

Development of Electrical Laboratory Information System using Model View Controller Architecture

Eka Fitrianto¹, Zen Resti Maulana^{1*}, Muhammad Ilhamdi Rusydi¹, Asrul Azani Mahmood²

¹Department of Electrical Engineering, Faculty of Engineering, Universitas Andalas, Padang, Indonesia

²Centre of Research & Field Services, University Malaysia Terengganu, Malaysia

*Corresponding author, e-mail: zenrestimaulana@gmail.com

Abstract— Effective and efficient laboratory inventory management is crucial to support academic and research activities in higher education. The manual system currently used to manage eight laboratories in the electrical engineering department often leads to various problems, such as recording errors, asset loss, and inaccuracy in reporting. This is the urgency of establishing an integrated information system to improve the accuracy and efficiency of laboratory inventory management. This research aims to digitize the laboratory inventory through the implementation of an application-based laboratory inventory management information system, namely Silab Elektro. In addition, this system also aims to increase transparency and accountability in laboratory asset management. This information system was developed using UML (Unified Modeling Language) modeling with MVC (Model-View-Controller) architecture. This research produces a laboratory information system that can manage equipment inventory more effectively and efficiently, as well as improve data accuracy.

Keywords: *Inventory Management Laboratory, Information Systems, Management Efficiency.*

This article is licensed under the [CC-BY-SA](#) license.

1. Introduction

Effective laboratory inventory management is a critical component in supporting academic and research activities in higher education institutions[1], [2]. In electrical engineering laboratories, inventory complexity is increased due to the diversity of equipment types, frequent borrowing activities, and the need to monitor item condition, location, and usage history [3], [4]. Inaccurate inventory records can result in equipment loss, inefficient practicum scheduling, and reduced service quality[1], [5].

At the Department of Electrical Engineering, Universitas Andalas, inventory management across multiple laboratories is still conducted using manual record-keeping. Borrowing and return transactions are documented in physical logbooks, which are prone to recording errors, data inconsistency, and limited traceability[1], [5]. These limitations hinder real-time monitoring, complicate audit processes, and reduce transparency in laboratory asset management[5].

Several studies have proposed digital inventory management systems to improve operational efficiency and data accuracy[3], [5]. However, most existing studies focus on general inventory systems or non-laboratory contexts and do not specifically address the operational characteristics of electrical engineering laboratories, such as practicum-based borrowing workflows, equipment mutation across laboratories, and compensation mechanisms for damaged items[3], [6], [7].

This study addresses this gap by developing a laboratory information system specifically tailored to the operational needs of electrical engineering laboratories. The proposed system, named

Silab Elektro, integrates inventory management, borrowing and return workflows, consumable usage tracking, and item relocation within a unified web-based platform. The system is designed using the Model–View–Controller (MVC) architectural pattern to ensure modularity, maintainability, and scalability[7], [8], [9].

In order to address the identified research gap, this study contributes by designing and implementing a laboratory inventory information system that explicitly reflects the real operational workflows of electrical engineering laboratories, including practicum-based borrowing, equipment relocation, consumable usage tracking, and compensation handling for damaged items. Unlike previous inventory management studies that focus on general organizational contexts or non-laboratory environments [3], [4], this system is specifically tailored to the complexity and operational characteristics of academic electrical laboratories. Furthermore, the application of the Model–View–Controller (MVC) architectural pattern ensures consistency between system design and implementation, enabling clear separation of concerns, improved maintainability, and scalability of the web-based application[7], [10]. The system’s functional reliability is validated through systematic black-box testing based on predefined user requirements, demonstrating that the proposed solution effectively meets operational needs while providing a robust foundation for future development and integration within higher education laboratory management systems[8], [11].

2. Method

This study adopts a Research and Development (R&D) approach, as the objective is not only to analyze problems but also to design, implement, and validate a laboratory information system tailored to the operational needs of electrical engineering laboratories. The R&D method is widely applied in information systems research to develop and evaluate technological artifacts in real-world contexts[8], [12]. Within this framework, the Waterfall software development model is used to structure development into sequential stages—requirements analysis, system design, implementation, testing, and maintenance—making it suitable for projects with stable functional requirements and clear scope[6], [13]. In addition, the system applies the Model–View–Controller (MVC) architectural pattern, which separates data management, application control, and user interface presentation to improve modularity and maintainability in web-based applications[7], [10]. The combination of Waterfall and MVC ensures traceability between requirements, design, and implementation while supporting systematic development and future system evolution.

2.1 Requirements Analysis

The requirements analysis phase aims to identify the functional and operational needs of the proposed laboratory inventory information system. In this study, requirements were gathered through direct observation, semi-structured interviews with laboratorians, and document analysis of existing manual inventory records. This triangulation approach ensures that the resulting requirements reflect actual laboratory workflows while reducing ambiguity and stakeholder misinterpretation[14], [15]. Observation focused on day-to-day activities such as inventory recording, borrowing and return procedures, and item relocation, while interviews captured user expectations and operational constraints. Document analysis provided insight into legacy transaction data and existing inventory structures. Based on this analysis, the identified needs were formalized into a set of functional system requirements, as summarized in Table 1, which serve as the foundation for subsequent system design and implementation stages[14], [16].

