JURNAL ELTEK
Vol. 24, No. 2, October 2026, pp. 121~130
p-ISSN: 1693 – 4024 | e-ISSN: 2355-0740
DOI: 10.33795/eltek.v24i2.5543

Implementation of user-centered design and white-box testing for a laboratory equipment loan application

Mohamad Dimyati Ayatullah[1], I Wayan Suardinata[2], Mega Devita Sari[3], Lukman Hakim[4] , Aracely Amanda Gusta[5]
e-mail: dimyati@poliwangi.ac.id, wayan.suardinata@poliwangi.ac.id, megadevita@poliwangi.ac.id, lukmanhakim@poliwangi.ac.id, aracelyamanda468@gmail.com
[1,2,3,4,5]Jurusan Bisnis dan Informatika, Teknologi Rekayasa Perangkat Lunak, Politeknik Negeri Banyuwangi, Indonesia

Article Information

Article History

Received August 19, 2026
Revised August 27, 2026
Published October 30, 2026

***Keywords*:**

Equipment Loan Application
User-Centered Design
White Box Testing
Laboratory

ABSTRACT

The laboratory equipment loan process in the Department of Business and Informatics at Politeknik Negeri Banyuwangi was previously managed using paper-based forms, which resulted in limited administrative efficiency and the absence of real-time loan status monitoring. This study aims to develop a web-based laboratory equipment loan application using the User-Centered Design (UCD) approach and to evaluate the reliability of the proposed system workflow through White Box Testing. The UCD process involved identifying user requirements from four user roles, resulting in 14 functional requirements that were fully implemented in the developed system. A task-flow comparison between the existing manual process and the proposed system showed that the minimum number of steps required for three representative loan activities was reduced from eight steps to four steps, indicating a 50% reduction in process complexity. Furthermore, White Box Testing on the student and lecturer loan modules obtained a cyclomatic complexity value of V(G)=2, with all four test scenarios successfully executed. The results demonstrate that the developed system provides complete functional requirement coverage and simplifies the laboratory equipment loan workflow. However, this study does not evaluate processing time improvement or user satisfaction through standardized usability instruments; therefore, claims regarding statistical performance improvement or user acceptance are beyond the scope of this work.

Corresponding Author:
Mohamad Dimyati Ayatullah
Jurusan Bisnis dan Informatika, Teknologi Rekayasa Perangkat Lunak,
Politeknik Negeri Banyuwangi
Jl. Raya Jember No.KM13, Kawang, Labanasem, Kec. Kabat, Kabupaten Banyuwangi, Jawa Timur 68461
Email: dimyati@poliwangi.ac.id

1. INTRODUCTION

Banyuwangi State Polytechnic, located in Banyuwangi, East Java, is a higher education institution actively engaged in technological development across various departments, including the Business and Informatics Department. This department is equipped with several laboratories—such as those for wireless technology, databases, programming (Labs 1 and 2), and multimedia—which serve as vital facilities for practical work and research. Currently, the equipment borrowing process in these laboratories relies on a manual system; students and lecturers must manually fill out forms provided by laboratory technicians [1]. This system has proven inefficient, as it is time-consuming, prone to administrative errors, and often fails to provide accurate information regarding equipment availability [2]. Furthermore, the manual system increases the risk of laboratory equipment damage, loss, or wear and tear [3]. To enhance efficiency and address these issues, developing a web-based application for laboratory equipment borrowing offers an ideal solution. This application aims not only to automate and accelerate the borrowing process but also to ensure an optimal user

experience through effective User Interface (UI) and User Experience (UX) design [4]. The application's development will follow the User-Centered Design method—proven effective in understanding and meeting user needs—by focusing on the stages of analysis, design, evaluation, and implementation [5].

Regarding the development of this laboratory equipment borrowing application, the problem statement focuses on two key aspects. First, how to design the front-end of the application for the Business and Informatics Department using the User-Centered Design method. Second, how to conduct unit testing to ensure the application's functionality and reliability. The objective of this development is to gain a detailed understanding of and optimize user needs regarding equipment borrowing, as well as to improve the user interface so that users can complete the borrowing process quickly and efficiently, without delays or setbacks. The expected benefits of developing this application include providing insights into the design of a web-based system for laboratory equipment loans within the Business and Informatics Department, streamlining practical work processes at Banyuwangi State Polytechnic, enhancing control over the equipment lending process, and facilitating the borrowing of laboratory equipment for both lecturers and students. By integrating the User-Centered Design method for an optimal user experience and White Box Testing to ensure the application's quality and reliability, this development is expected to contribute positively to improving the efficiency and effectiveness of laboratory management in the Business and Informatics Department at Banyuwangi State Polytechnic.

