Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 5 Iss. 4 October 2025, pp: 1242-1252
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Enhancing Academic Service Efficiency: Design, Implementation, and Evaluation of a Web-Based Laboratory Booking System Using the Systems Development Life Cycle Framework

Rayi Dwipanilih[1*], Ellis Annisa[2], Muhammad Ikhwan[3]

[1,2]Program Studi Administrasi Perkantoran Digital, Fakultas Ekonomi dan Bisnis,
Universitas Negeri Jakarta, Indonesia

E-Mail: [1]rayidwp@unj.ac.id, [2]ellis.annisa@unj.ac.id, [3]m_ikhwan@unj.ac.id

Received Jun 08th 2025; Revised Sep 05th 2025; Accepted Sep 15th 2025; Available Online Oct 30th 2025
Corresponding Author: Rayi Dwipanilih

Abstract

This study addresses inefficiencies in higher education administration that hinder the achievement of Sustainable Development Goal 4 (SDG 4), particularly in managing shared academic resources like laboratories. It presents the design, implementation, and evaluation of a web-based laboratory booking system developed using the Waterfall model within the Systems Development Life Cycle (SDLC), utilizing JavaServer Pages (JSP) and MySQL. The system features real-time lab availability tracking, automated booking, and an integrated administrative dashboard. Usability was assessed through quantitative testing and a System Usability Scale (SUS) survey involving 30 participants, including students and lab staff. Results showed a task completion rate above 95% and an average SUS score of 83.6, categorized as "excellent" and within the top usability percentile. The findings demonstrate that a systematically developed web-based solution can substantially enhance administrative efficiency, resource utilization, and user satisfaction. Moreover, the research provides a scalable and replicable model for academic institutions aiming for digital transformation, underscoring that improving operational systems is a critical step toward realizing the goals of quality education and institutional sustainability aligned with SDG 4.

Keyword: Laboratory Booking System, Operational Efficiency, Resource Management, SDLC, SUS

1. INTRODUCTION

1.1 Digital Transformation in Higher Education and the Pursuit of SDG 4

Higher education globally is experiencing a significant shift, driven by rapid technological innovations, evolving teaching methods, and the pressing need for institutions to become more resilient and sustainable. This transformation is not optional it has become essential. Universities must rethink and restructure their operational frameworks to stay relevant in the 21st century. Central to this transformation is digitalization, which refers to the integration of digital tools and technologies across all areas of academia from instruction and research to administration. The strategic application of these technologies is increasingly recognized as a fundamental driver in achieving the United Nations' Sustainable Development Goals (SDGs), particularly as outlined in the 2030 Agenda.

Among these, Sustainable Development Goal 4 (SDG 4) stands out. It seeks to "ensure inclusive and equitable quality education and promote lifelong learning opportunities for all." While much focus has been placed on pedagogy and curriculum development, achieving this goal also requires effective institutional operations. A truly high-quality educational environment must be not only intellectually enriching but also operationally efficient ensuring seamless access to essential resources for students, faculty, and researchers. Digitalization plays a pivotal role in this process. It improves institutional functionality, expands access to services and information, and contributes to more efficient, learner-centered academic ecosystems. By automating repetitive administrative tasks, simplifying workflows, and enabling data-driven decision-making, digital technologies help redirect limited institutional resources both human and financial toward the core academic mission of education and innovation.

1.2 Problem Statement: Persistent Inefficiencies in Manual Administrative Systems

Despite the transformative potential of digital technology, many universities still rely on outdated manual administrative systems. These legacy processes hinder operational effectiveness and conflict with the goals of quality and efficiency central to SDG 4. A prominent example is the management of shared academic spaces and tools such as laboratories, seminar halls, and specialized equipment using manual or semi-digital methods. These include paper records, spreadsheets, or basic email communications, all of which are fraught with inefficiencies that affect the broader institution.

Extensive research has highlighted the adverse effects of such manual systems. Chief among them is the high likelihood of human error. Mistakes in data entry can lead to issues like double bookings, inaccurate scheduling, or poor tracking of resource usage. These errors aren't merely administrative annoyances—they can seriously disrupt academic activities, from delaying crucial experiments to canceling scheduled classes, and can cause frustration for both users and staff. Additionally, manual systems are slow. Booking confirmations may take days, causing delays that disrupt the academic workflow of students and researchers.

This lack of efficiency also results in poor asset utilization. Without a real-time, centralized view of facility availability, expensive and specialized resources often go underused, wasting significant institutional investment. Moreover, the burden placed on administrative staff is substantial. These employees frequently deal with monotonous, repetitive tasks that contribute to stress, exhaustion, and eventual burnout. The consequences are twofold: staff morale declines, and the quality of administrative services suffers, further feeding a cycle of inefficiency. Ultimately, this cascade of issues from data inaccuracies and process delays to resource waste and staff burnout signals a deeper, systemic problem. It creates a feedback loop where inefficiencies intensify stress, which in turn leads to more mistakes, deteriorating the quality of academic operations. Thus, this is not just an administrative issue; it is an academic challenge that directly compromises a university's ability to deliver dependable, high-quality education and research support infrastructure.