Table 1. Functional System Requirements of the Silab Elektro System

Req. ID	Requirement Description	Actor	Related Use Case	Related Data Entities
FR-01	The system shall authenticate users before granting access to system functionalities.	Student, Laboratorian	Log in	Users
FR-102	The system shall allow students to submit	Student	Apply for a	Student,

02	loan requests for laboratory goods.		loan of goods	Borrowing, Goods
FR-03	The system shall allow students to view their loan history.	Student	View loan history	Borrowing, Return
FR-04	The system shall display a list of available laboratory items.	Laboratorian	View the list of items	Goods, Category, Location
FR-05	The system shall allow laboratory assistants to add new laboratory items.	Laboratorian	Add items	Goods, Category, Location
FR-06	The system shall allow laboratory assistants to edit item specification details.	Laboratorian	Edit item specification details	Goods
FR-07	The system shall allow laboratory assistants to delete laboratory items.	Laboratorian	Delete items	Goods
FR-08	The system shall record item mutation transactions.	Laboratorian	Add item mutations	Mutation, Goods, Location
FR-09	The system shall display item mutation history.	Laboratorian	View item mutations	Mutation, Goods
FR-10	The system shall record consumable usage transactions.	Laboratorian	Add usage transactions	Usage, Goods
FR-11	The system shall display the usage history of consumable items.	Laboratorian	View the use of consumables	Usage
FR-12	The system shall display submitted loan applications.	Laboratorian	View loan applications	Borrowing
FR-13	The system shall allow laboratory assistants to approve or suspend loan applications.	Laboratorian	Approve / Suspend loan applications	Borrowing
FR-14	The system shall manage item return transactions.	Laboratorian	Managing returns	Return, Borrowing
FR-15	The system shall record compensation information for damaged or missing items.	Laboratorian	Compensation	Compensation, Return
FR-16	The system shall allow laboratory assistants to manage item categories.	Laboratorian	Add item category	Category
FR-17	The system shall allow laboratory assistants to manage item locations.	Laboratorian	Add item location	Location

Each requirement represents a specific operational need of the laboratory and is derived from observed workflows, user interactions, and data management processes. These requirements cover inventory data management, item condition and usage tracking, location mutation monitoring, borrowing and return workflows, consumable usage recording, and compensation handling for damaged items. The functional requirements listed in Table 1 are further represented through use case modelling and entity relationship modelling to ensure consistency between requirements analysis and subsequent system design stages[1], [5].

2.2 System Design

The data generated from the needs analysis stage is then used to design the system. Each data point from the needs analysis will be translated into an organized system structure[16]. In this research, the information system design process is carried out by creating use case diagram models, entity diagrams, and activity diagrams.

2.2.1 Use Case Diagram

The use case diagram illustrates interactions between system actors and system functions. Two primary actors are identified: laboratorian and students. Laboratorian are authorized to manage inventory data, process borrowing and return requests, and record item mutations, while students are permitted to submit borrowing requests and view transaction histories. Relationships between use cases are represented using <<include>> and <<extend>> associations, as shown in Figure 1, which indicate functional dependencies and optional process extensions within the system workflow. This modeling approach provides a structured representation of functional requirements by linking user roles with system operations, thereby supporting consistency between requirements analysis and subsequent system design stages in UML-based information system development[4], [17].

