

Design and Development of a Web-Based Maintenance Employee Management Application Using Laravel Framework

Moch Aldi Hardiansyah¹; Bambang irawan¹;
Department of Informatics Engineering, Faculty of Computer Science
¹Esa Unggul University
Harapan Indah Boulevard No.4, Harapan Indah City Bekasi
mochamadaldihardiansyah@student.esaunggul.ac.id;
*Corresponding author: Bambang.irawan@esaunggul.ac.id

Received January 2025; accepted February 2025

ABSTRACT

This research focuses on the design and development of a web-based application for managing maintenance employees at a leading automotive components manufacturing company. The current manual system for managing maintenance employees has led to inefficiencies in maintenance task scheduling, slow reporting processes, increased risk of data recording errors, and potential loss of important information. To address these issues, a web-based application using the Laravel framework was developed to integrate all aspects of maintenance employee management. The Laravel framework was chosen for its ease of development, robust security features, and extensive community support. The application enables the company to manage employee data, create maintenance schedules, monitor performance, and generate automated real-time reports. This study implemented a digitalized maintenance employee management system to replace the manual system, developed effective scheduling and performance monitoring features, and created an automated reporting system that provides real-time information. The benefits include improved efficiency in managing maintenance employee data, optimized scheduling and task distribution, minimized risk of errors in data recording and management, and facilitated decision-making based on accurate and real-time reports. Through the development of this application, the operational efficiency of the company is expected to improve, supporting sustainable growth and optimizing human resource utilization in the maintenance department.

Keywords: Web Application, Laravel, Maintenance Employee Management, Information System, Digitalization.

1. Introduction

The digital transformation of industrial processes has become crucial for manufacturing companies to maintain competitiveness in today's rapidly evolving business environment. One of the key areas that requires effective management is the maintenance department, which plays a vital role in ensuring continuous production operations. The management of maintenance employees, their tasks, schedules, and performance evaluation has traditionally been conducted through manual, paper-based systems in many manufacturing facilities.[1]

Manual maintenance employee management systems present several challenges, including inefficient task scheduling, delayed reporting processes, increased risks of data recording errors, and potential loss of critical information. These issues can lead to reduced productivity, increased downtime, and higher operational costs. As manufacturing companies strive for operational excellence, there is a growing need to digitalize maintenance employee management systems to improve efficiency, accuracy, and decision-making capabilities.[2]

Web-based applications offer a promising solution to address these challenges by providing an integrated platform for managing all aspects of maintenance employee operations. The Laravel framework, with its robust features, security, and extensive community support, presents an ideal platform for developing such applications. By leveraging web technologies, companies can create systems that enable real-time monitoring, automated reporting, and efficient resource allocation.[3]

This research focuses on the design and development of a web-based maintenance employee management application using the Laravel framework. The application aims to replace manual systems with a digitalized solution that streamlines employee data management, task scheduling, performance monitoring, and reporting processes. By implementing this application, manufacturing companies can expect to achieve higher operational efficiency, better resource utilization, and improved decision-making capabilities based on accurate, real-time data.[4]

2. Methodology

2.1 System Requirements Analysis

The initial phase involved gathering and analyzing requirements from stakeholders in the maintenance department. This included interviews with maintenance managers, supervisors, and employees to understand their workflows, pain points, and needs [5]. The analysis of system needs is divided into, functional needs, non-functional needs, and development needs:

2.1.1 Functional Needs Analysis

Functional Needs Analysis is the process of identifying and defining the main functions that a system must have to meet user requirements. These needs include the features and services that the system must provide without considering how the system will be technically implemented [6].

The analysis of the functional requirements for the system to be developed is shown in Table 1.

Table 1: Functional Needs

No	Functional Needs	Role		
		Admin	Leader	MTN
1	The system can perform login using email and password.	Yes	Yes	Yes
2	The system can display the dashboard page.	Yes	Yes	Yes
3	The system can add section data.	Yes	No	No
4	The system can update section data.	Yes	No	No
5	The system can add line data.	Yes	No	No
6	The system can update line data.	Yes	No	No

7	The system can add machine data.	Yes	No	No
8	The system can update machine data.	Yes	No	No
9	The system can add user data.	Yes	No	No
10	The system can update user data.	Yes	No	No
11	The system can display a summary of the number of users, machines, lines, and sections.	Yes	No	No
12	The system can display the orders page.	Yes	Yes	Yes
13	The system can display the schedule page.	Yes	Yes	Yes
14	The system can display the reports page.	Yes	Yes	Yes

2.1.2 Non-Functional Needs Analysis

Non-Functional Needs Analysis is the process of identifying and defining the characteristics or qualities that a system must have to function optimally. These needs are not directly related to the system's features or services (what the system does), but rather to how the system operates.[7]

The non-functional requirements for the system to be developed are listed in Table 2.