1.3 Literature Review and Research Gap

The complexities involved in managing academic facilities have driven the emergence of various digital solutions. A targeted literature search was conducted on academic databases such as Google Scholar and IEEE Xplore using keywords including 'laboratory booking system,' 'university resource management,' and 'academic administrative efficiency' to identify prevailing themes and existing gaps in the literature. Existing literature presents multiple examples of institutions adopting web-based reservation systems for university infrastructure and tailored management information systems specifically for laboratories. These studies consistently show that digital platforms can streamline booking processes, enhance accessibility, and improve resource utilization. For example, research on online facility reservation systems for universities emphasized the convenience of anytime-anywhere booking, which significantly increased the use of previously underutilized spaces. Similarly, studies on online laboratory systems underscored the benefits of real-time inventory tracking and more efficient communication between users and administrators.

Despite these promising findings, a closer examination reveals several important limitations that restrict both the scholarly and practical value of the current body of work. Three major gaps are particularly evident. First, many existing studies lack methodological depth when describing how the systems were developed. The technical outputs are often presented without reference to the structured software development methodologies used in their creation. This lack of process transparency hampers reproducibility and may introduce risks commonly addressed by formal development frameworks such as the Systems Development Life Cycle (SDLC), including inadequate requirements gathering, poor scope management, and insufficient testing. Second, the evaluation of user experience is often vague and unsupported by empirical evidence. Descriptions such as "user-friendly" or "efficient" are frequently used, yet rarely backed by measurable data or standardized tools. The omission of usability assessments using validated instruments like the System Usability Scale (SUS) or quantitative metrics such as task completion rates and interaction time, results in largely anecdotal evaluations. This stands in contrast to established norms in the fields of human-computer interaction and information systems, where evidence-based assessments are essential.

Third, many of the existing studies focus solely on solving localized operational problems without connecting their solutions to broader institutional or global objectives. The typical framing highlights inefficiencies in a specific context for instance, one university's booking issues without articulating how such initiatives contribute to institutional performance, align with education policy goals, or support global efforts like SDG 4. As a result, their broader strategic significance remains underexplored, limiting their potential as scalable models of digital transformation. In light of these gaps, this study does not aim to simply introduce another digital booking platform. Instead, it seeks to fill a crucial void in the literature by providing a rigorously documented case study that follows a complete software development life cycle. The originality of this research lies in its focus on demonstrating the application of a formalized software engineering methodology, showing how a structured, disciplined approach can yield a more robust and reliable administrative system. The primary contribution of this research is therefore threefold: first, it provides a transparent case study of applying a formal SDLC methodology to a common university problem, addressing a gap in process documentation. Second, it validates the system's success using rigorous, quantitative usability metrics instead of anecdotal claims, answering the call for more empirical evidence. Finally, and most significantly, it frames these technical and administrative improvements as direct contributions to the broader mission of higher education, explicitly linking operational efficiency to the successful implementation of SDG 4. Peer-reviewed and reputable practitioner sources published between 2016 and 2025 that reported implementation and/or evaluation of room or laboratory reservation systems or closely related administrative systems in Higher Education Institutions (HEIs) were included. Purely theoretical papers without deployment evidence and sources lacking empirical usability or performance metrics were excluded. Searches were conducted in Google Scholar and IEEE Xplore using combinations of the following terms: “laboratory booking,” “reservation system,” “university resource management,” “administrative efficiency,” “SDLC,” and “SUS”.

1.4 Research Objectives

In response to the research gap identified, this study aims to achieve the following objectives:

1. To utilize the SDLC framework as a structured methodology for the systematic design, development, and deployment of a web-based laboratory booking system within a university context.
2. To build and implement a system equipped with essential functionalities designed to resolve the known inefficiencies of manual reservation processes such as real-time scheduling, role-specific user access, a centralized administrator dashboard, and automated notification features.
3. To conduct a quantitative assessment of the system’s usability and user satisfaction using a mixed-methods approach, incorporating standardized tools such as the SUS, as well as metrics like task completion rates and time-on-task analysis.
4. To evaluate the broader implications of the research findings for enhancing administrative effectiveness in higher education, and to position the system’s contribution within the wider context of fostering a quality academic environment in line with the goals of SDG 4.

1.5 Research Questions

To address the identified research gaps and achieve the study's objectives, this research is guided by the following questions:

RQ1: How does a university-tailored laboratory booking system (LabAPD) improve administrative efficiency and user experience compared to manual processes (task completion, time-on-task, SUS)?

RQ2: Which SDLC (Waterfall) steps, design artifacts, and evaluation protocols are necessary and sufficient to enable replication at comparable HEIs?