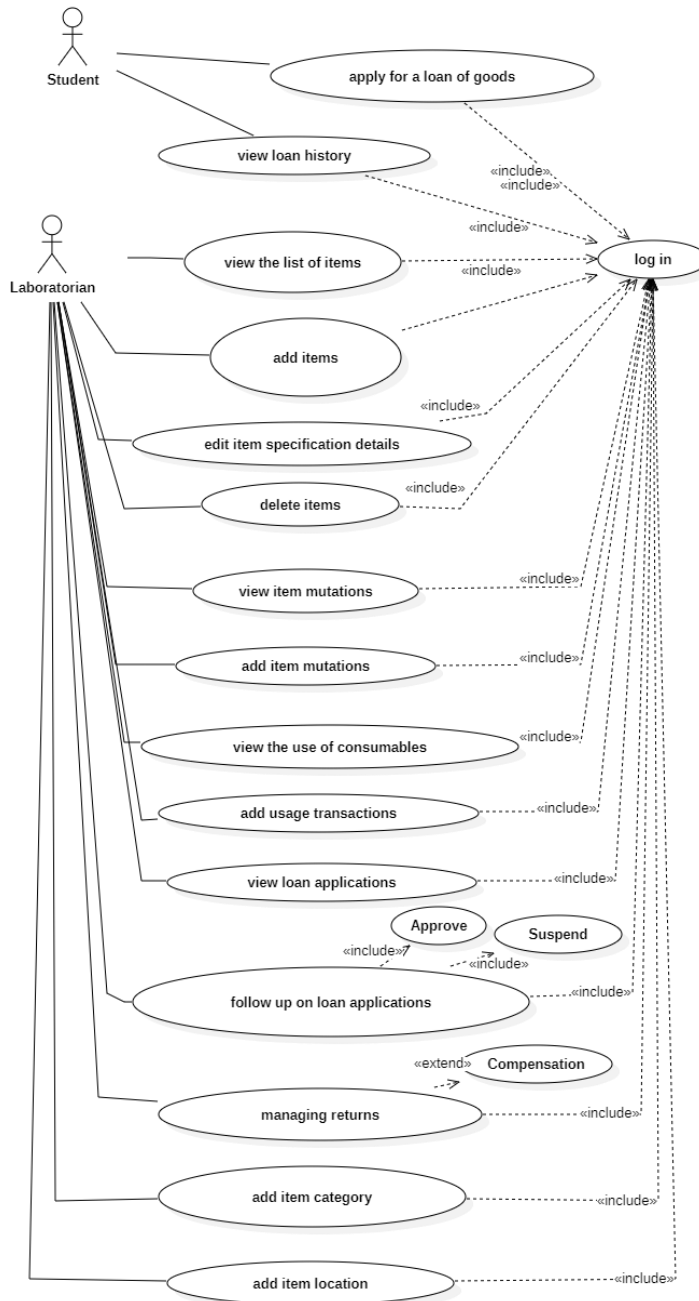


Figure 1. Use Case Diagram

2.2.2 Entity Relationship Diagram (ERD)

Based on the use case diagram presented in Figure 1, the development process proceeds from functional modeling toward data modeling to ensure that every identified system interaction is supported by an appropriate database structure. The use cases describing inventory management, borrowing and return transactions, mutation tracking, consumable usage recording, and compensation handling imply that the system must store, relate, and maintain consistent information across multiple operational activities. Therefore, an Entity Relationship Diagram (ERD) is constructed as a logical consequence of the use case analysis, translating user-oriented functional requirements into persistent data entities and their relationships. As shown in Figure 2, the ERD defines the data foundation required to implement the functionalities captured in the use case model, ensuring that actor interactions and system processes are consistently mapped into structured database components. This transition from use case representation to ER modeling supports traceability between requirements, system design, and implementation stages, which is a fundamental practice in information system development[17], [18].

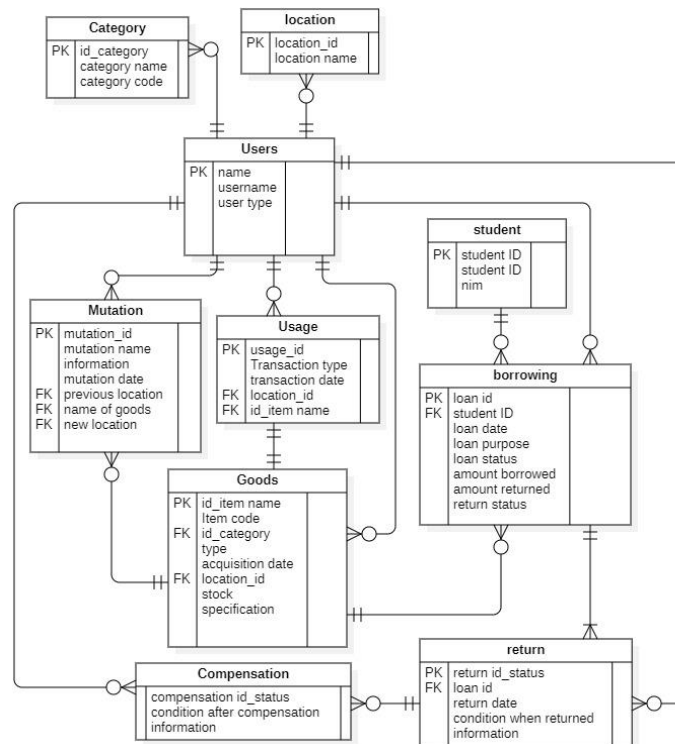


Figure 2. Entity Relation Diagram

2.2.3 Activity Diagram

Following the functional requirements captured in the use case model, the system design stage is further refined through the development of activity diagrams to represent the detailed operational behavior of key processes. While use case diagrams define *what* functions are provided and which actors interact with them, activity diagrams describe *how* these functions are executed within the system by modeling sequential actions, decision points, and responsibilities across system actors[15]. Therefore, the activity diagrams presented in Figure 3 to Figure 6 are derived directly from the core use cases to provide a more explicit representation of workflow execution, particularly for processes that involve validation, conditional branching, and multi-actor coordination. As shown in the login, item input, borrowing, and return activity models, each diagram translates the corresponding use case into structured action flows between students, laboratorians, and the system, ensuring that functional requirements are consistently mapped into implementable process logic. This refinement supports

traceability from requirements analysis to behavioral modeling and serves as a foundation for subsequent implementation and testing stages in UML-based information system development[7], [17].

