, we conduct multiplatform testing with different operating systems and screen sizes to check its compatibility [10, 12].

To sum up, the authors will analyze and elaborate on the issues with proper development, approaches, and the testing process for the redevelopment of the Proxsis Workspace. Hence, we addressed the issues in three research questions, such as:

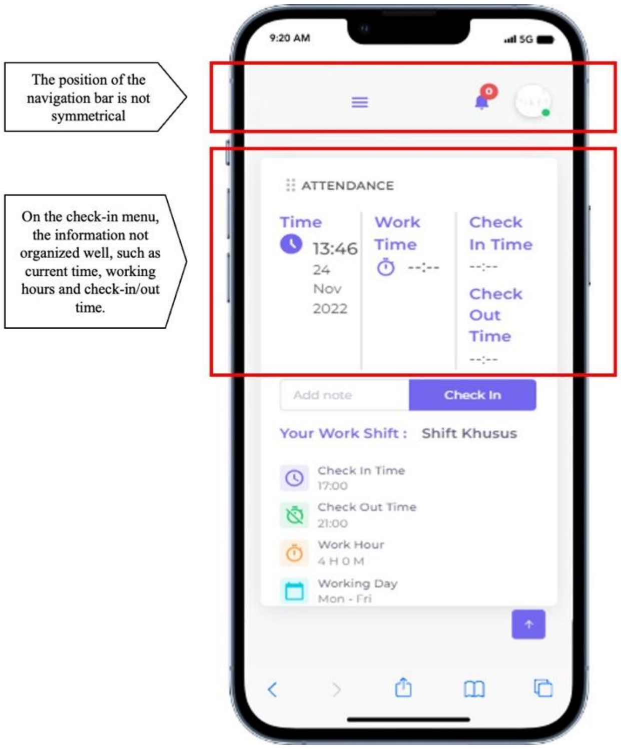
1. What are the differences between the redevelopment of the Proxsis Workspace with responsive design and multiplatform approaches compared to the original version?
2. What are the software testing results of the redevelopment of the Proxsis Workspace with responsive design and multiplatform approaches?
3. Are there any improvements in the user perspectives after the redevelopment of the Proxsis Workspace with responsive design and multiplatform approaches?

II. LITERATURE REVIEW

A. *The original version of Proxsis Workspace.*

Proxsis Workspace is a web-based human resource information system (HRIS) to manage the employees in the Proxsis Group, as shown in Fig. 1. The Proxis Workspace was developed and utilized in 2022 by their subsidiary software company, namely Techno Infinity [13]. There were several features that mainly manage the employee, such as a monitoring dashboard to see their performances, attendance, leaving management, profile, and salary.

However, several issues need to be addressed in terms of the technical part after the authors conducted observation with several employees and one of the top executives. For example, the Proxsis Workspace did not properly render when opened in different screen resolutions or operating systems, particularly on mobile devices (see Fig 1). In this study, the authors mainly focused on the three features only based on the observation results, such as dashboard, attendance, and profile features. It is because the employee mainly used the three features daily.



The position of the navigation bar is not symmetrical

On the check-in menu, the information not organized well, such as current time, working hours and check-in/out time.

Fig. 1. The attendance feature in the original version of Proxis Workspace when opened in the mobile screen size.

B. *The redevelopment with responsive design and multiplatform approaches.*

The responsive design approach ensures that the redesigned application will adapt to different device types and screen sizes, providing an optimal user experience across various platforms. However, the platform commonly utilized in previous research was website-based [14]. Despite employing a website platform, the display necessitated large screens to ensure all information was visible [9, 14].

Furthermore, the original version of the Proxsis Workspace cannot render properly in small screen size resolutions like tablet or mobile devices. This issue poses a challenge for executives and employees with high mobility or limited office presence since the COVID-19 pandemic. Thus, there is a need for a system that is accessible via smartphones or tablets. However, the original version did not utilize responsive design in the smartphones or tablet devices [3]. Consequently, while accessible via smartphones, the employees encountered difficulty viewing information as it was not well displayed on their devices. This challenge extends to application developers, where creating responsive designs and multiplatform capabilities represents a significant challenge and requires substantial resources [1].

In addition, the mobile-first strategy in the responsive design approach could help to analyze and guide the redevelopment process with the main focus on mobile devices [11]. Several previous studies have mentioned that using a mobile-first strategy for responsive design is useful [7, 11, 15, 16]. It is because the mobile-first strategy include comprehensive criteria to analyze the system, such as, a) organization part, for example, ensuring the navigation placement does not disrupt the display of existing content, navigation options are streamlined to highlight the website's key features, maintaining clarity and focus; b) action part, for example, ensuring that buttons or touch targets are sufficiently large to facilitate ease of use and prevent inadvertent touches; c) Inputs part, for example, input elements in the UI will be customized based on input types and attributes, with Android and iOS mobile devices each featuring input controls tailored to their respective platforms; and d) layout part, the UI incorporates a vertical linear layout for mobile views (including header, content, and footer), relative layout for desktop views, with input derived from a content reference diagram from the needs analysis, focusing on responsive design and mobile-centric features in the mobile-first approach.