2. RESEARCH METHOD

A research method is a series of procedures or steps used to collect and analyze data with the aim of solving a problem or answering a research question [6]. In the context of developing a laboratory equipment lending application for the Department of Business and Informatics, the research methods employed include the User-Centered Design (UCD) method and the White Box Testing method.

2.1. User-Centered Design (UCD) Method

According to [7], the User-Centered Design method is an approach to product or application development that focuses on the end-user. This approach ensures that the product or application design is not only aesthetically pleasing but also meets user needs and expectations [8]. The User-Centered Design method involves four main stages:

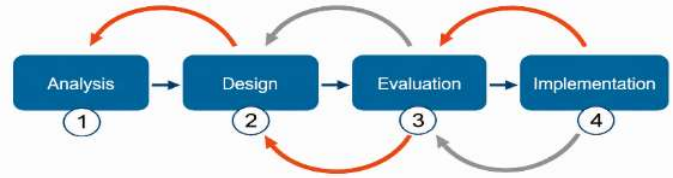


Figure 1. User-Centered Design Method

To ensure the UCD implementation is traceable, the activities, information sources, and outputs at each stage are summarized in Table 1. This approach clarifies that requirements are not merely a list of users but are derived from tasks, issues within manual processes, and the authority associated with each role. A final evaluation was conducted via a requirement-to-feature walkthrough, verifying that each established requirement corresponded to a specific function within the application. As the study lacks documented questionnaire or structured interview data, the manuscript makes no claims regarding subjective user satisfaction measurements.

Table 1. Implementation of UCD and evidence at each stage

UCD Stage	Research Activities	Data/Instrument	Output
Understanding the Context	Identifying the existing manual loan process, user roles, and main problems in the current workflow.	Manual loan workflow and existing loan documents/forms.	Identification of four user roles and limitations of the manual process.
Identifying Requirements	Mapping tasks and user needs for each role into system functional requirements.	Task mapping, identified problems, and functional requirements listed in Tables 2–3.	Functional requirements for each user role.

UCD Stage	Research Activities	Data/Instrument	Output
Designing the Solution	Designing a role-based equipment loan workflow and developing the web-based interface.	User requirements, system workflow, and interface prototype/implementation.	Loan submission, verification, equipment management, tracking, and reporting features.
Evaluation and Implementation	Conducting a requirement-to-feature walkthrough and evaluating the implementation results and logical flow of the loan modules.	Requirement-feature matrix, manual vs. web-based task-flow comparison, and White Box Testing results.	Complete coverage of 14/14 functional requirements, simplified task-flow, and successful test case results.

User analysis is deepened by linking roles, tasks, issues/needs, and system responses. This mapping improves upon the user table in the initial draft—which was merely a list of roles—while also clarifying the distinction in authority between
lecturers and technicians.

Table 2. User analysis and key requirements

User	Main Tasks	Problems/Needs	System Response
Student	Submit equipment loan requests and monitor the loan process.	Requires a simple submission process and clear information regarding loan status.	Provides loan submission forms, borrowing history, status tracking, and loan confirmation records.
Lecturer	Verify student loan requests and submit personal equipment loan requests.	Requires verification authority and access to request equipment as a borrower.	Provides student request verification, lecturer loan submission forms, and loan tracking features.
Laboratory Technician	Manage user registration, equipment data, and check equipment availability.	Requires control over equipment information and availability confirmation before loan approval.	Provides user management, equipment database management, and availability verification features.
Head of Laboratory	Verify equipment loan requests submitted by lecturers.	Requires special authorization for lecturer equipment loan approval.	Provides a lecturer loan request verification menu.