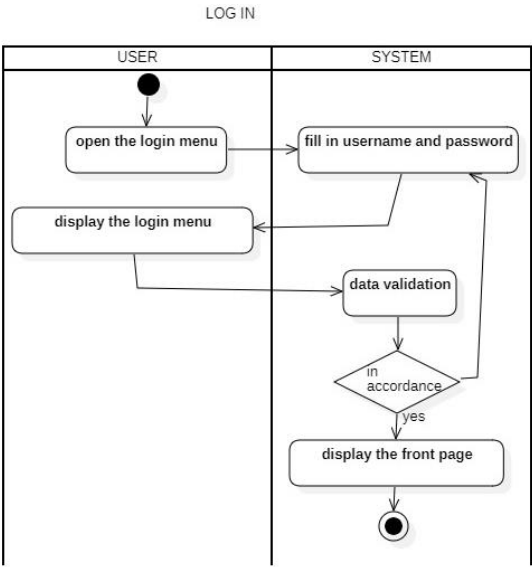


Figure 3. Login Activity Diagram

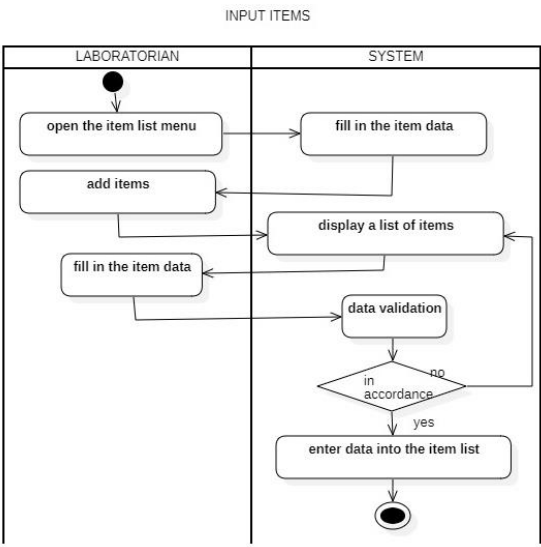


Figure 4. Input Goods Activity Diagram

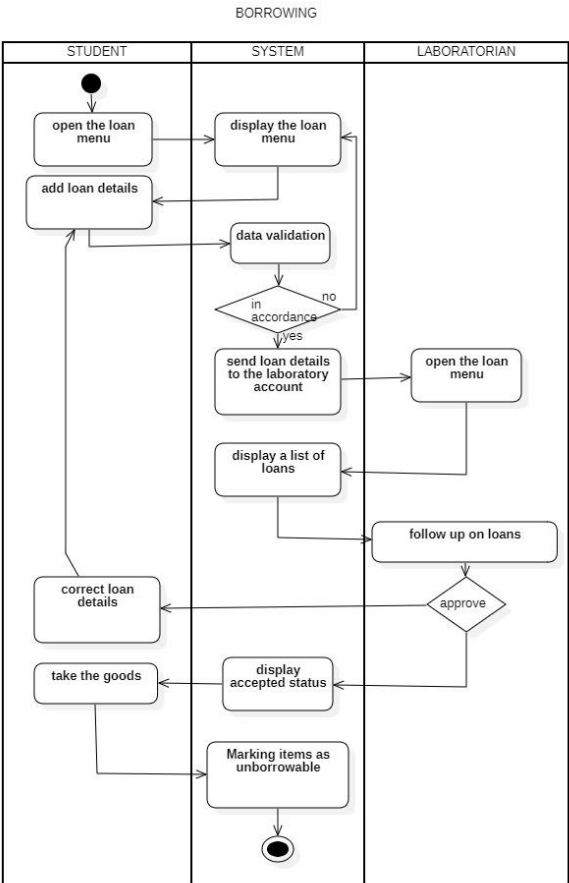


Figure 5. Loan Activity Diagram

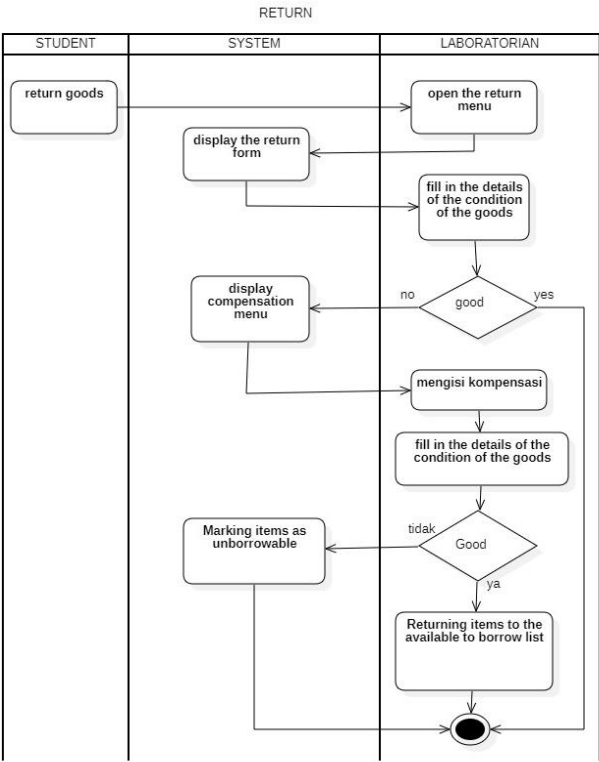


Figure 6. Return Activity Diagram

2.3 Implementation

The Silab Elektro system was developed as a web-based laboratory inventory information system using PHP as the server-side programming language and MySQL as the database management system. Laboratory operational data—including inventory records, borrowing and return transactions, consumable usage, and equipment mutation tracking—are managed through a centralized relational database to ensure real-time availability, traceability, and transparency. To enhance maintainability and scalability, the system applies the Model–View–Controller (MVC) architectural pattern, which is derived from the UML-based system design artifacts. The use case diagram defines actor–system interactions that are mapped into controller modules, the entity relationship diagram establishes the persistent data structure implemented in the model layer, and the activity diagrams guide workflow logic, validation, and decision processes within controllers. This integration of functional, structural, and behavioral modeling provides traceability from requirements to implementation and results in the modular MVC structure shown in Figure 7 [7], [10].