Hence, the primary principle in the responsive design approach with mobile-first was to prioritize website layout for mobile devices and allow for expansion when displayed on desktops [16]. This approach mitigates the previous issue of adapting content from large horizontal screens to smaller vertical ones. Furthermore, the mobile-first strategy encompassed designing interfaces for both mobile and desktop platforms to ensure that the mobile experience is as optimal as the desktop experience. In addition, the previous study also mentioned that implementing mobile-first in the responsive design approach could enhance usability [17].

On the other hand, the multiplatform approach that enables software to adapt in the different operating systems and devices were the suitable approach to combine with the responsive design approach. It is because the original version of Proxis Workspace with the web-based system could be converted to packaged software that could be executed in different operating systems and devices [18]. Furthermore, Google developed a development tool that supports not only responsive design but also the building of the application across various platforms or multiplatforms, named the Flutter Framework [19]. This enables application developers to simplify their tasks and eliminate the need for extensive resources to develop systems using Flutter [19]. Flutter is an open-source UI software development kit created by Google, which allows for the creation of natively compiled applications for mobile, web, and desktop from a single codebase. Developers could make a single system using Dart language, which can then be executed on different platforms such as Android, iOS, and websites [1], with the system's display becoming more responsive, adjusting to screen resolution sizes like those on smartphones, tablets, and even desktops. Therefore, with the capabilities offered by the Flutter Framework, it can serve as the primary development tool for implementing UI/UX designs created with design applications such as Figma or Sketch [20] , thus enabling the creation of systems with responsive design and multiplatform execution easier.

C. The users' perspectives with usability testing.

A robust system has undergone internal testing before direct user deployment [21]. Therefore, both system developers and researchers must consider several prerequisite tests. Various tests can be conducted to ensure system reliability, including functionality testing, UI/UX design testing, usability testing, and reliability testing adjusted to international standards, like ISO 25010 [22].

UI responsiveness testing is carried out using two methods. The first method involves establishing layout rules in Flutter according to the resolution, whether for mobile, tablet, or desktop [19]. This approach entails indirectly simulating testing to verify the responsive design of the created monitoring dashboard system. The second method involves directly accessing or running the monitoring dashboard system using the hardware available, such as smartphones, tablets, or desktops [21]. While this method is the most effective for testing responsive design, it may be hindered by the researcher's device limitations. Consequently, some previous studies borrowed devices and ran them online, utilizing services like AWS Device Farm or devices simulator on the computer. This allows for online testing while running the system on its actual hardware [23].

Additionally, usability testing can be conducted to measure the extent to which an application can be utilized by users with ease and satisfaction [3]. Several studies have used standardized instruments such as the System Usability Scale (SUS) for usability testing. The SUS questionnaire, comprising aspects such as learnability, efficiency, memorability, errors, and satisfaction, provides a comprehensive framework for assessing user perceptions regarding usability. The users act as respondents to conduct the SUS questionnaire, which is filled out using Likert scales. Then, the SUS results will be calculated according to standardized formulas. The usability testing results are expected to surpass a score of 68 points, indicating that the monitoring dashboard system is suitable for use and meets user expectations [2]. Furthermore, the previous study suggested at least using 5 (five) participants to conduct usability testing with SUS [24].

III. RESEARCH METHOD

A. The participants

The participants in this study were 8 (eight) employees who usually used the original version of the Proxis Workspace, and they were randomly selected. They participated in evaluating the usability of the original version and the redevelopment version of the Proxsis Workspace.

B. The redevelopment procedure

The flowchart in Fig. 2 outlines the methodology for the redevelopment procedure of the Proxsis Workspace application from the website to a mobile application platform. The methods in this study incorporate evaluation, design, development, and testing phases.

The process begins with evaluating the original version of Proxsis Workspace using the System Usability Scale (SUS). The SUS is a standardized tool that provides a quick, reliable measure of a product's usability. By assessing the original application with the SUS, we aim to establish a baseline of usability, identifying areas for improvement from the user's perspective.

Following the evaluation, the research involves analyzing and applying the responsive design approach with responsive design approach including the mobile-first strategy. A responsive design is crucial in the current technological landscape, where users might access applications on many devices with differing capabilities. Concurrently, we applied the multiplatform approach with the Flutter Framework. Utilizing Flutter for redevelopment signifies an intention to streamline development processes and ensure consistency in user experience across different platforms.