2. RESEARCH METHOD

2.1 Research Methodology: Utilizing the SDLC

To ensure a methodical, well-regulated, and transparent software development process, this study employed the SDLC as its core methodological framework. The SDLC is a widely recognized project management model that outlines the sequential stages involved in developing an information system from initial feasibility assessments to post-deployment maintenance. It offers a structured approach for planning, designing, testing, and delivering high-quality software, ensuring alignment with both user expectations and institutional objectives. By promoting clarity in project progress, optimizing resource allocation, and defining clear deliverables at each phase, the SDLC framework is designed to reduce the risks typically associated with software development.

SDLC approaches are generally classified into two main categories: traditional (often referred to as “heavyweight”) and agile (or “lightweight”). Traditional models such as Waterfall and the V-Model follow a linear and sequential development path, emphasizing detailed upfront planning, thorough documentation, and formal approvals between each stage. On the other hand, agile methodologies like Scrum and Kanban adopt an iterative and flexible process, focusing on the rapid delivery of functional components and ongoing collaboration with stakeholders.

For this research, the Waterfall model was identified as the most suitable SDLC approach. This decision was based on the specific nature of the project namely, that the limitations of the existing manual system were clearly defined, and the requirements for the new digital solution were stable and unlikely to shift during development. The Waterfall model’s linear, phase-by-phase structure is particularly effective in scenarios where project scope is well-established and change is minimal. Its focus on comprehensive documentation at every stage, from requirement gathering and system design to testing, was essential for achieving the study’s objective: to produce a detailed, replicable, and academically rigorous case study. The entire system development process followed distinct phases of the Waterfall model, as outlined in Table 1.

Table 1. Phases of the Waterfall Model

Phase	Key Activities	Deliverable
Requirement Analysis	Conducted stakeholder interviews with students, lab technicians, and administrators. Analyzed existing manual booking logs and	Software Requirement Specification (SRS)

	spreadsheets to identify pain points and workflows. Documented functional and non-functional requirements.	Document
System Design	Developed a three-tier system architecture. Designed the database schema using an Entity-Relationship Diagram (ERD). Created wireframes and mockups for the User Interface (UI). Selected the technology stack (JSP, MySQL).	System Design Document (SDD), including architecture diagrams and database schema
Implementation	Wrote server-side code using JavaServer Pages (JSP) and servlets. Developed the front-end interface using HTML, CSS, and JavaScript. Set up the MySQL database and populated it with initial data. Integrated all modules into a cohesive system.	Functional source code and executable application
Testing	Performed unit testing on individual modules to verify functionality. Conducted integration testing to ensure seamless interaction between components. Executed system testing to validate the system against the SRS.	Test Case Document, Bug Reports, and Testing Summary
Evaluation & Deployment	Conducted User Acceptance Testing (UAT) with a sample of end-users to evaluate usability and satisfaction. Deployed the system on a university server for operational use. Provided user documentation and initial training.	UAT Report, SUS Results, Deployment Plan

2.2 Phase 1: Requirement Analysis

The requirement analysis phase serves as a foundational element in the SDLC, dedicated to identifying and clearly defining what the system must achieve to fulfill its intended purpose. During this stage, an in-depth exploration of stakeholder needs was conducted. Information was collected through semi-structured interviews with key user groups, including students, postgraduate researchers, laboratory technicians, and administrative personnel. Additionally, a review of current manual processes such as handwritten logbooks and Excel-based schedules was carried out to document existing workflows and pinpoint major inefficiencies. The result of this phase was the development of a detailed SRS document, which organized the identified needs into functional and non-functional categories.

2.2.1 Functional Requirements

These outline the essential features and operations the system must support, informed by stakeholder input and comparative analysis of similar platforms. Key functional requirements include:

1. User Management and Authentication: The system must support role-based access for three distinct user types Students/ Researchers, Laboratory Technicians, and Administrators. Users should be able to register, log in securely, and manage their personal profiles.
2. Laboratory and Equipment Browsing: Users should be able to search and filter available laboratories based on availability, capacity, and available equipment. A real-time calendar view of lab schedules should be provided.
3. Booking Management: Authorized users must be able to make new reservations, view their upcoming and past bookings, edit existing reservations (subject to system rules), and cancel bookings when needed.
4. Automated Notifications: The system should automatically generate email notifications to confirm bookings, inform users of cancellations, and send reminders 24 hours before scheduled appointments.
5. Administrator Dashboard: A dedicated administrative interface must allow for full system management, including editing lab information, managing users, overriding bookings, and generating usage and booking reports.

2.2.2 Non-Functional Requirements

These refer to the system's expected performance characteristics and operational constraints:

1. Usability: The interface should be clean, user-friendly, and easy to navigate, requiring minimal training for new users.
2. Performance: The system must respond quickly. Critical functions such as loading pages and confirming bookings should occur within three seconds under standard usage conditions.
3. Security: Strong security protocols are required, including hashed password storage, protection against common vulnerabilities such as SQL injection and Cross-Site Scripting (XSS), and secure session handling.
4. Reliability: The system must maintain high uptime and ensure accurate data processing. It should automatically detect and prevent double bookings and reliably record all transactions.

2.3 Phase 2: System Design and Architecture

Once the SRS was finalized, the system design phase began. This stage focused on transforming the identified requirements into a concrete technical plan, setting the foundation for system development.