Table 2: Non- Functional Needs

No	Non- Functional Needs
1	The system must provide a fast response time to user requests.
2	The system must be available at all times with minimal downtime.
3	The system must have a strong authentication system and allow access only to authorized users.
4	The system must be designed to use hardware resources efficiently.
5	The user interface must be responsive across various devices and screen sizes.

2.2 System Design

Based on the requirements analysis, the system architecture was designed. This phase included creating database schemas, defining system modules, and designing user interfaces.[7]. Entity Relationship Diagrams (ERD) and Class Diagrams were developed to model the data structure and relationships. Use Case Diagrams were created to visualize the system's functionality from the user's perspective.

2.2.1 Entity Relationship Diagram (ERD)

ERD is a diagram to visually depict the logical structure of an employee database.[8]. This diagram is used to show relationships between employees or objects.

The following depicts the Entity-Relationship Diagram (ERD) for the developed

system, as illustrated in Figure 1

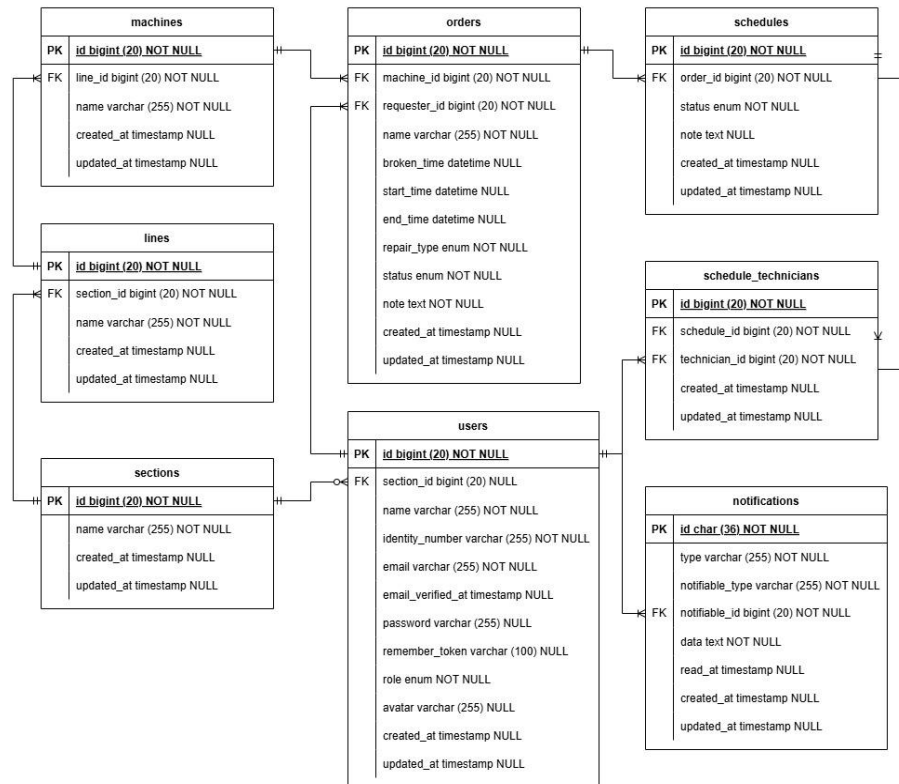


Figure 1. Entity Relationship Diagram (ERD)

2.2.2 Class Diagram

Class Diagram is used to model the structure of an object-oriented system. This diagram represents the classes within the system along with their attributes, methods, and relationships between classes.[9]

The following illustrates the Class Diagram of the system to be developed, as shown in Figure 2.