The next stage involves testing the new redevelopment application, ensuring the responsive design and multiplatform elements perform as intended. Compatibility testing in these contexts would likely involve various device types and operating systems to confirm that the application is responsive and functional across multiple platforms.

Finally, the new redevelopment application undergoes usability testing using the SUS again. This system usability testing serves as a comparative analysis to the initial evaluation, providing metrics on the usability improvements. It also offers feedback for any further refinements needed before the final deployment of the redesigned app.

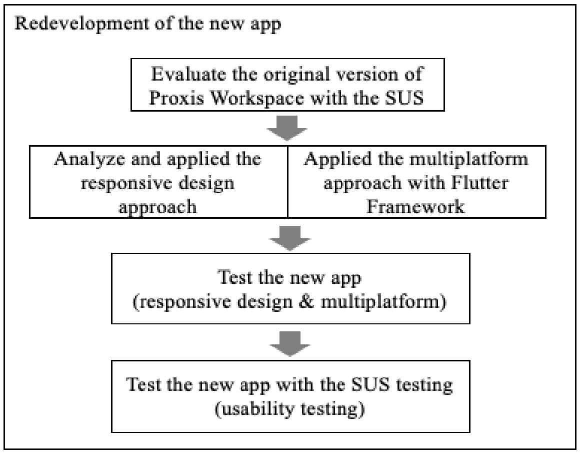

Redevelopment of the new app
Evaluate the original version of Proxis Workspace with the SUS
Analyze and applied the responsive design approach
Applied the multiplatform approach with Flutter Framework
Test the new app (responsive design & multiplatform)
Test the new app with the SUS testing (usability testing)

Fig. 2. The redevelopment procedure.

C. Research variables, data collection, and measurement.

The research variables encompassed in this study, include usability evaluation, responsive design testing, and multiplatform compatibility assessment. Firstly, the System Usability Scale (SUS) serves as a crucial variable for evaluating both the original and redeveloped versions of the application. The questionnaire enables authors to quantitatively measure the effectiveness of the application's usability improvements.

Secondly, the variable of responsive design testing involves assessing the application's performance across various screen sizes, namely mobile, tablet, and desktop screens. This variable allows researchers to evaluate the adaptability and functionality of the application's design across different device types, ensuring a consistent user experience irrespective of screen dimensions. Additionally, the multiplatform variable involves testing the application's compatibility across various platforms, including Android, iOS, and desktop environments. By measuring the application's performance on these platforms, researchers can ascertain its versatility and usability across diverse operating systems, contributing to a comprehensive understanding of its effectiveness and user accessibility. The authors labeled the testing with the passed label if the multiplatform app could perform and render properly.

Data collection procedures encompassed administering SUS questionnaires for usability evaluation and conducting responsive design and multiplatform testing on different screen sizes and platforms with device simulators. We collected the SUS questionnaire data directly using a paper-based method after the participants had finished using the original and the redevelopment version of the Proxis Website, respectively. Afterward, we calculated the SUS scores and compared the scores from the original and the redevelopment version of the Proxis Website. Furthermore, statistical analysis facilitated through SPSS software with the Mann-Whitney Test was used to quantify usability improvements. In addition, the authors conducted an interview with three participants who were selected randomly to understand deeply their perspectives on the data-coding analysis.