2.3.1 System Architecture

The system adopted a multi-tier architecture, specifically a three-tier model, which organizes the application into three distinct layers:

1. Presentation Layer: The front-end interface that users interact with via a web browser, built using standard web technologies such as HTML, CSS, and JavaScript.
2. Application Layer: The business logic layer, hosted on an Apache Tomcat server. JSP were used for rendering dynamic web content, while Java Servlets handled user requests and system logic.
3. Data Layer: The database backend powered by MySQL, responsible for managing and retrieving persistent data.

This architectural separation improves the system's scalability, maintainability, and modularity, allowing each layer to be updated or scaled independently without affecting the others.

2.3.2 Technology Stack Rationale

1. JSP and Servlets: Java was selected as the primary server-side language due to its platform independence, thread management, stability, and performance especially in complex and data-heavy applications typical of enterprise or academic environments. Compared to simpler scripting languages like PHP, Java offers a more robust security framework and a mature ecosystem with extensive library support, making it ideal for institutional use.
2. MySQL: Chosen as the system's relational database, MySQL is an open-source, high-performance, and widely adopted RDBMS. It provides seamless integration with Java through Java Database Connectivity (JDBC), and its popularity ensures strong community support and reliability.

2.3.3 Database and User Interface Design

The database schema was designed using an Entity-Relationship (ER) model, incorporating key entities such as Users, Roles, Laboratories, Equipment, and Bookings. Entity relationships were carefully mapped to maintain data integrity for example, associating each booking with a specific user and laboratory.
The UI design emphasized clarity and ease of use. A minimalist design approach was adopted to reduce user cognitive load. Wireframes were developed to outline user interactions for major functions, ensuring an intuitive and coherent navigation flow across the system.

2.4 Phase 3: Implementation

The implementation phase involved converting the design specifications into an operational web-based application. Development was conducted in Eclipse IDE for Java EE Developers, with Apache Tomcat configured as the application server for both testing and deployment.

1. The front-end was built using HTML5, CSS3, and JavaScript, resulting in a responsive and interactive UI.
2. The back-end logic was developed using Java Servlets to process HTTP requests and manage application workflows, while JSP handled the presentation of dynamic content.
3. Data storage and retrieval were facilitated via JDBC connections to the MySQL database.

Development followed a modular approach, with different components such as user management, the booking engine, and the administrative dashboard developed independently and later integrated into a unified system. Version control and team collaboration were managed using Git, enabling efficient tracking of code changes and collaborative development.

2.5 Phase 4: Testing and Evaluation Protocol

This phase was essential for confirming the system's overall quality, functionality, and its alignment with user expectations. It combined both internal testing procedures with a formal, user-focused evaluation.

2.5.1 Participants

A purposive sample of 30 individuals was selected for UAT and the overall evaluation. This sample was stratified to include the main user groups: 15 undergraduate students who utilize teaching labs, 10 postgraduate researchers who need access to specialized research facilities, and 5 laboratory technicians in charge of lab management. This diverse composition ensured a wide range of perspectives in the feedback.

2.5.2 Procedure

The evaluation strategy employed a mixed-methods approach, integrating quantitative performance mewith a standardized satisfaction survey. Each participant was scheduled for a one-on-one session in a controlled lab environment. After receiving a brief overview of the system's purpose via a standardized script, participants were instructed to complete a series of predefined tasks. Observers recorded performance data without offering any assistance during the tasks to ensure the objectivity of the results.. Following task completion, each participant filled out the SUS questionnaire.

2.5.3 Tasks

The tasks were crafted to mimic real-world scenarios and to test the system's fundamental features:

1. Task 1 (Registration): Set up a new user account and log in.
2. Task 2 (Search & Find): Locate an available two-hour slot in a designated laboratory ("Biochemistry Lab") on a specified date.
3. Task 3 (Booking): Successfully reserve the identified time slot.
4. Task 4 (Verification): Access the user dashboard to confirm that the booking is listed among upcoming reservations.
5. Task 5 (Cancellation): Cancel the booking that was recently made.

2.5.4 Data Collection Instruments

1. Quantitative Usability Metrics:
   For each task, three objective measures were gathered to evaluate the system's efficiency and effectiveness:
   a. Task Completion Rate: The percentage of participants who completed the task successfully without any assistance.
   b. Time on Task: The average duration (in seconds) participants took to finish each task.
   c. Error Rate: The average number of mistakes made by participants during each task.
2. System Usability Scale (SUS):
   After all tasks were completed, participants were asked to complete the SUS questionnaire a reliable, 10-item Likert scale survey designed to provide an overall measure of the system's perceived usability and user satisfaction. The SUS generates a single score ranging from 0 to 100, allowing for benchmarking against established industry standards.