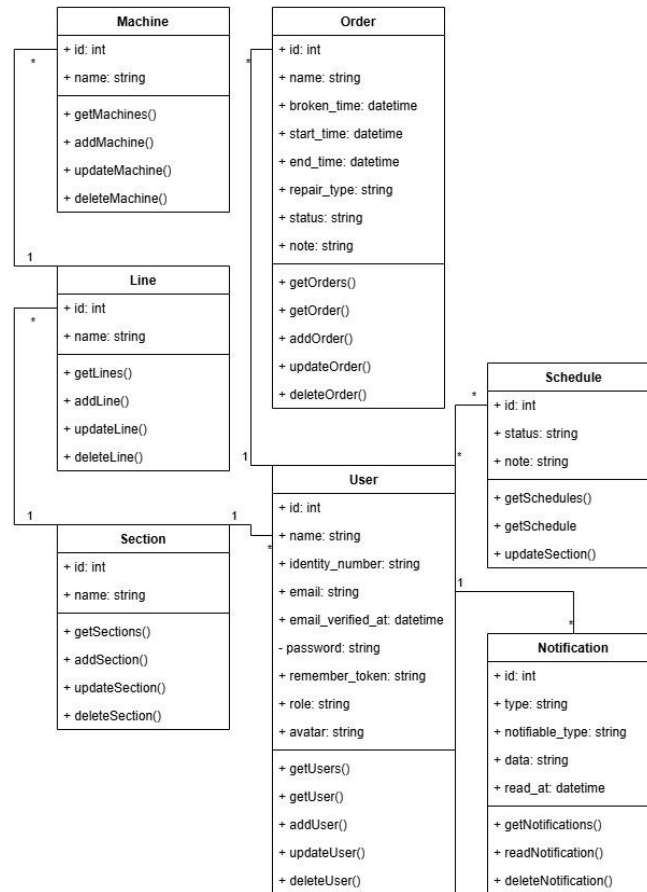


Figure 2. Class Diagram

2.2.3 Use Case Diagram

Use Case Diagram is used to model the interaction between users (actors) and the system. This diagram illustrates how the system is used from the user's perspective, highlighting the main functions (*use cases*) provided by the system.[10]

The following illustrates the Use Case Diagram of the system to be developed, as depicted in Figure 3.

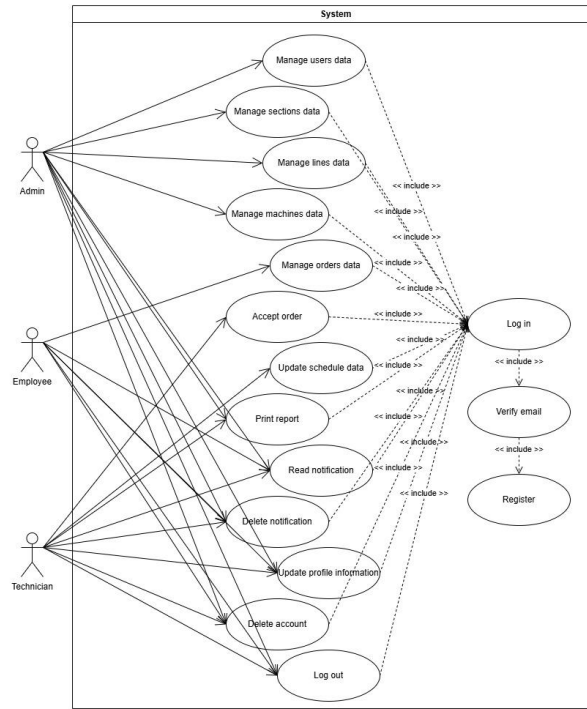


Figure 3. Use Case Diagram

2.2.4 Activity Diagram

The Activity Diagram is a visual representation of the workflow or business process within a system, delineating the execution steps for activities within a software system [11]. Figure 4 represents the Login activity, while Figure 5 illustrates the Logout process. Figures 6 through 9 showcase the addition of a line, the acceptance of an order, the addition of a machine, and the addition of an order, respectively. Figures 10 and 11 depict the deletion of an account and a notification. Figure 12 presents the function of printing a report, and Figure 13 illustrates the process of reading a notification. Figures 14 and 15 represent the updating of a profile and schedule, respectively. Finally, Figures 16 and 17 illustrate the registration of a new user and the verification of an email, respectively.