IV. RESULTS AND DISCUSSION

A. The comparison between the original version and the redevelopment version.

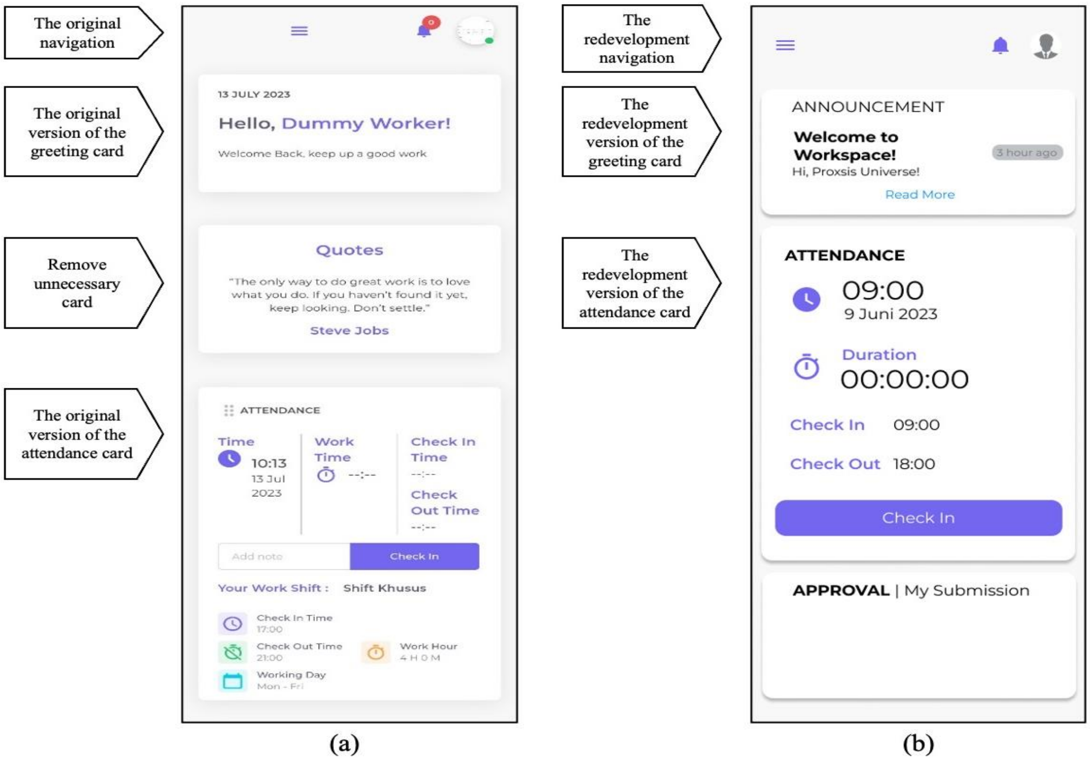
The redevelopment process of a mobile interface adheres to the principles of responsive design with the mobile-first strategy. This redevelopment begins with developing an interface suited for the smallest screens, gradually enhancing and adapting it for larger screens [11]. The problem identified with the original version, according to Table 1, the elements such as card greetings and quotes on mobile devices extended the screen's display, necessitating additional scrolling for users to access critical features. Hence, adopting a responsive design with a mobile-first strategy, the solution proposed in the redevelopment version aims to conceal these less essential elements, thereby streamlining the user interface. This enhances usability, reduces visual clutter, and allows for a more direct path to important features like the quick check-in, demonstrating a responsive design's priority to facilitate ease of use on smaller devices without sacrificing functionality, as shown in Fig. 3.

Beyond adjusting content visibility, the redevelopment focuses on refining the user interface. The original version's input field for attendance notes presented a challenge due to the small screen size, making entering data complicated. The redevelopment version's solution was expanding the "Check-in" button and removing the input field for notes. Furthermore, by streamlining the display of key elements like the "Announcement" card and adjusting the layout to minimize the need for scrolling, the redesign shows an acute awareness of mobile users' interaction patterns and preferences. This approach illustrates a commitment to usability enhancements by the mobile-first strategy, which focuses on delivering a seamless and intuitive experience even on the most constrained screens.

Lastly, organizational adjustments in the redevelopment process reflect a proper application of a mobile-first design strategy. The redesigned interface only displays essential information in the original version, prioritizing clarity and minimizing information overload. Moreover, the strategy of removing redundant actions, such as direct links to messaging applications from user profiles, simplifies the user journey, revealing an understanding of the context in which mobile interfaces are often used. These thoughtful redesign choices underscore the imperative to not only responsively adapt visual elements but also consider the logical structure and interaction flow within the user interface, thereby upholding the tenets of a responsive and mobile-first strategy.

TABLE I
THE ANALYSIS OF THE FEATURES FOR THE REDEVELOPMENT PROCESS

Features / Component	Principle	Original version (problem)	The redevelopment version (solution)
Dashboard	Organization	The inclusion of card greetings and quotes on mobile devices elongates the display and necessitates additional scrolling to access the quick check-in feature.	To optimize the mobile interface, concealing the "Greetings" and "Quotes" elements would enhance usability and streamline access to essential features such as quick check-in.
	Action	The presence of an input field for attendance notes within the "Check-in" button reduces its size, thereby complicating user interaction, particularly on mobile displays where precise clicking is challenging.	Expanding the size of the "Check-in" button and removing the input field for notes would optimize the user experience on mobile displays, facilitating easier interaction and eliminating unnecessary complexity.
	Input	-	-
	Layout	Placing the announcement card beneath the attendance card in the mobile view necessitates users to scroll to view the announcement, which may disrupt the user experience and accessibility of important information.	In the mobile view, the announcement card is displayed above the attendance card, ensuring users can readily access announcements without excessive scrolling.
Attendances	Organization	In the website view, comprehensive user attendance information contributes to a lengthened display, potentially decreasing the user experience.	In the mobile view, the "Requested Attendances" card is hidden from view to streamline the interface and optimize the user experience for smaller screens.
	Action	-	-
	Input	-	-
	Layout	-	-
Profile	Organization	In the website view, the full display of staff profile information extends the layout on mobile devices, potentially leading to a lengthened presentation that requires additional scrolling to access all details.	In the mobile view, only essential information is displayed to streamline the layout and ensure optimal viewing on mobile devices, prioritizing key details for accessibility and user convenience.
	Action	The inclusion of direct buttons leading to "WhatsApp" and "Email," which direct users to their contact information, is unnecessary since the profile menu is accessed by the users themselves.	The removal of the "WhatsApp" and "Email" buttons, replaced by direct buttons linking to the user's social media accounts, enhances the interface by providing more relevant and convenient options for user engagement and communication.
	Input	-	-
	Layout	The "Employee Information" card is relocated to the Profile tab because it contains details pertinent to the user's status as an employee within the company.	The "Employee Information" card is relocated to a separate tab to streamline the interface and organize content more effectively. This ensures that users can access employee-related details without cluttering the primary interface.