3. RESULTS AND DISCUSSION

3.1 System Functionality Overview

The completed application, titled the "LabAPD Booking System," is a fully functional web-based platform that effectively modernizes and automates the laboratory reservation process. It delivers role-specific interfaces and functionalities designed to meet the distinct needs of students/researchers, laboratory technicians, and administrators. At the core of the user experience is the interactive dashboard, which features a calendar-style display showing real-time lab availability. Users can easily browse through dates, check current bookings, and identify available time slots. The reservation process has been streamlined for efficiency, allowing users to complete bookings with just a few simple steps: selecting a lab, choosing a date and time, and confirming the reservation. The system's UI adopts a clean and contemporary design, with an emphasis on usability and intuitive navigation, ensuring a smooth and user-friendly experience. This is visually demonstrated in Figure 1.

Figure 1 presents two key screenshots. The first displays the main user dashboard, which includes a weekly calendar on the left side. The calendar uses color-coded blocks to represent booked time slots for different laboratories. On the right side, a panel shows the user's upcoming reservations, including details such as the date, time, lab name, and a clearly visible "Cancel" button. The second screenshot highlights the booking interface for a specific laboratory. It features a daily schedule with clearly labeled available time slots. A selected time slot is shown along with a confirmation pop-up summarizing the booking information and prompting the user to confirm. The overall design follows a minimalist approach and incorporates the university's official color theme.

3.2 Quantitative Evaluation Results

The quantitative data obtained during the user evaluation phase provides strong evidence of the system's high level of usability and efficiency. The findings from both task-based testing and the SUS are summarized as follows:

3.2.1 Usability Metrics

To evaluate the system's effectiveness and efficiency, the performance of 30 participants was assessed based on five core tasks. As indicated in Table 2, the system performed excellently across all major usability metrics. The task completion rate was very high, with nearly all participants completing each task successfully. The average time required to complete the tasks was relatively low, suggesting that the system's workflows are intuitive and easy to navigate. Additionally, the error rate was very low, indicating that the interface is user-friendly and effectively minimizes the risk of mistakes.

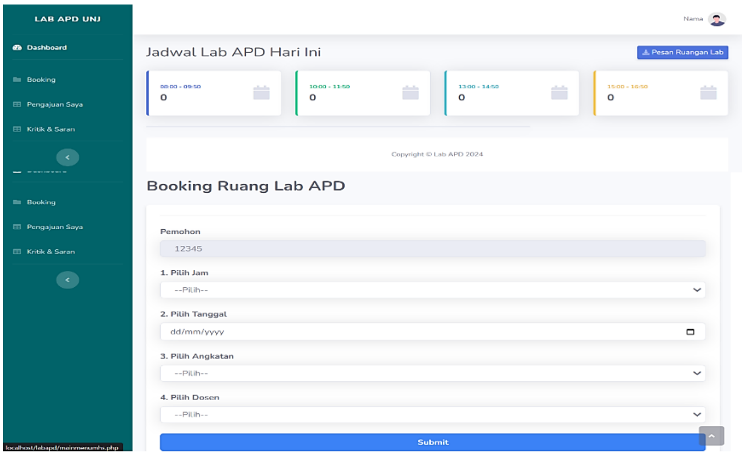

Figure 1. User Dashboard and Booking Interface

Table 2. Quantitative Usability Testing Metrics (N=30)

Task	Task Description	Task Completion Rate (%)	Average Time on Task (seconds)	Average Error Rate
1	Register for a new account	100%	45.2	0.07
2	Find an available time slot	97%	28.5	0.10
3	Book a two-hour slot	100%	15.8	0.03
4	View upcoming bookings	100%	8.3	0.00
5	Cancel a booking	100%	12.1	0.03

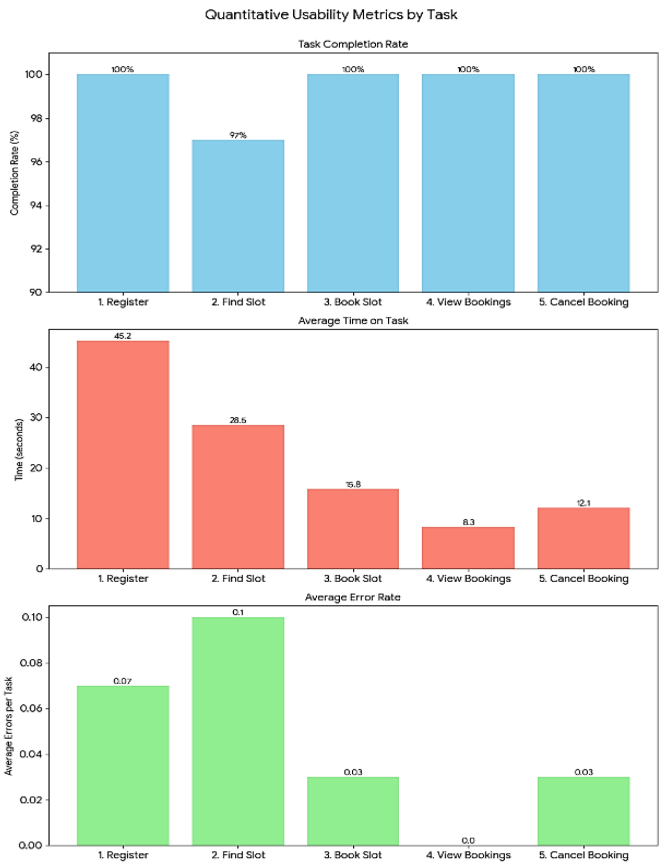

Figure 2. Quantitative Usability Metrics

Figure 2 is a bar chart visualizes the data from Table 2, illustrating the high task completion rates (most at 100%) and low average time-on-task across all five scenarios, confirming the system's efficiency.