Table 3. Comparison of the manual system and the proposed system

Aspect	Previous System (Manual)	Proposed Web-Based System
Submission Method	Loan requests were submitted using manually completed paper forms [1].	Digital loan submission forms with integrated data validation.
Equipment Availability Information	Equipment stock information was not always accurate or up to date [2].	Equipment availability is verified by laboratory technicians before the loan process proceeds.
Status Monitoring	The manual process did not provide an online mechanism for monitoring loan status.	Loan status can be monitored through tracking pages and borrowing history features.
Authorization	The process relied on manual workflows and direct coordination between users.	Role-based verification involving lecturers, laboratory technicians, and the head of the laboratory.
Data Recapitulation	Manual record keeping increased the risk of administrative errors.	Loan records are stored digitally and supported by data export and recapitulation features.
Efficiency Indicator Used	The minimum number of steps in three representative activities was analyzed based on the existing manual workflow.	The minimum number of steps in the web-based system task-flow was compared with the manual process

Based on these requirements, the proposed system implements distinct verification workflows for student and lecturer loan requests. Student requests require lecturer verification prior to an availability check by a technician, whereas lecturer requests are verified by the laboratory head before being forwarded to the technician. Following the verification process, users can track the status and obtain proof of the loan.



Figure 2. Overview of the Proposal System

This design encompasses user interface (UI) and user experience (UX) elements aimed at facilitating user interaction with the application [10]. Following the creation of the initial design, an evaluation phase is conducted to test the application prototype with potential users. This evaluation ensures that the application meets user expectations and allows for efficient usage. The final stage involves implementing the evaluated design into the actual product or application.

2.2. White Box Testing Method

White Box Testing—often referred to as functional testing [11]—is a software testing technique that evaluates an application's functionality by examining its internal structure or source code [12]. This method focuses on testing the application's internal structure or code and is also known as open-box testing, structural testing, or code-based testing. Unlike Black Box Testing, which considers only inputs and outputs without regard for internal processes, White Box Testing requires a deep understanding of the program code, algorithms, and underlying logic [13]. The primary objective of White Box Testing is to ensure that all code paths within the application function correctly, every logical condition is tested, and the program's internal structure adheres to design specifications and requirements [14]. A White Box test design table is presented below [15]. White Box Testing evaluates internal structure or program code by tracing conditions, branches, and execution paths [12]–[14]. Testing in this study focuses on the function for creating loan records for students and lecturers. The testing stages consist of: (1) identifying nodes and edges in the program flow; (2) constructing a flow graph; (3) calculating cyclomatic complexity using $V(G) = E - N + 2$ or $V(G) = P + 1$; (4) determining independent paths; and (5) executing test cases for both failure and success paths.

3. RESULTS AND DISCUSSION

3.1. Results of UCD Application and Requirement Implementation

The UCD analysis identified four user groups with distinct authorization levels, which were translated into role-based features. To conserve page space and avoid illegible interface visuals, low-resolution screenshots of the dashboard, forms, history, and tracking sections from the initial draft were removed. Evidence of the implementation is summarized as a functional mapping in Table 5 to ensure information remains clearly legible.

Table 4. Mapping user needs to implementation

User	Implemented Features	Requirement Fulfillment
Student	Login, equipment loan submission, borrowing history, loan status tracking, and proof-of-loan download.	Supports equipment loan requests and enables users to monitor loan status.
Lecturer	Login, student loan request verification, lecturer loan submission, and loan tracking.	Supports the lecturer role as both a request verifier and an equipment borrower.

User	Implemented Features	Requirement Fulfillment
Laboratory Technician	User management, equipment data management, and equipment availability verification.	Supports administrative control and ensures equipment availability management.
Head of Laboratory	Verification of lecturer equipment loan requests.	Supports authorization and approval control for lecturer equipment loans.

The mapping indicates that all functional requirements listed in Table 3 have corresponding implementation features. A total of 14 requirements were mapped: four for students, four for lecturers, four for technicians, and two for the laboratory head. All 14 items are accommodated within the application, resulting in a requirement coverage of 100% (14/14). This figure demonstrates the alignment of functions with the defined requirements but is not treated as a measure of user satisfaction or subjective usability.

3.2. Implementation System

The UI/UX design was tested with potential users. Evaluations indicated that the application design was user-friendly and met user needs. The evaluated design was implemented into a web-based application featuring a comprehensive set of functions tailored to user requirements. Below is the application interface for students.