Welcome Back, keep up a good work

"The only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle."

Welcome to Workspace!
Hi, Proxsis Universe!

Fig. 3. The example of the redevelopment process with responsive design in mobile screen view, such as a) the original version of the Proxsis Workspace based on the website platform and b) the redevelopment version of the Proxsis Workspace based on the Android mobile platform (multiplatform).

Furthermore, the authors embarked on a redevelopment result for the original Proxsis Workspace, transitioning from a website platform to multiplatform applications, with a particular emphasis on mobile platforms in alignment with a mobile-first strategy, as shown in Fig. 4. It illustrates the redevelopment process within the context of the attendance feature exemplifies the multiplatform approach, with specific instances including different platform and operating system. Furthermore, further examination encompassed responsive design and multiplatform testing, which are elaborated in the subsequent subsection.

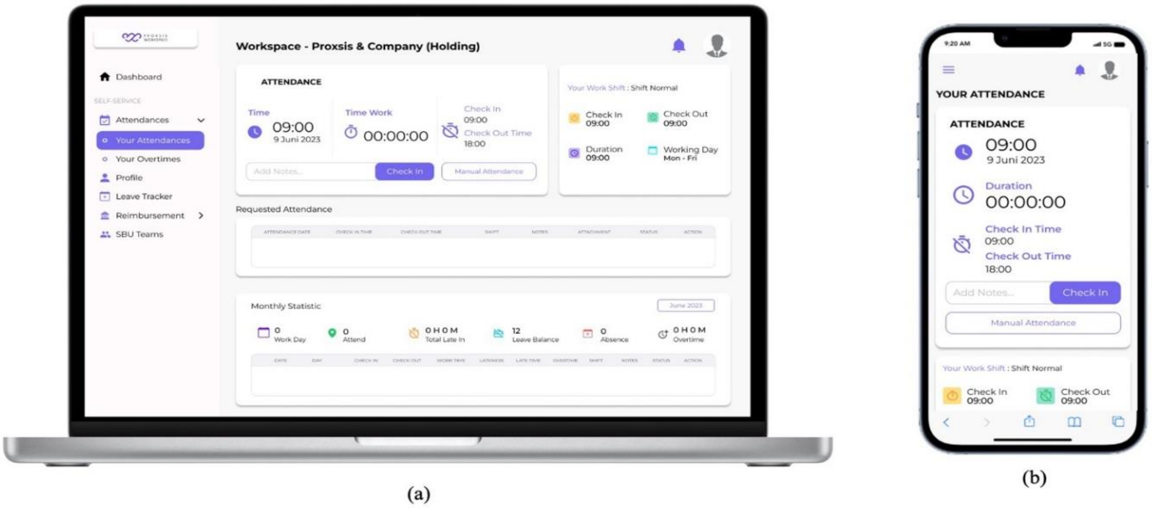


Fig. 4. The example of the redevelopment process in the attendance feature with a multiplatform approach, such as a) the desktop screen with the MacOS platform, and b) the mobile screen with the Android platform.

B. The software testing results of the redevelopment version.

The software testing results of the redevelopment version are summarized in Table II. The testing encompassed multiplatform testing and responsive design evaluation across various screen sizes. The simulator devices used in this testing were three different versions, such as the old device version, the common version, and the newest version. Hence, there were limited simulator devices that could be used for testing. Furthermore, the redevelopment version underwent compatibility testing across three different platforms, such as Android, iOS, and MacOS. For Android, the application was successfully tested on mobile, tablet, and desktop screens, with all testing instances yielding positive results denoted as "Passed." Similarly, on iOS, the application demonstrated successful performance on mobile and tablet screens, meeting the predefined criteria for usability. However, testing on a desktop screen for iOS was not feasible since iOS does not support desktop screens, thereby rendering this evaluation unattainable.