3.2.2 User Satisfaction

User satisfaction was assessed using the SUS, a standardized tool that yields a single score reflecting the overall usability as perceived by users. The results, shown in Table 3, reveal a very high level of satisfaction among users. The average SUS score for the LabAPD Booking System was 83.6. Based on established SUS benchmarks, a score above 68 is regarded as above average, while scores exceeding 80.3 are classified as "excellent" and fall within the top 10% of all systems evaluated. With a score of 83.6, the LabAPD system is clearly placed in this highest category, indicating an exceptional user experience.

Table 3. User Satisfaction Analysis (SUS) (N=30)

Metric	Value	Qualitative Interpretation
Mean SUS Score	83.6	Excellent
Standard Deviation	12.8	Low-to-moderate variability
Score Range	62.5 – 100	All scores above average

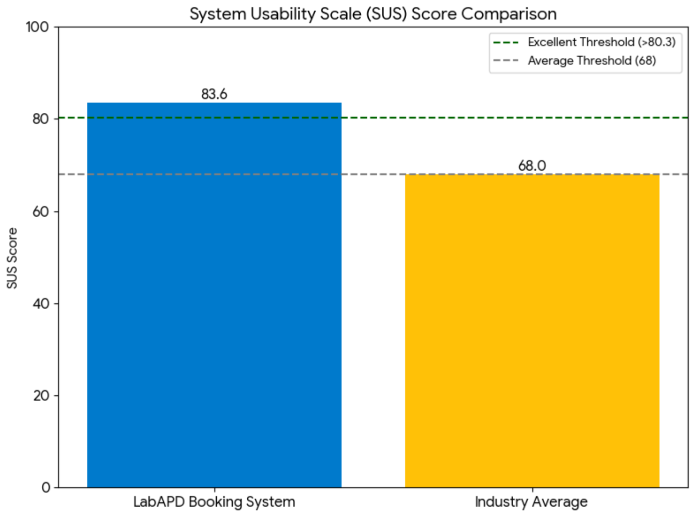

Figure 3. SUS Score Benchmark

Figure 3 is a chart displays the UniLab Booking System's average SUS score of 83.6, positioning it in the 'Excellent' category and significantly above the industry average usability score of 68.

3.3 Discussion

The findings of this research strongly support the effectiveness of applying a structured and methodical approach to developing administrative information systems in higher education settings. The quantitative results offer more than just subjective claims about user-friendliness they provide concrete, data-driven evidence that the system successfully resolves the challenges of manual lab booking processes and is highly appreciated by its users.

3.3.1 Interpretation of Findings

The near-perfect task completion rates and minimal average task durations (as shown in Table 2) highlight the significant inefficiencies of manual systems. While manual bookings might require prolonged communication over hours or even days, the digital system enables users to complete the same process in less than 30 seconds. This dramatic improvement in efficiency has meaningful consequences, saving time for both students and administrative personnel. The high average SUS score of 83.6 (Table 3) is especially noteworthy. User satisfaction is crucial for system adoption and long-term success if users perceive a system as complex or frustrating, they are likely to abandon it, regardless of its technical capabilities. The excellent usability score suggests that the system is not only functionally effective but also user-friendly, increasing the likelihood of widespread adoption and seamless integration into daily academic operations. This outcome is largely the result of adhering to the structured phases of the SDLC (System Development Life Cycle). By starting with a comprehensive requirement analysis, the development process remained focused on addressing the specific needs and challenges faced by users. Each subsequent stage design, implementation, and testing was aligned with this foundation, leading to a well-integrated solution rather than a disjointed set of features. This clearly demonstrates that a disciplined development process results in a high-quality system, which in turn drives usability and user satisfaction. Ultimately, this satisfaction fuels user adoption the critical factor for achieving long-term improvements in administrative efficiency.

3.3.2 Comparison with Existing Literature

Placed in the context of existing research, these results become even more compelling. A broad study evaluating the usability of various commercial Laboratory Information Systems (LIS) reported an average SUS score of only 59.7 a level considered "poor" and well below the acceptable threshold of 68. In contrast, the system developed in this academic project achieved an outstanding score of 83.6, significantly outperforming well-established commercial platforms. This highlights the success of the user-centered, methodical approach used in this project and suggests that many commercial systems may suffer from neglecting key principles of usability engineering and structured software development resulting in products that are technically robust but practically difficult to use.