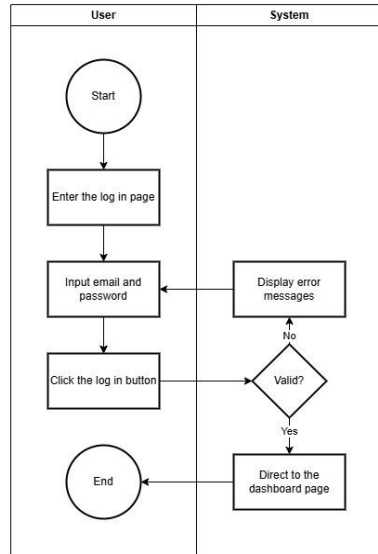


Figure 4. Login

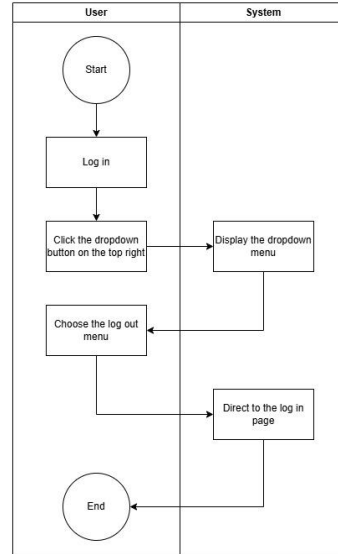


Figure 5. Logout

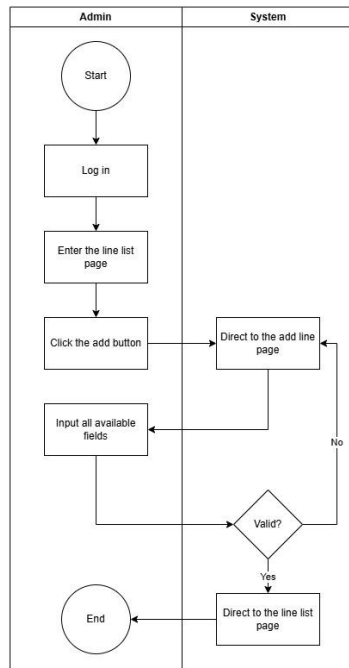


Figure 6. Add-line

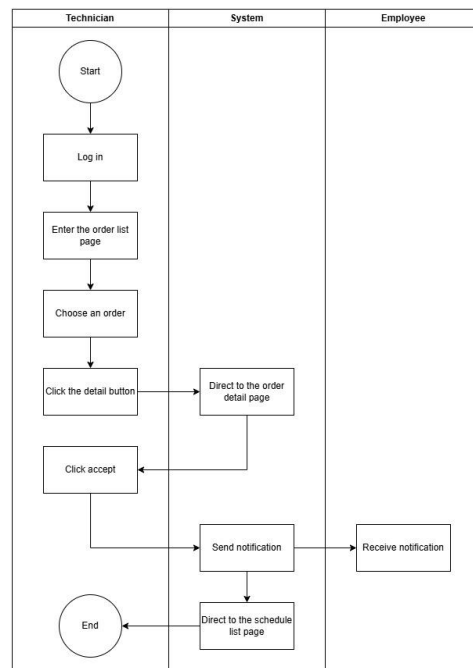


Figure 7. Accept-order

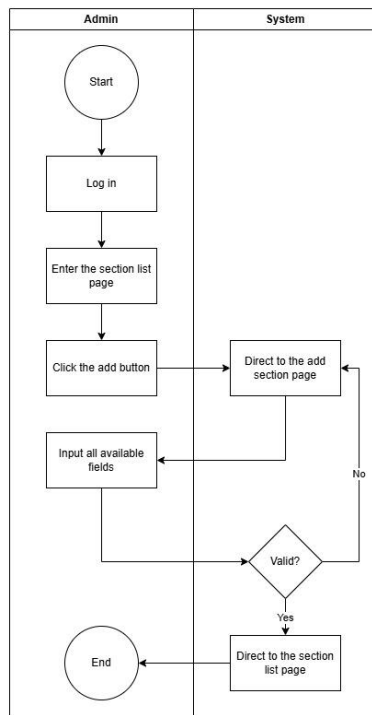


Figure 8. Add-Machine

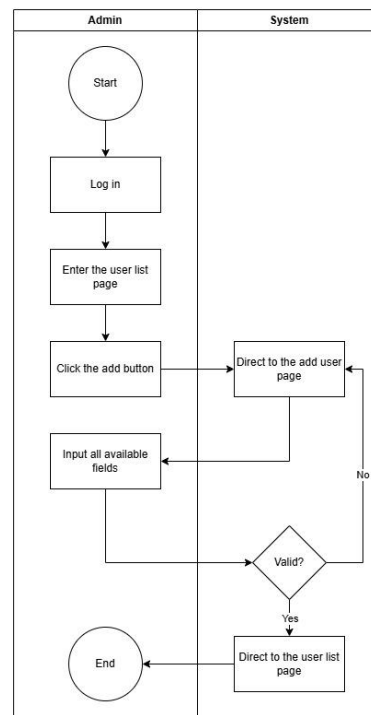


Figure 9. Add-order

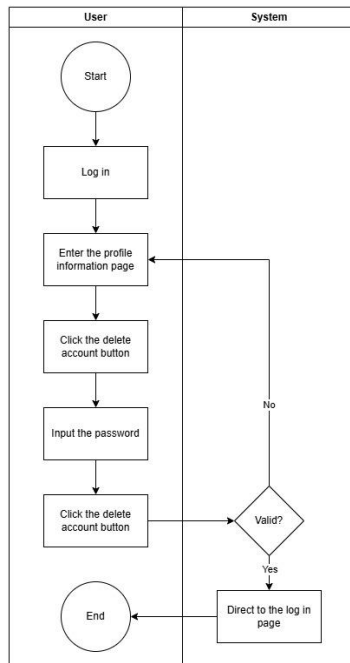


Figure 10. Delete-Account

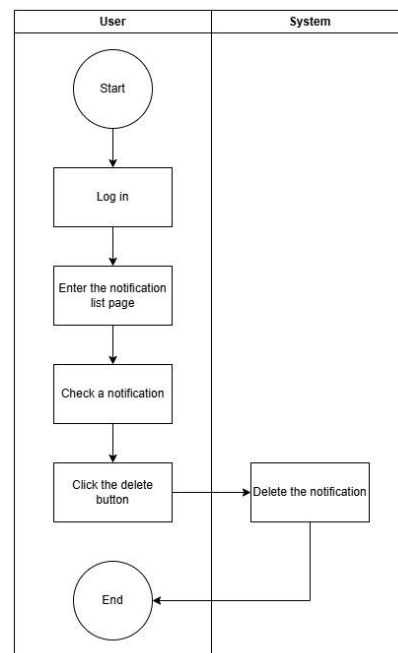