In contrast, the testing scenario differed for MacOS, where certain testing instances could not be conducted due to the platform's inherent limitations. Specifically, testing on mobile and tablet screens was impossible, as MacOS does not support these screen sizes. However, the application successfully passed the testing conducted on a desktop screen in the MacOS, indicating its compatibility and functionality within this platform's environment. These results underscore the importance of considering platform-specific constraints and capabilities when conducting software testing, as each platform presents unique challenges and opportunities for application development and deployment [10, 19].

Overall, the software testing results of the redevelopment version highlight the importance of comprehensive testing methodologies encompassing multiplatform and responsive design testing. By systematically testing the compatibility of the application across different platforms and screen sizes, we can gain insights into its performance and usability across diverse environments. These insights enable informed decision-making regarding further refinements and enhancements to optimize the application's functionality and user experience, ultimately contributing to its effectiveness and success in meeting user needs and preferences [15].

TABLE II
THE SOFTWARE TESTING RESULTS

Multiplatform	Responsive Design	Device Information in Simulator	Results
Android	Mobile screen	Pixel 2; Pixel 5; Pixel 7	Passed
	Tablet screen	Pixel C; Medium Tablet; Pixel Tablet	Passed
	Desktop screen	-	No
iOS	Mobile screen	iPhone 8; iPhone X; iPhone 15	Passed
	Tablet screen	iPad; iPad Air; iPad Pro	Passed
	Desktop screen	-	No
MacOS	Mobile screen	-	No
	Tablet screen	-	No
	Desktop screen	MacBook Air (intel); MacBook Pro (intel); MacBook Pro (arm)	Passed

Notes: Passed = the user interface is appropriately rendered without blocking; No = The authors cannot test the app since the platform did not provide the specific screen resolution.

C. The improvement of the redevelopment version.

The assessment of system usability is a crucial aspect in the development and enhancement of software applications. In this study, the authors conducted a comparative analysis of System Usability Scale (SUS) scores between the original version and the redeveloped version of Proxsis Workspace, as shown in Fig. 5. The SUS questionnaire is a widely used tool for evaluating the usability of software interfaces, providing valuable insights into user experiences and acceptance levels.

Initially, the SUS scores for the original version yielded an average score of 52.50 points. These results categorized the user acceptance level as marginal low, with a grade falling within the F category. Notably, the score fell below the minimum criteria of 68 points, indicating a significant need for improvement in the usability of the original version of Proxsis Workspace. Such findings underscore the necessity for redevelopment efforts to address usability issues and enhance the overall user experience [2].

Subsequently, following the redevelopment of Proxsis Workspace, a final evaluation using the SUS questionnaire was conducted. This evaluation revealed a notable improvement, with an average SUS score of 72.81 points. The detailed results indicated a shift to the acceptable category regarding user acceptance level, with the obtained grade scale falling within the C category. This improvement signifies the efficacy of redevelopment efforts in addressing usability concerns and enhancing the overall usability of the software application [6]. In line with the result, one of the participants mentioned that “the redevelopment version of Proxis Workspace was easier to use and more efficient than the original version” (participant-01).

The substantial increase in SUS scores from the original to the redeveloped version highlights the effectiveness of the redevelopment process in improving system usability. By addressing identified usability issues and implementing enhancements, the redeveloped version of Proxsis Workspace achieved a higher level of user acceptance. Such improvements are crucial for ensuring user satisfaction, productivity, and overall effectiveness of software applications in various contexts. In the interview, all participants were satisfied with the redevelopment version of Proxis Workspace because it was used and rendered properly on mobile and tablet devices.

Hence, the comparative analysis of SUS scores between the original and redeveloped versions of Proxsis Workspace underscores the significance of continuous improvement in system usability. The findings indicated the importance of incorporating user feedback and iterative evaluations to enhance software applications' usability and overall user experiences.