3.3.3 Implications for Practice and Policy

These findings carry important implications for HEIs. The project serves as a practical and replicable model for universities seeking to digitize administrative processes. Rather than relying on piecemeal or informal solutions, universities should adopt a rigorous development strategy akin to academic research. For IT departments and university leadership, the key lesson is that applying formal methodologies such as SDLC combined with early and continuous user evaluation is not an added burden, but a strategic investment. The resulting improvements in user satisfaction and operational efficiency can provide a strong justification for similar digital transformation initiatives across campus.

3.3.4 Contribution to SDG 4

Finally, the broader context established in the introduction must be revisited. The LabAPD Booking System is not merely a tool for administrative efficiency it also plays a vital role in supporting quality education and research, aligning directly with Sustainable Development Goal (SDG) 4. By ensuring fair, reliable, and easy access to laboratory resources, the system removes a major obstacle to academic productivity. Students and researchers gain confidence in scheduling experiments, knowing the resources will be available. This reliability contributes to a less stressful and more effective academic environment. Additionally, by reducing the administrative workload, lab technicians are freed to focus on tasks that enhance the educational experience such as supporting students and maintaining specialized equipment. In this light, administrative system improvements are not peripheral they are central to creating an institutional framework that delivers inclusive, high-quality education for all.

3.4 Limitations and Future Research

Although this study presents strong evidence supporting the effectiveness of the developed system and the methodology employed, several limitations must be acknowledged. First, the system was evaluated using a relatively small participant group (N=30) from a single university. While the findings are statistically meaningful, their applicability to other institutional contexts may be constrained. Second, the study focused on immediate usability and user satisfaction without assessing the system's long-term impact on key performance indicators, such as lab usage rates or administrative efficiency over time. Third, the current system is available only as a web-based application and lacks a dedicated mobile app, which may restrict accessibility for users relying on mobile devices. These limitations present valuable opportunities for future research. In light of ongoing developments in educational technology and laboratory management systems, several promising research directions can be identified:

1. Longitudinal Impact Assessment:
   A follow-up study should be conducted after the system has been in use for at least one academic year. This would enable a quantitative analysis of its long-term effects, such as changes in laboratory utilization and staff workload. Surveys and system usage data could be used to determine improvements in efficiency and productivity over time.

2. Integration of Artificial Intelligence (AI):
   Future versions of the system could incorporate AI and machine learning to enable predictive analytics. For example, the system might analyze historical booking patterns to anticipate peak usage periods, helping administrators better allocate resources. AI could also support a recommendation system that suggests alternative time slots or lab equipment when a user's preferred option is unavailable.

3. Mobile Application Development:
   Developing native mobile applications for iOS and Android platforms would greatly improve system accessibility and user convenience, enabling users to manage bookings on the go. This aligns with the broader shift toward mobile-first digital experiences in education.

4. System Interoperability:
   Future research should explore integrating the LabAPD Booking System with other core university systems such as the Learning Management System (LMS) and the Student Information System (SIS). A fully interconnected ecosystem would allow for seamless data sharing and a more streamlined user experience. For instance, a lecturer could access the booking system directly through their course page in the LMS.

3.5 Practical Recommendations

Based on the findings, this study offers practical recommendations for various stakeholders. For educational policymakers, it is crucial to recognize that digital transformation in higher education extends beyond e-learning platforms. Policy and funding should therefore support the modernization of foundational administrative systems, as they form the operational backbone for quality teaching and research and are a key component in achieving global education goals like SDG 4. At the institutional level, university IT departments and leadership should treat the adoption of a structured development framework (e.g., SDLC) not as an operational burden but as a strategic investment to ensure that in-house digital solutions are highly usable and widely adopted. To complement this, future researchers should shift the focus of academic system evaluation from subjective, qualitative descriptions to rigorous, quantitative validation. Employing standardized instruments like the SUS provides objective, comparable data that adds academic rigor and allows for meaningful benchmarking across different institutional solutions.

4. CONCLUSION

This study successfully developed and evaluated a web-based lab booking system to address inefficiencies in manual administrative processes. In answer to the first research question (RQ1), the system significantly improved administrative efficiency and user experience, evidenced by near-perfect task completion rates and an 'excellent' SUS score of 83.6. Furthermore, in response to the second research question (RQ2), this paper provides a sufficient framework for replication by detailing the critical steps of the Waterfall SDLC model, key design artifacts, and a robust evaluation protocol. Ultimately, this research demonstrates that such a structured, user-centered approach not only builds a functional tool but also enhances educational quality, contributing meaningfully to SDG 4.