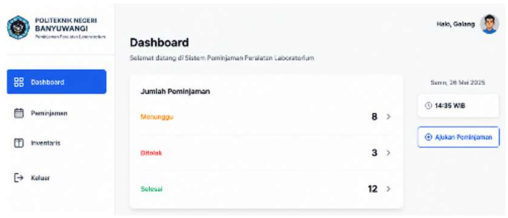


Figure 3. Student Dashboard Page

When requesting to borrow equipment, students fill out a form specifying the borrower's name, study program, department, equipment name, and the quantity to be borrowed. The form includes a feature to add items if a student wishes to borrow more than one piece of equipment. A "submit" button is provided to send or save the completed form for borrowing equipment from the Business and Informatics Department laboratory. The page layout is shown below.

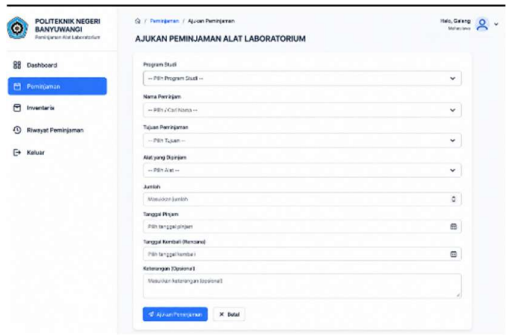


Figure 4. Equipment Loan Request Form

After completing the form, the equipment loan request history will appear. This history displays details such as the borrower's name, study program, department, item, quantity, request date, status, and action (indicating success or rejection). An Excel export option is available to generate a record of the loan. The history page is displayed below.

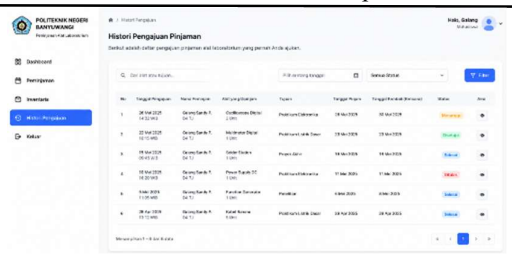


Figure 5. Loan Application History Page

This page allows students to track the status of their equipment loan requests. It covers the submission of requests, lecturer and technician validations, and the downloading of loan confirmation documents. The table includes examples of rejected requests along with the accompanying notes. Here is the interface view.

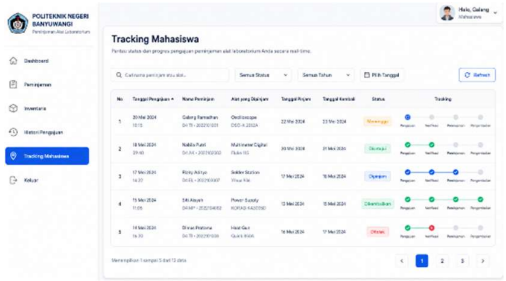


Figure 6. Student Tracking Page

3.3. White Box Testing

. In both modules, the process begins with the validation of the request data. The system then checks whether at least one item has been selected. If no items are selected, the system returns an error message. If items are present in the request, the loan data is saved, the association for each item is recorded, and the user is redirected to the loan list with a success message. The workflow structure is summarized in Figure below Screenshots of the source code have been omitted because they were too small to be legible in the original manuscript.

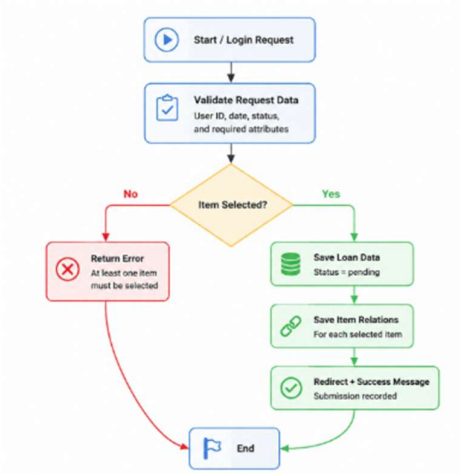


Figure 7. Flowchart for testing the loan module

The flowgraph derived from that flow has 11 edges (E), 11 nodes (N), and one predicate node (P); thus, V(G) = 11 − 11 + 2 = 2 and V(G) = 1 + 1 = 2. Two independent paths represent the failure condition—where the item is not selected—and the success condition, where the submission can be saved.