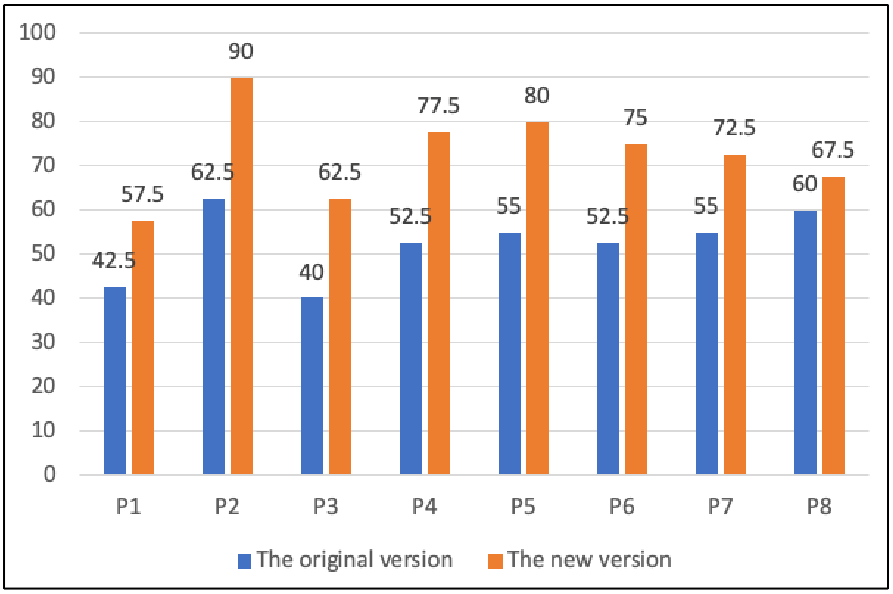


Fig. 5. The comparison of the SUS Scores from eight participants (P1-P8).

Furthermore, we conducted other evidence to compare the SUS results. The Mann-Whitney Test results found a significant difference between the users’ perspectives with the SUS scores of the original version compared to the redevelopment version ($U = 2.500$, $Z = -3.105$, $p < .002$). The SUS score of the original version was 52.50 points, and the redevelopment version had 72.81 points. Hence, there was a significant difference after the authors redeveloped the application. On average, the participants were satisfied, and gained more interest in using the redevelopment version of the Proxis Workspace.

V. CONCLUSION

The redevelopment of the Proxsis Workspace with responsive design and multiplatform approaches differs significantly from the original version. The redevelopment emphasizes responsiveness, usability improvements, and understanding of mobile user preferences. The Proxis Workspace application performed well on Android and iOS with mobile and tablet screens in the test. As a result, there was an enhancement in user perspectives after the redevelopment of Proxsis Workspace. Initially, SUS scores for the original version fell below the 68-point usability threshold, but after redevelopment, they increased to an average of 72.81 points, shifting user acceptance to an acceptable level. In addition, there were significant differences between users' perceptions of the original and the redevelopment version, indicating improved user satisfaction and interest in using the application from the redevelopment version.

In conclusion, while our adherence to the SUS provided valuable insights, the limitation of a minor participant pool suggests the potential for enhanced reliability with a larger sample size. In addition, the authors could not explore all possibilities with Flutter Framework, since several functions still used webview methods. Future studies should explore user experiences with foldable phones to broaden our understanding of their usability implications, paving the way for more comprehensive evaluations and informed design decisions in mobile interface development.

ACKNOWLEDGMENT

This research was supported by the Directorate of Research and Community Service, Telkom University (No. 673/PNLT3/PPM/2023).

REFERENCES

[1] H. Mohr and Z. Walter, "Cross-Platform Interface Design: To Conform or Not to Conform?," *International Journal of Human–Computer Interaction,* pp. 1-13, 2022, doi: 10.1080/10447318.2022.2105480.

[2] W. Welda, D. M. D. U. Putra, and A. M. Dirgayusari, "Usability Testing Website Dengan Menggunakan Metode System Usability Scale (Sus)s," *International Journal of Natural Science and Engineering,* vol. 4, no. 3, pp. 152-161, 11/25 2020, doi: 10.23887/ijnse.v4i2.28864.

[3] A. Parlakkiliç, "Evaluating the effects of responsive design on the usability of academic websites in the pandemic," *Education and Information Technologies,* vol. 27, no. 1, pp. 1307-1322, 2022/01/01 2022, doi: 10.1007/s10639-021-10650-9.

[4] K. Król and D. Zdonek, "Local government website usability on mobile devices: test results and recommendations," *Digital Policy, Regulation and Governance,* vol. 23, no. 2, pp. 173-189, 2021, doi: 10.1108/DPRG-05-2020-0065.

[5] C. Mullins, "Responsive, mobile app, mobile-first: untangling the UX design web in practical experience," presented at the Proceedings of the 33rd Annual International Conference on the Design of Communication, Limerick, Ireland, 2015. [Online]. Available: doi:10.1145/2775441.2775478.

[6] S. Cosgrove, "Exploring usability and user-centered design through emergency management websites: advocating responsive web design," *Commun. Des. Q. Rev,* vol. 6, no. 2, pp. 93–102, 2018, doi: 10.1145/3282665.3282674.

[7] I. Swarna, J. Purnama, and R. Anthony, "Cross-Platform Analysis and Development of Online Catering Platform (Kunyahku)," *Journal of Applied Information, Communication and Technology,* vol. 7, no. 2, pp. 79-89, 03/17 2021, doi: 10.33555/jaict.v7i2.106.