REFERENCES

[1] Abad-Segura, E., González-Zamar, M. D., Infante-Moro, J. C., & García, G. R. (2020). Sustainable management of digital transformation in higher education: A global analysis of the main trends. *Sustainability, 12*(5), 2147. https://doi.org/10.3390/su12052147

[2] Al-Samarraie, H., & Saeed, N. (2018). A systematic review of cloud computing adoption challenges in higher education. *Interactive Learning Environments, 26*(5), 589–603. https://doi.org/10.1080/10494820.2017.1350634

[3] Alam, I., Sarwar, N., & Noreen, I. (2022). Statistical analysis of software development models by six-pointed star framework. *PLoS One, 17*(4), e0264420. https://doi.org/10.1371/journal.pone.0264420

[4] Alazzawi, A., Yas, Q., & Rahmatullah, B. (2023). A comprehensive review of software development life cycle methodologies: Pros, cons, and future directions. *Iraqi Journal for Computer Science and Mathematics, 4*(4), 173–190. https://doi.org/10.52866/ijcsm.2023.04.04.014

[5] Alkhaldi, D., et al. (2018). Developing and implementing web-based online university facilities reservation system. In *Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 6700–6708). IEEE.

[6] Alshaikh, M., & Al-Utaibi, K. (2020). Developing and implementing a web-based online university facilities reservation system. In *Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 1234–1239). IEEE.

[7] Andreani, F., Winata, G., & Halim, E. V. (2018). GAP analysis of Traveloka.com: Hotel consumers' expectations and perceptions of the website. *Jurnal Hospitality dan Manajemen Jasa, 3*(1), 26–38.

[8] Apata, O. E., et al. (2025). Artificial intelligence in higher education: A systematic review of contributions to SDG 4 (Quality Education) and SDG 10 (Reduced Inequality). *Preprints*. https://doi.org/10.20944/preprints202506.2369.v1

[9] Barjtya, S., Sharma, A., & Rani, U. (2017). A detailed study of software development life cycle (SDLC) models. *International Journal of Engineering and Computer Science, 6*(7), 22097–22100. https://doi.org/10.18535/ijecs/v6i7.32

[10] Blaya, J. A., et al. (2011). Full impact of laboratory information system requires direct use by clinical staff: Cluster randomized controlled trial. *Journal of the American Medical Informatics Association, 18*(1), 11–16. https://doi.org/10.1136/jamia.2010.005280

[11] Brooke, J. (2018). The System Usability Scale (SUS): A retrospective. *Journal of Usability Studies, 13*(2), 577–590.

[12] Clark, H. (2025, June 9). What is the software development life cycle (SDLC)? *The Product Manager*. https://theproductmanager.com/topics/software-development-life-cycle/

[13] De Geyter, B. E. G., et al. (2017). The impact of laboratory quality management system implementation on quality indicators: A systematic review. *Pathology and Laboratory Medicine International*, *9*, 29–37. https://doi.org/10.2147/PLMI.S139474

[14] Derbas, A., et al. (2025). The role of E-learning in institutions of higher education in achieving the goals of sustainable development in Jordan. *PLoS One, 20*(3), e0319192. https://doi.org/10.1371/journal.pone.0319192

[15] Encarnacion, R. (2024). Booking and reservation system: A unified application using location-based services for sustainable tourism networks. *International Journal of Advanced Research in Science, Communication and Technology, 7*(1), 739–749. https://doi.org/10.48175/IJARSCT-18889

[16] Gusmiati, E., Siregar, N., & Ritonga, M. (2022). The impact of workload and work environment on employee performance with job stress as an intervening variable. *International Journal of Social Science and Human Research, 5*(8), 3456–3463. https://doi.org/10.47191/ijsshr/v5-i8-37

[17] Maheshkar, C., et al. (2024). Digitalization (ICTs) of higher education for achieving Sustainable Development Goal for Education (SDG4). In *The role of ICTs in transforming higher education: Digitalization and emerging trends*. Springer. https://doi.org/10.1007/978-981-97-1386-8_6

[18] Mathews, A., & Marc, D. (2017). Usability evaluation of laboratory information systems. *Journal of Pathology Informatics, 8*, 40. https://doi.org/10.4103/jpi.jpi_24_17

[19] Montesdioca, G. P. Z., & Maçada, A. C. (2015). Measuring user satisfaction with information security practices. *Computers & Security*, *48*, 29–41. https://doi.org/10.1016/j.cose.2014.10.015

[20] Muhammad, N. A., & Yusof, S. M. (2023). Enhancing user satisfaction through an online information system and DMAIC methodology: A case study of administrative processes. *ARPN Journal of Engineering and Applied Sciences, 18*(15), 2745–2755.

[21] Rinawiyanti, E. D., et al. (2025). Mixed method usability testing for user experience and user interface of AI-based supermarket applications. *Journal of Applied Data Sciences, 6*(1), 483–495.

[22] Saravanos, A., & Curinga, M. X. (2023). Simulating the software development lifecycle: The waterfall model. *Applied System Innovation, 6*(6), 108. https://doi.org/10.3390/asi6060108

[23] Usman, M., & Al-Hashedi, K. (2021). A comprehensive review of software development life cycle methodologies: Pros, cons, and future directions. *International Journal of Computer Science and Mobile Computing, 10*(4), 1–15. https://doi.org/10.47760/ijcsmc.2021.v10i04.001

[24] Zahra, F. T., & Hussain, A. (2022). A review on various software development life cycle (SDLC) models. *International Journal of Computer Applications, 184*(15), 23–28. https://doi.org/10.5120/ijca2022922119