Figure 11. Delete-Notification

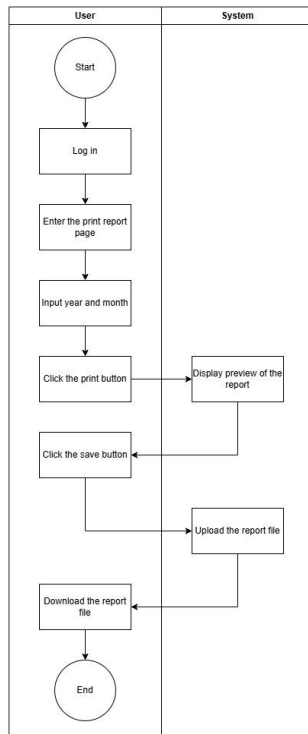


Figure 12. Print-Report

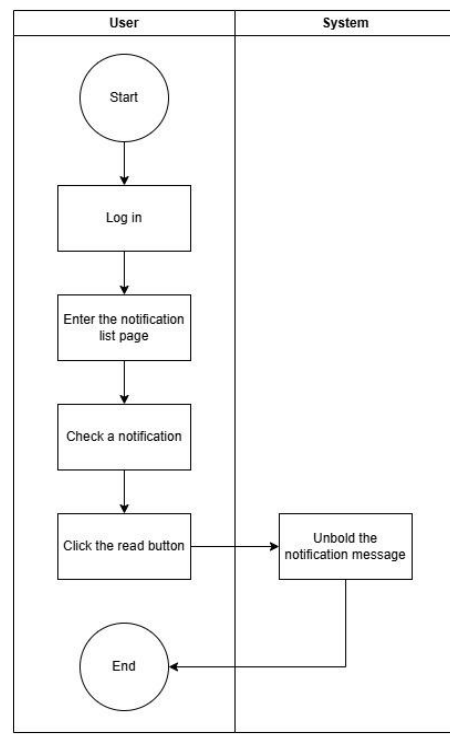


Figure 13. Read-Notification

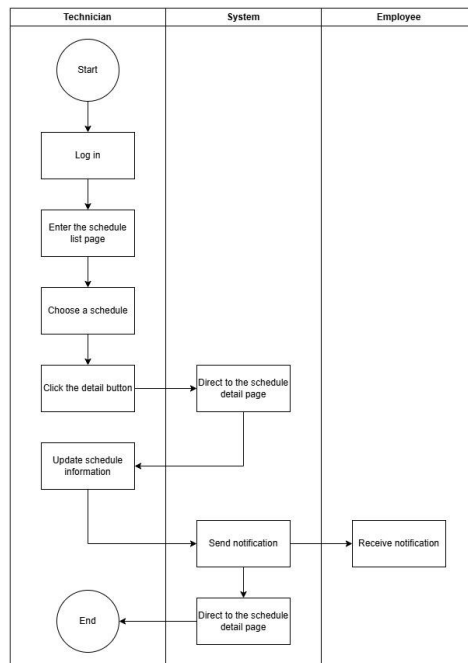


Figure 14. Update-Profile

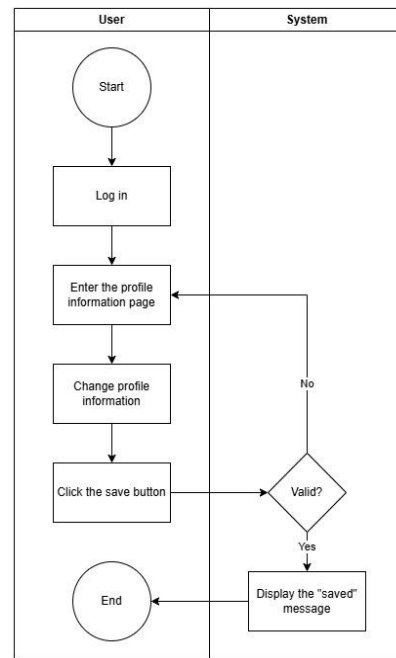


Figure 15. Update-Schedule

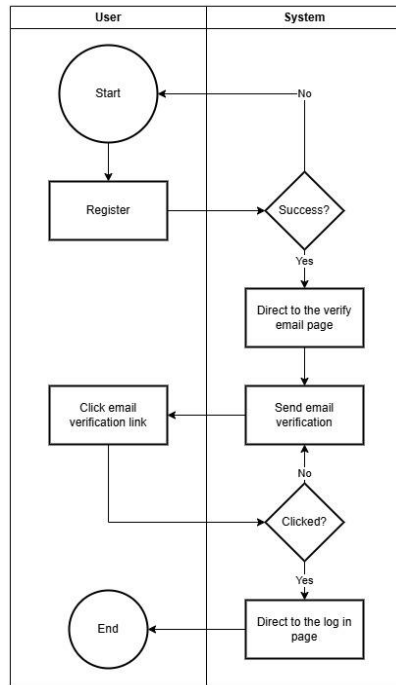


Figure 16. Register

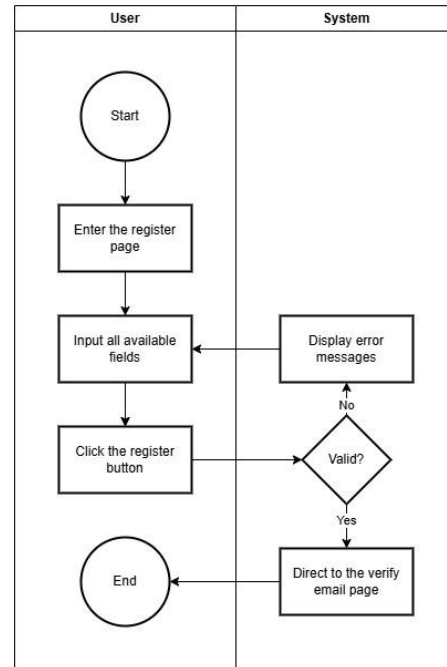


Figure 17. Verify-Email

3. Results and Discussion

3.1 Implementation

The application was deployed in a controlled environment, with data migration from the existing manual system.[12]. Training sessions were conducted for all users to ensure smooth adoption of the new system. A phased implementation approach was used to minimize disruption to ongoing maintenance operations.

The following is the user interface implementation using the Laravel framework:

- **Login Page**

On the login page, there is a form for logging in using a registered email and password. As shown in the figure 18.