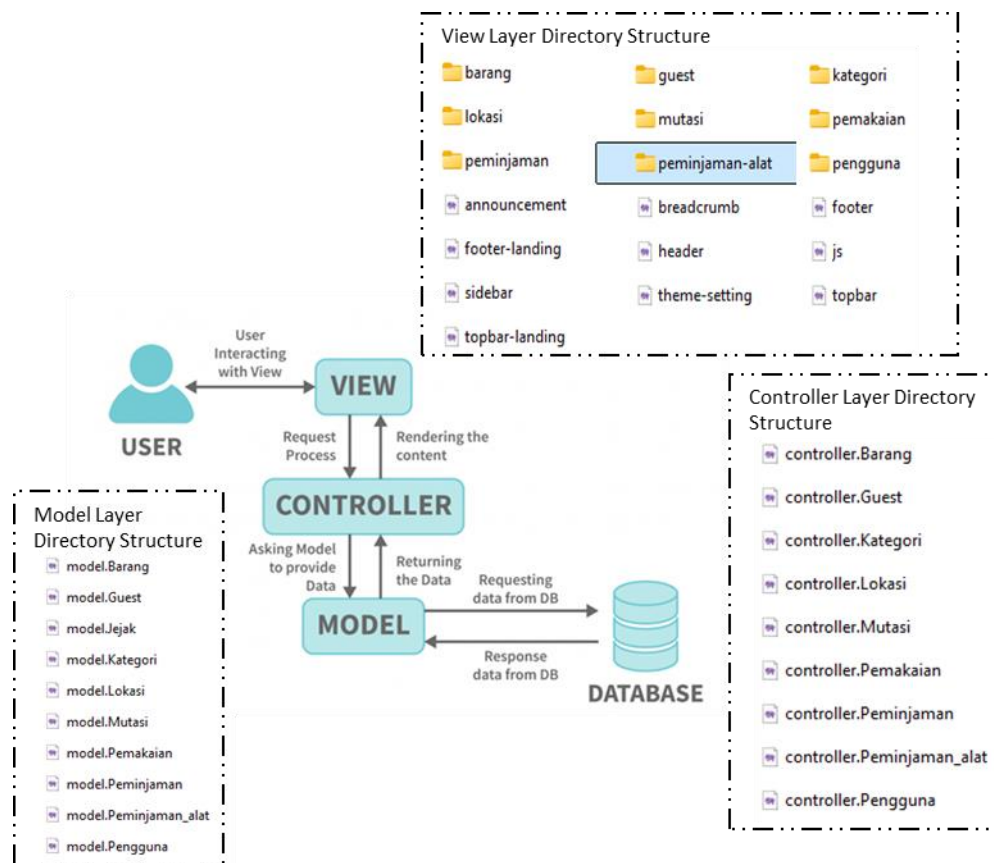


Figure 7. Structural representation of MVC implementation in Silab Elektro, including conceptual architecture and directory-level separation of concerns

The program-level implementation is further demonstrated by the controller code snippet presented in Figure 8, which illustrates a concrete controller–model–view interaction. The controller method performs session validation, prepares metadata for the user interface, invokes a borrowing-related model method to retrieve application data, and conditionally processes the returned results before rendering the appropriate view. This execution flow confirms that controllers act as intermediaries between models and views, enforcing separation of concerns and preventing direct database access from the view layer[1], [3], [19].

```

public function Home(): void
{
    logger( substr(string: strrchr(haystack: __CLASS__, needle: "\\"), offset: 1), __FUNCTION__, 'Membuka halaman Home/Depan' );

    session_check(self::$uri);

    $data['metadata']['title'] = 'Home';
    $data['breadcrumb']['page_title'] = 'Depan';
    $data['breadcrumb']['breadcrumb'] = ['Home' => false];
    $data['chosen']['chosen'][0] = true;

    $model_peminjaman = $this->loadModel('Peminjaman');
    $temp = $model_peminjaman->peminjamanNeedRespond();
    if ( $temp['jml'] > 0 ) {
        $data['chosen']['need_respond'] = $temp['jml'];
    }

    $this->view('able-pro/guest/home', $data);
}

```

Figure 8. Source-code example illustrating controller–model–view coordination in the Silab Elektro system implementation

Through the application of the Model–View–Controller structure, the Silab Elektro system is organized as a web-based application in which application responsibilities are distributed across model, controller, and view components. This implementation defines the structural and operational foundation of the system, which subsequently serves as the basis for functional validation and testing, as discussed in the following section.

2.4 Testing

Testing is carried out to ensure that the system runs according to the needs that have been previously analyzed [13]. The testing method used is black box testing, where each function is tested based on the output produced as shown in Table 2[20].

Table 2. Black Box Testing

Modules/Features	Test Case	Input	Expected Output	Status
User Login	Login with valid username and password	username / password	Enter home	√
User Login	Login with wrong password	incorrect username/password	The message "data is invalid" appears.	√
Student Login	Login with valid NIM and password	2210001 / pass123	Log in to the student dashboard	√
Add Location	Enter new location	"Microprocessor Lab"	Location added successfully	√
Add Location	Empty location	" "	Validation appears: "please fill out this field"	√
Add Category	Enter category name	"Measuring instrument"	Saved categories	√
Add Item	Enter all complete and valid data	Name, Location, Category, Stock	Item successfully saved	√
Add Item	Do not fill in the item name	" "	The validation appears "please fill out this field"	√
Use of Goods	Input the type of usage and quantity	"Internal Use" / 2	Transaction successfully recorded	√
Student Loans	Apply for a loan with available items	Item A, borrowing date	Submission successfully submitted	√
Student Loans	Submit stock items 0	Item B, stock = 0	Loan rejected, message appears	√