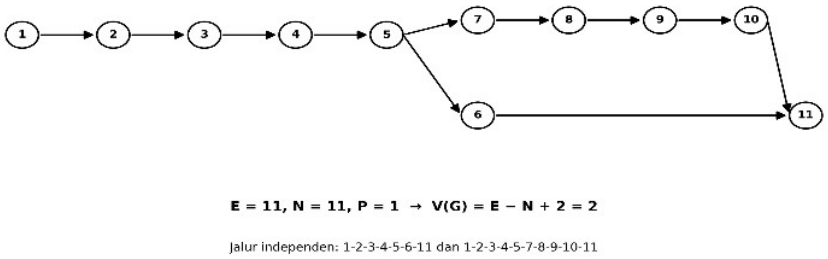


Figure 8. Flowgraph and cyclomatic complexity of the borrowing module

Table 5. Summary of cyclomatic complexity

Module	E	N	P	V(G)	Number of Independent Paths
Student Loan Module	11	11	1	2	2
Lecturer Loan Module	11	11	1	2	2

Table 6. White-box test case results

ID	Module/Scenario	Expected Result	Actual Result	Status
M-01	Student – No item selected	An error message is displayed.	An error message is displayed.	Valid
M-02	Student – Item selected	Loan request is successfully created.	Loan request is successfully created.	Valid
D-01	Lecturer – No item selected	An error message is displayed.	An error message is displayed.	Valid
D-02	Lecturer – Item selected	Loan request is successfully created.	Loan request is successfully created.	Valid

All four test cases yielded valid results, resulting in a 100% pass rate for the tested scenarios. A V(G) value of 2 for both modules indicates the presence of two independent logic paths requiring testing. These results support the conclusion that the student and lecturer loan creation functions operate according to the established scenarios. However, this testing does not conclusively prove that the entire system is error-free, as the scope was limited to the two loan functions described in the study. Software testing was conducted using the White Box Testing method to ensure that all system components function according to specifications and are free of errors.

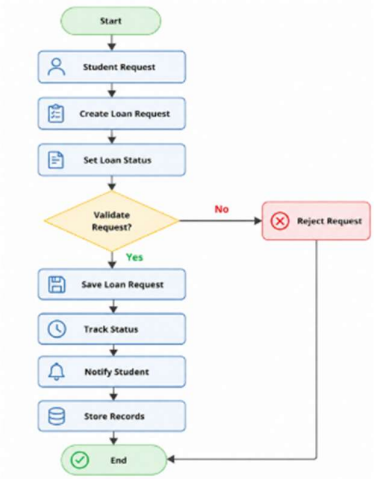


Figure 9. Student Borrowing Process Flowchart

The process begins when a student submits a request to borrow laboratory equipment via a form on the web application. The application verifies that the data entered by the student meets the established criteria:

1. `jumlah_total` (total quantity) must be provided and be an integer.
2. `status` must be one of the specified values (pending, accepted, or rejected).

3. `tanggal_pengajuan` (request date) and `tanggal_pengembalian` (return date) must be provided and be valid dates.
4. `kondisi` (condition) can be empty or one of the specified values (not_returned, returned, or damaged).
5. `mahasiswa_id` (student ID) must exist in the `mahasiswa` (student) table.
6. `nama_lab_id` (lab name ID) must exist in the `nama_lab` (lab name) table.
7. `dosen_id` (lecturer ID) must be provided.
8. `teknisi_approver` (approving technician) can be empty.

If validation succeeds, the application creates a new `PeminjamanBarangMahasiswa` (Student Equipment Loan) object and populates its attributes with data from the student's request. Attributes such as `jumlah_total`, `tanggal_pengajuan`, `tanggal_pengembalian`, `mahasiswa_id`, `dosen_id`, `nama_lab_id`, and `teknisi_approver` are populated according to the request data. The status is set to "pending," and the condition is set to `null`. The application checks whether any items were selected by the student in the request. If items were selected, the `PeminjamanBarangMahasiswa` object is saved to the database. For each selected item, a new `PeminjamanBarangMahasiswa_Barang` object is created to store information regarding that specific item's loan. If no items were selected, the application redirects the student back to the form with an error message indicating that at least one item must be selected. If all steps succeed, the application redirects the student to the loan list page with a success message confirming that the loan has been added. The corresponding flowgraph is shown in Figure 10.