[8] F. Ma, F. Khan, K. U. Khan, and S. XiangYun, "Investigating the Impact of Information Technology, Absorptive Capacity, and Dynamic Capabilities on Firm Performance: An Empirical Study," *SAGE Open,* vol. 11, no. 4, p. 21582440211061388, 2021, doi: 10.1177/21582440211061388.

[9] A. G. Aginsha and B. Noranita, "Designing a performance dashboard as a monitoring tool at PT Sun Star Motor MT Haryono Semarang: data approach," *Journal of Physics: Conference Series,* vol. 1918, no. 4, p. 042133, 2021/06/01 2021, doi: 10.1088/1742-6596/1918/4/042133.

[10] S. Y. Ameen and D. Y. Mohammed, "Developing Cross-Platform Library Using Flutter," *European Journal of Engineering and Technology Research,* vol. 7, no. 2, pp. 18-21, 03/04 2022, doi: 10.24018/ejeng.2022.7.2.2740.

[11] Y. Cheon and C. Chavez, "Converting Android Native Apps to Flutter Cross-Platform Apps," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 15-17 Dec. 2021 2021, pp. 1898-1904, doi: 10.1109/CSCI54926.2021.00355.

[12] S. V. Santos, F. R. S. Ramos, R. Costa, and L. Batalha, "Assessment of the quality of a software application for the prevention of skin lesions in newborns,", *Rev Lat Am Enfermagem,* vol. 28, p. e3352, Sep 7 2020, doi: 10.1590/1518-8345.3711.3352.

[13] T. Infinity. "Techno Story." https://technoinfinity.co.id/techno-story/ (accessed 01 Jan, 2024).

[14] M. C. Setiadi, Z. Zulfiandri, F. Fitroh, and G. P. Utama, "Pengembangan Sistem Informasi Pengawasan Keuangan Berbasis CodeIgniter Framework," *Applied Information System and Management (AISM),* vol. 4, no. 1, pp. 31-36, 2021, doi: 10.15408/aism.v4i1.18537.

[15] S. J. Swierenga, D. B. Propst, J. Ismirle, C. Figlan, and C. K. Coursaris, "Mobile Design Usability Guidelines for Outdoor Recreation and Tourism," Cham, 2014: Springer International Publishing, in HCI in Business, pp. 371-378.

[16] L. Wroblewski, *Mobile First*. A Book Apart, 2011.

[17] A. Zulkarnain, "Penerapan Mobile-First Design pada Antarmuka Website Profil Sekolah Menggunakan Metode Human-Centred Design (Studi Kasus: SMPN 21 Malang)," *Jurnal Ilmiah Teknologi Informasi Asia,* no. 2, pp. 125-136%V 13, 2019-09-27 2019, doi: 10.32815/jitika.v13i2.408.

[18] G. G. Mustofa, B. Mulyanti, and I. Widiaty, "The implementation of multi-platform technology," *IOP Conference Series: Materials Science and Engineering,* vol. 1098, no. 2, p. 022112, 2021/03/01 2021, doi: 10.1088/1757-899X/1098/2/022112.

[19] M. D. Aditya and M. Susanty, "Studi Komparasi Maintainability Antara Aplikasi yang Dikembangkan dengan Framework Flutter dan React Native," *Jurnal Informatika,* vol. 9, no. 2, pp. 159-171, 2022, doi: 10.31294/inf.v9i2.12885.

[20] S. Sudjiran, M. Saefudin, and S. A. Perdana, "DIGITAL SYSTEM UI/UX DESIGN MANAGEMENT SUBMISSION OF AGRICULTURAL COST LOANS USING FIGMA SOFTWARE," *JISICOM (Journal of Information System, Informatics and Computing),* no. 1, pp. 74-85, 2023-06-19 2023, doi: 10.52362/jisicom.v7i1.1090.

[21] J. Fu, Y. Wang, Y. Zhou, and X. Wang, "How resource utilization influences UI responsiveness of Android software," *Information and Software Technology,* vol. 141, p. 106728, 2022/01/01/ 2022, doi: 10.1016/j.infsof.2021.106728.

[22] E. Nasr, W. Alsaggaf, and D. Sinnari, "Developing Usability Guidelines for mHealth Applications (UGmHA)," *Multimodal Technologies and Interaction,* vol. 7, no. 3, p. 26, 2023, doi: 10.3390/mti7030026.

[23] S. Laso, J. Berrocal, P. Fernández, A. Ruiz-Cortés, and J. M. Murillo, "Perses: A framework for the continuous evaluation of the QoS of distributed mobile applications," *Pervasive and Mobile Computing,* vol. 84, p. 101627, 2022/08/01/ 2022, doi: 10.1016/j.pmcj.2022.101627.

[24] J. Nielsen. "Why You Only Need to Test with 5 Users." https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (accessed 01 Jan, 2024).