Loan Verification	Approve student applications	Click the “Approve” button	Status changed to approved	√
Loan Verification	Suspend application	Click “Suspend”, fill in the reason	Status becomes suspended, reason saved	√
Submission Revision	Student loan revision suspended	Change date or item	Submission updated and resubmitted	√
Return of Goods	Return the goods in good condition	Click “Restore”	Status back → finished	√
Return of Goods	Item returned damaged	Select condition: “Damaged”	The system asks for compensation input.	√
Compensation	Nominal input and compensation reasons	Rp. 200,000, “Cable damage”	Compensation data stored	√
Item Transfer	Transfer of goods to a new location	Goods A → Lab A to Lab B	Recorded mutation history	√
Logout	Click the logout button	–	Exit the system, return to the login page	√

2.5 Maintenance

After the system is deployed, improvements are made based on user feedback as well as bug fixes or feature adjustments if necessary. By using the R&D method and the Waterfall model, this research is expected to produce a laboratory information system that is functional, efficient, and meets user needs in an electrical engineering laboratory environment

3. Result and Discussion

This study successfully developed the Silab Elektro laboratory inventory information system as a web-based application to support inventory governance and operational workflows in electrical engineering laboratories. Following the Research and Development (R&D) approach and the Waterfall development stages, the implemented system represents the realization of the functional requirements identified during the requirements analysis phase (Table 1). The resulting system integrates inventory management, borrowing and return transactions, item mutation tracking, consumable usage recording, and compensation handling, which collectively address the limitations of manual inventory documentation.:

3.1 Inventory Management Module

The inventory management module provides structured storage of laboratory goods data, including item specifications, stock quantity, storage location, and condition status. As illustrated in Figure 9 and Figure 10, the system enables laboratorians to access detailed item information efficiently, improving traceability and reducing errors compared to manual record-keeping. New inventory items can be added through the input form shown in Figure 11, where submitted data are validated before being stored in the database. This module forms the foundation for subsequent operational processes such as borrowing and consumable usage:

#	KODE	NAMA BARANG	KATEGORI	JENIS	TANGGAL PEROLEHAN	LOKASI	STOK	ACTION
1	BHP2B12	BATERAI A2	Bahan Pakas Peralat	Tidak Ada	—	STORAGE DTE	8	Detail Edit Hapus
2	KL0890	POINTER	Laptop dan Komputer	Tidak Ada	27 October 2025	STORAGE DTE	1	Detail Edit Hapus
3	KL0889	POINTER	Laptop dan Komputer	Tidak Ada	27 October 2025	STORAGE DTE	1	Detail Edit Hapus
4	KL0888	POINTER	Laptop dan Komputer	Tidak Ada	27 October 2025	STORAGE DTE	1	Detail Edit Hapus
5	PP0804	STV32	Peralatan Praktikum	Tidak Ada	27 October 2025	Electrical Workshop	1	Detail Edit Hapus
6	PP0803	Unit of Measuring Capacitor 1200 pF	Peralatan Praktikum	Tidak Ada	—	Laboratorium Teknik Tegangan Tinggi (TTT)	1	Detail Edit Hapus
7	PP0802	Unit of Measuring Capacitor 1200 pF	Peralatan Praktikum	Tidak Ada	—	Laboratorium Teknik Tegangan Tinggi (TTT)	1	Detail Edit Hapus
8	PP0801	Unit of Measuring Capacitor 1200 pF	Peralatan Praktikum	Tidak Ada	—	Laboratorium Teknik Tegangan Tinggi (TTT)	1	Detail Edit Hapus
9	PLR0002	mga	Furnitur	Tidak Ada	—	STORAGE DTE	1	Detail Edit Hapus
10	PLR0001	kursi	Furnitur	Tidak Ada	—	STORAGE DTE	1	Detail Edit Hapus

Figure 9. Inventory List of Goods

#	KODE BARANG	NAMA BARANG	KATEGORI	JENIS	LOKASI
1	PP0812	function generator	Peralatan Praktikum	Tidak Ada	STORAGE DTE

#	TANGGAL PERUBAHAN	REVISI	KETERANGAN	ACTION
1	21 October 2024 08:38	Bali	xx121 Inertia x	

#	TANGGAL PEMBELAN	PEMBELAN	TGL DIBEMBAKURAN	ACTION
1	27 Mei 2025	2228952025 - MUHAMMAD FARHAN PRAGASTA	27 Mei 2025	Detail

Figure 10. Item Condition Details

Figure 11. Add Item Form

3.2 Mutation and Consumable Usage Tracking

To support laboratory asset monitoring, the system includes a mutation feature that records item relocation activities. The mutation management interface (Figure 12) and its corresponding input form (Figure 13) allow laboratorians to document location changes systematically, ensuring accountability of equipment movement within laboratory facilities. In addition, consumable materials are managed through the usage module, where stock transactions are recorded as shown in Figure 14 and Figure 15. These features were derived from requirements emphasizing operational transparency and historical usage traceability.

Figure 12. Mutation Management Menu

Home > Kelola Mutasi > Tambah Mutasi

Tambah Mutasi

Nama Mutasi

Tanggal Mutasi

Keterangan

Lokasi Sebelumnya

Lokasi Baru

Barang #1

-- Pilih Lokasi Dahulu --

1

+ Barang

Submit

Reset

Batal

Figure 13. Add mutation form

Home > Kelola Pemakaian

Kelola Pemakaian

Lokasi...