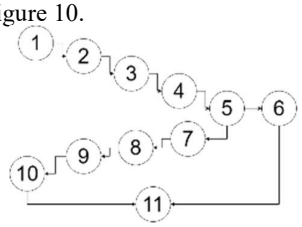


Figure 10. Flowgraph for Student Borrowing Process

The provided flowgraph is used to calculate the cyclomatic complexity of a function or program. Cyclomatic complexity is a software development metric used to measure a program's complexity based on the number of independent paths within the code.

Table 7. Cyclomatic Complexity of Student Borrowing

Edges and Nodes	Predicate Nodes	Regions
V(G) = E – N + 2	V(G) = P + 1	V(G) = R
V(G) = 11 – 11 + 2	V(G) = 1 + 1	V(G) = 2
V(G) = 2	V(G) = 2	

Based on the calculation results above, two independent paths were identified:

1. 1-2-3-4-5-6-11 (Error message appears when no item is selected)
2. 1-2-3-4-5-7-8-9-10-11 (Student item loan request successfully created)

The following is the test case for creating a student item loan request.

Table 8. Student Loan Case

No.	Activity	Expected Result	Actual Result	Remarks
1	Student submits a student equipment loan request without selecting any item.	An error message is displayed indicating that an item must be selected.	An error message is displayed indicating that an item must be selected.	Valid
2	Student submits a student equipment loan request with selected item(s).	The loan request is successfully created.	The loan request is successfully created.	Valid

3.4. Evaluation of Operational Efficiency and Alignment with Needs

Operational efficiency was evaluated based on the minimum number of task steps required for three activities that allow for direct comparison between the manual process and the web-based system: loan requests, status checks, and report generation. A "step" was defined as a primary action performed by a user or staff member to achieve a specific objective. This metric was employed because "before-and-after" time data was unavailable. The results presented in Table 8 indicate that the total minimum number of steps decreased

from eight to four—a 50% reduction. This figure reflects the simplification of the task flow rather than statistical significance regarding processing time.

Table 9. Analysis of process steps and requirements coverage

Activity	Manual Process (Minimum Steps)	Web-Based System (Minimum Steps)	Step Reduction
Loan Submission	3 steps: obtain the form, complete the form, and submit the form.	2 steps: complete the digital form and submit the request.	33.30%
Status Checking	2 steps: contact the officer/verifier and receive confirmation.	1 step: open the tracking page.	50.00%
Report Recapitulation	3 steps: collect records, reorganize data, and prepare the report.	1 step: access loan history/export data.	66.70%
Total	**8 steps**	**4 steps**	**50.00%**
Requirement Fulfillment	14 functional requirements identified.	14 functional requirements implemented.	100% coverage (14/14)

The analysis results indicate that digitalization not only transfers forms to a web-based medium but also reduces the need for media switching and direct coordination within the analyzed activities. A 50% reduction in the minimum number of steps serves as a measurable indicator of process simplification. Furthermore, a 100% requirement coverage demonstrates that all formulated functional requirements have been implemented. However, this study did not employ statistical testing for processing time nor respondent-based usability instruments; consequently, the results are not interpreted as evidence of statistical significance or user satisfaction levels.

Given these limitations, the terms "intuitive" and "user-friendly" in the discussion are interpreted operationally—referring to workflow coherence, role-based functionality, status tracking, and a reduction in process steps—rather than as user perception scores. Future research is recommended to include further testing using the System Usability Scale (SUS), task completion time, or similar instruments to validate usability aspects from a subjective perspective.

4. CONCLUSION

The application of UCD resulted in mapping the needs of four user roles—students, lecturers, technicians, and laboratory heads—into 14 functional requirements. All these requirements were accommodated in the implementation, achieving 100% requirement coverage (14/14). Task-flow analysis regarding submission, status checking, and summarization showed a reduction in the minimum number of steps from eight in the manual process to four in the web-based system—a 50% reduction—indicating operational process simplification. White Box Testing on the student and lecturer borrowing modules yielded a cyclomatic complexity of V(G)=2 for each module and a 100% pass rate for the four test cases, confirming that the tested logic paths operated according to the scenarios. Based on these results, the system is deemed to meet the formulated functional requirements and provide a more structured process. As the study lacked experimental timing data or respondent-based usability questionnaires, the conclusions do not claim statistical significance or make subjective assertions regarding user-friendliness. SUS testing and task completion time measurements are recommended for future research.