Tanggal Pemakaian

Apply

Reset Filter

Tambah Transaksi Stok

Excel

Showing 1 - 1 From Total 1

Page 1 of 1

Per Page 10

#	JENIS	KETERANGAN	LOKASI	TANGGAL	JUMLAH JENIS BARANG	JUMLAH BARANG	ACTION
1	Pertama	==1st insert==	STORAGE DTE	—	1	6	Detail
2	Pertama	==1st insert==	STORAGE DTE	—	1	20	Detail
3	Pertama	==1st insert==	STORAGE DTE	—	1	2	Detail

Figure 14. Usage Menu

Figure 15. Add Usage Form

3.3 Borrowing and Return Features

A key contribution of the Silab Elektro system is the integration of borrowing and return workflows with the student academic portal. Students can submit borrowing requests through their authenticated accounts (Figure 16), while laboratorians process applications and monitor loan status through the administrative interface (Figure 17 and Figure 18). Recorded transaction data include borrower identity, borrowed items, loan dates, estimated return schedules, and return condition updates. This workflow ensures that item availability and loan history are consistently tracked, supporting both academic practicum and research activities.

#	TANGGAL	PEMINJAM	TUJUAN PEMINJAMAN	STATUS PENGANTARAN PEMINJAMAN	JUMLAH DIPINJAM	JUMLAH DIKEMBALIKAN	STATUS PENGEMBALIAN	ACTION
1	25 Februari 2025	2110952010 - IMRA NUR IZRIYAH	Praktikum	Disetujui	3	3	Belum Dikembalikan	Detail
2	30 Januari 2025	2110952010 - IMRA NUR IZRIYAH	Praktikum	Disetujui	4	4	Selesai	Detail
3	30 Januari 2025	2110952010 - IMRA NUR IZRIYAH	Praktikum	Disetujui	3	3	Belum Dikembalikan	Detail
4	5 September 2024	2110952010 - IMRA NUR IZRIYAH	Praktikum	Disetujui	3	3	Selesai	Detail

Figure 16. Student Loan Application Menu

No	Tanggal	Peminjam	Tujuan Peminjaman	Status Peminjaman	Jumlah Diterima	Jumlah Dikembalikan	Status Pengembalian	Action
1	8 Januari 2025	210903027 - NIKEL ANINDHA PUTRA	Peminjaman	Belum Dikembalikan	1	1	Belum Dikembalikan	[Detail] [Edit] [Hapus]
2	29 Desember 2023	19040200302004002 - Prima Giverni	Peminjaman	Belum Dikembalikan	1	0	Belum Dikembalikan	[Detail] [Edit] [Hapus]
3	22 Desember 2023	210903009 - ALGAR MOHAMMAD SAMUEL	Peminjaman	Belum Dikembalikan	1	0	Belum Dikembalikan	[Detail] [Edit] [Hapus]
4	19 Desember 2023	210903004 - ARIZ AL KAHFI	Peminjaman	Belum Dikembalikan	1	1	Belum Dikembalikan	[Detail] [Edit] [Hapus]
5	18 Desember 2023	210903003 - YUSUF RAHMAN DAMANAH	Peminjaman	Belum Dikembalikan	1	1	Belum Dikembalikan	[Detail] [Edit] [Hapus]
6	18 Desember 2023	210903003 - YUSUF RAHMAN DAMANAH	Peminjaman	Belum Dikembalikan	1	0	Belum Dikembalikan	[Detail] [Edit] [Hapus]
7	11 Desember 2023	19040200302004002 - Prima Giverni	Peminjaman	Belum Dikembalikan	1	0	Belum Dikembalikan	[Detail] [Edit] [Hapus]
8	9 Desember 2023	210903003 - YUSUF RAHMAN DAMANAH	Peminjaman	Belum Dikembalikan	1	1	Belum Dikembalikan	[Detail] [Edit] [Hapus]
9	8 Desember 2023	210903007 - ARIZAL	Peminjaman	Belum Dikembalikan	1	1	Belum Dikembalikan	[Detail] [Edit] [Hapus]
10	3 Desember 2023	210903004 - PRIMA ALQADIR	Peminjaman	Belum Dikembalikan	1	0	Belum Dikembalikan	[Detail] [Edit] [Hapus]

Figure 17. List of Loans Applied for by Students

No	Kode Barang	Nama Barang	Kategori	Status	Tanggal Dikembalikan	Kategori Dikembalikan	Aksi	Kategori Dikembalikan	Tanggal Dikembalikan
1	000001	multimeter FLUKE 68A	Dat	Hangus			[Detail] [Edit] [Hapus]		
2	000001	multimeter FLUKE 68A	Dat	Hangus			[Detail] [Edit] [Hapus]		

Figure 18. Loan Details

4. Conclusion

This study developed the Silab Elektro laboratory inventory information system as a web-based solution to support operational management in electrical engineering laboratories. By applying the Research and Development approach, structured through the Waterfall development stages and implemented using the Model–View–Controller architecture, the system provides an organized framework for managing inventory data, borrowing and return workflows, item mutations, consumable usage, and compensation handling.

The developed system improves laboratory inventory governance by enhancing traceability, accountability, and transaction transparency compared to manual documentation practices. Functional validation through requirement-based black-box testing indicates that the implemented modules operate according to the predefined system requirements. Future work may focus on integrating automated reporting features, extending interoperability with institutional academic platforms, and enhancing analytical capabilities for laboratory resource planning.

Acknowledgment

The authors would like to express their sincere gratitude to the *Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM) Universitas Andalas* for the financial support provided through the *Penelitian Karya Inovasi Tenaga Kependidikan* Number: 88/UN16.19/PT.01.03/PKITK/2024 research scheme, which made this research possible.

References

- [1] M. Barratt, T. J. Kull, and A. C. Sodero, "Inventory record inaccuracy dynamics and the role of employees within multi-channel distribution center inventory systems," *J. Oper. Manag.*, vol. 63, no. September, pp. 6–24, 2018, doi: 10.1016/j.jom.2018.09.003.
- [2] Â. Silva, M. Silva, and A. C. Ferreira, "Inventory Management and Its Influence on the Supply of High-Value Products: Case Study Evidence," *Logistics*, vol. 9, no. 4, p. 170, 2025, doi: 10.3390/logistics9040170.
- [3] L. Velasquez, J. Grados, S. Rubiños, J. Grados, and C. Marujo, "SOFTWARE DESIGN FOR INVENTORY MANAGEMENT," *J. Appl. Eng. Technol. Sci.*, vol. 6, no. 1, pp. 138–154, 2024.
- [4] S. S. Shin, "Teaching Method for Entity-Relationship Models Based on Semantic Network Theory," *IEEE Access*, vol. 10, pp. 94908–94923, 2022, doi: 10.1109/ACCESS.2022.3206028.
- [5] M. Khan, A. N. Alshahrani, and J. Jacquemod, "Digital Platforms and Supply Chain Traceability for Robust Information and Effective Inventory Management: The Mediating Role of Transparency," *Logistics*, vol. 7, no. 2, 2023, doi: 10.3390/logistics7020025.
- [6] Y. S. Rahayu, Y. Saputra, and D. Irawan, "Implementasi Metode Waterfall Pada Pengembangan Sistem Informasi Mobile E-Disarpus," *Zo. J. Sist. Inf.*, vol. 6, no. 2, pp. 523–534, 2024, doi: 10.31849/zn.v6i2.20538.
- [7] A. Sunardi and Suharjito, "MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based," *Procedia Comput. Sci.*, vol. 157, pp. 134–141, 2019, doi: 10.1016/j.procs.2019.08.150.
- [8] A. R. Hevner, S. T. March, J. Park, and S. Ram, "'Design science in information systems research.' MIS quarterly (2004): 75-105.," *Source MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.
- [9] K. Mahmood, G. Rasool, F. Sabir, and A. Athar, "An Empirical Study of Web Services Topics in Web Developer Discussions on Stack Overflow," *IEEE Access*, vol. 11, no. February, pp. 9627–9655, 2023, doi: 10.1109/ACCESS.2023.3238813.
- [10] G. Paolone, R. Paesani, M. Marinelli, and P. Di Felice, "Empirical assessment of the quality of MVCWeb applications returned by xgenerator," *Computers*, vol. 10, no. 2, pp. 1–18, 2021, doi: 10.3390/computers10020020.
- [11] R. Gu, J. M. Rojas, and D. Shin, *Software testing for extended reality applications: a systematic mapping study*, vol. 32, no. 2. 2025. doi: 10.1007/s10515-025-00523-7.
- [12] A. A. Adekunle, B. L. Abolore, G. Mutiu, and A. M. Olalekan, "Design and Implementation of a Web-Based Laboratory Management System for Efficient Resource Tracking," *Asian J. Electr. Sci.*, vol. 13, no. 2, pp. 19–24, 2024, doi: 10.70112/ajes-2024.13.2.4248.
- [13] A. Saravanos and M. X. Curinga, "Simulating the Software Development Lifecycle: The Waterfall Model," *Appl. Syst. Innov.*, vol. 6, no. 6, 2023, doi: 10.3390/asi6060108.
- [14] P. Bambazek, I. Groher, and N. Seyff, "Requirements engineering for sustainable software systems: a systematic mapping study," *Requir. Eng.*, vol. 28, no. 3, pp. 481–505, 2023, doi: 10.1007/s00766-023-00402-1.
- [15] T. Koboyatshwene and Y. Ayalew, "Requirements Traceability: A Systematic Literature Review," *Proc. ACM Symp. Appl. Comput.*, no. February, pp. 1509–1510, 2025, doi: 10.1145/3672608.3707952.
- [16] P. Teixeira, C. Eusébio, and L. Teixeira, *Understanding the integration of accessibility requirements in the development process of information systems: a systematic literature review*, vol. 29, no. 2. Springer London, 2024. doi: 10.1007/s00766-023-00409-8.
- [17] M. M. I. Molla, J. Ahmad, and W. M. N. Wan Kadir, "A Comparison of Transforming the User Stories and Functional Requirements into UML Use Case Diagram," *Int. J. Innov. Comput.*, vol. 14, no. 1, pp. 29–36, 2024, doi: 10.11113/ijic.v14n1.463.
- [18] T. Ahmad, J. Iqbal, A. Ashraf, D. Truscan, and I. Porres, "Model-based testing using UML activity diagrams: A systematic mapping study," *Comput. Sci. Rev.*, vol. 33, pp. 98–112, 2019, doi: 10.1016/j.cosrev.2019.07.001.
- [19] P. Souvik, C. Atrayee, and G. Digbijay, "Study of smart inventory management system based on the internet of things (IOT)," *IJRTBT Int. J. Recent Trends Bus. Tour.*, vol. 3, no. 3, pp. 27–34, 2019,

- [Online]. Available: <https://ejournal.lucp.net/index.php/ijrtbt/article/view/749/662>
- [20] S. Pargaonkar, “A Comprehensive Review of Performance Testing Methodologies and Best Practices: Software Quality Engineering,” *Int. J. Sci. Res.*, vol. 12, no. 8, pp. 2008–2014, 2023, doi: 10.21275/sr23822111402.