5. ACKNOWLEDGMENTS

The authors would like to thank Politeknik Negeri Banyuwangi (Poliwangi) for providing research support, facilities, and also acknowledge the contribution of the Department of Business and Informatics in supporting the data collection and system validation process.

6. REFERENCES

[1] Widharma, I. S., Sukarata, P. G., Sajayasa, I. M., Sangka, I. N. G., & Sunaya, I. N. (2022). Perancangan Sistem Informasi Manajemen Laboratorium Otomasi Berbasis Web Dengan Metode Prototype. Jurnal Ilmiah Vastuwidya, 5(2), 10-16.

[2] Munjazi, A., & Matondang, N. (2022, August). Perancangan Sistem Informasi Inventarisasi Barang Berbasis Web (Studi Kasus Jurusan Teknik Komputer dan Jaringan SMKN 2 Pandeglang). In Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya (Vol. 3, No. 2, pp. 363-375).

[3] Yudanto, A. L., Tolle, H., & Brata, A. H. (2017). Rancang Bangun Aplikasi Sistem Informasi Manajemen Laboratorium Biomedik Fakultas Kedokteran Universitas Brawijaya. Jurnal pengembangan teknologi informasi dan ilmu komputer, 1(8), 628-634.

[4] Hartawan, M. S. (2022). Penerapan *User Centered Design* (UCD) pada wireframe desain *User Interface* dan *User Experience* aplikasi sinopsis film. JEIS: Jurnal Elektro dan Informatika Swadharma, 2(1), 43-47.

[5] Husna, H. T., Susanti, F., & Pratondo, A. (2020). Perancangan Dan Implementasi Desain *User Interface* Dan *User Experience* Pada Aplikasi Pendidikan Seks Untuk Anak Usia 6-12 Tahun. EProceedings of Applied Science, 6(2).

[6] Triandini, E., Jayanatha, S., Indrawan, A., Putra, G. W., & Iswara, B. (2019). Metode systematic literature review untuk identifikasi platform dan metode pengembangan sistem informasi di Indonesia. Indonesian Journal of Information Systems, 1(2), 63-77.

[7] Subhiyakto, E. R., Astuti, Y. P., & Umaroh, L. (2021). Perancangan *User Interface* Aplikasi Pemodelan Perangkat Lunak Menggunakan Metode *User Centered Design*. KONSTELASI: Konvergensi Teknologi dan Sistem Informasi, 1(1), 145-154.

[8] Gunawan, R., Joharudin, A. M., Yudiana, Y., & Awalludin, D. (2023, May). Analisis Dan Implementasi Metode *User Centered Design* (UCD) Pada Pembuatan Sistem Informasi Perangkat Mengajar Guru Berbasis Mobile. In Prosiding Seminar Nasional Inovasi dan Adopsi Teknologi (INOTEK) (Vol. 3, No. 1, pp. 12-25).

[9] Iqbal, M., Marthasari, G. I., & Nuryasin, I. (2020). Penerapan Metode UCD (*User Centered Design*) pada Perancangan Aplikasi Darurat Berbasis Android. Jurnal Repositor, 2(2).

[10] Tazkiyah, S., & Arifin, A. (2022). Perancangan UI/UX pada Website Laboratorium Energy menggunakan Aplikasi Figma. Jurnal teknologi terpadu, 8(2), 72-78.

[11] Ichsanudin, M. N., Yusuf, M., & Suraya, S. (2022). Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula. STORAGE: Jurnal Ilmiah Teknik dan Ilmu Komputer, 1(2), 1-8.

[12] Loyola-Gonzalez, O. (2019). Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. IEEE access, 7, 154096-154113.

[13] Pintelas, E., Livieris, I. E., & Pintelas, P. (2020). A grey-box ensemble model exploiting black-box accuracy and white-box intrinsic interpretability. Algorithms, 13(1), 17.

[14] Wintana, D., Pribadi, D., & Nurhadi, M. Y. (2022). Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing. Jurnal Larik Ladang Artikel Ilmu Komputer, 2(1), 8-16.

[15] Ramadan, A. C., Panduardi, F., & Febrita, R. E. (2024). Sistem Informasi Program Cover Partner Berbasis Web pada CV Dewangkara Pujangga Sejahtera. INFOMATEK: Jurnal Informatika, Manajemen dan Teknologi, 26(1), 113-124.