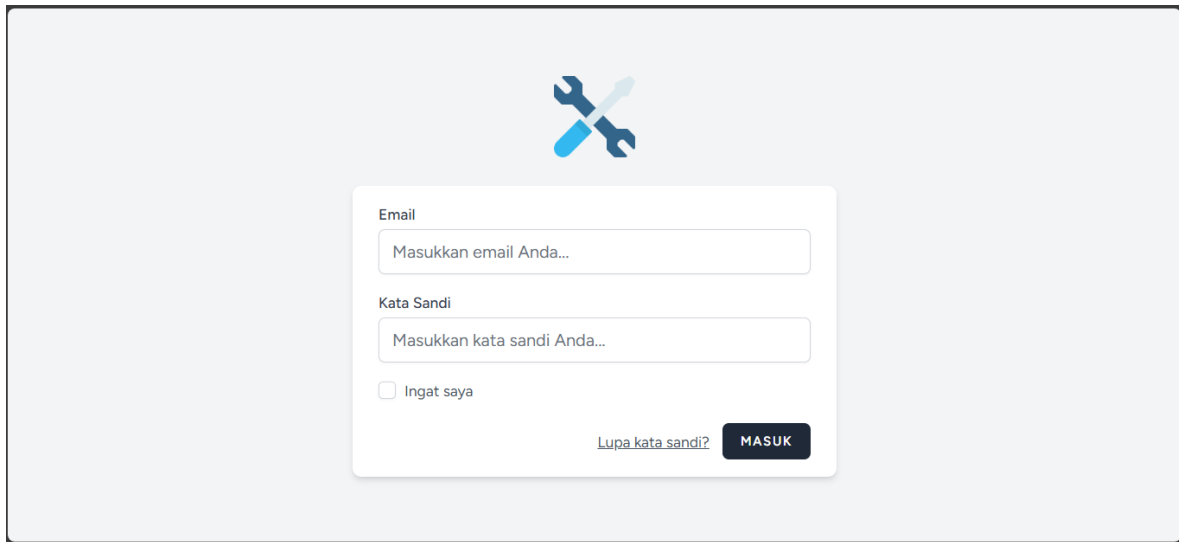


Figure 18. Login Page

○ Dashboard Page

On the dashboard page, there is a list of employee orders for machines that are damaged on the respective line, as well as the maintenance schedule showing which lines are currently being worked on. As shown in the figure 19.

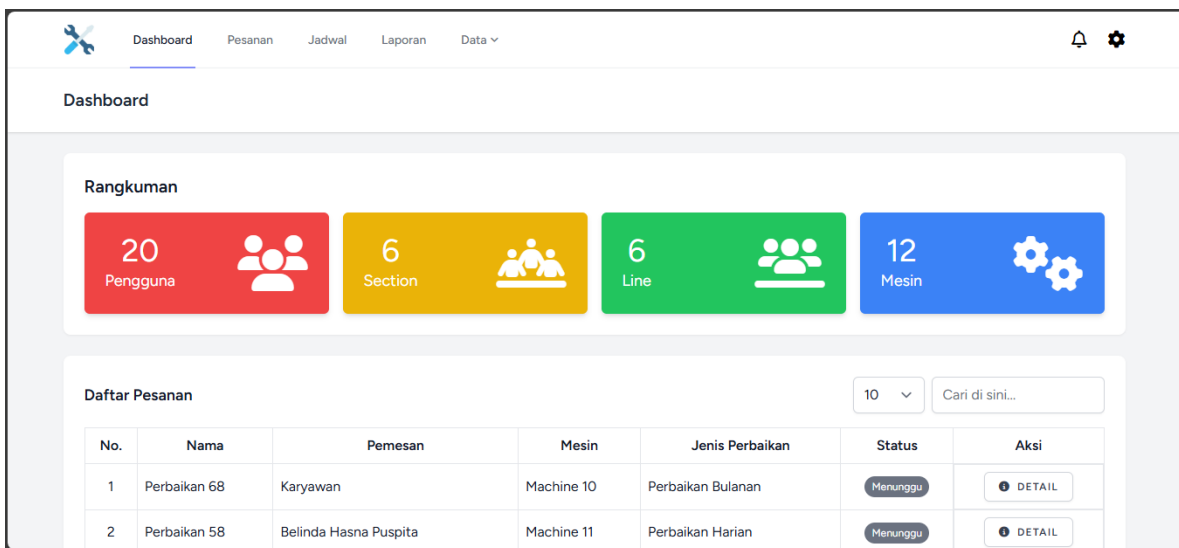
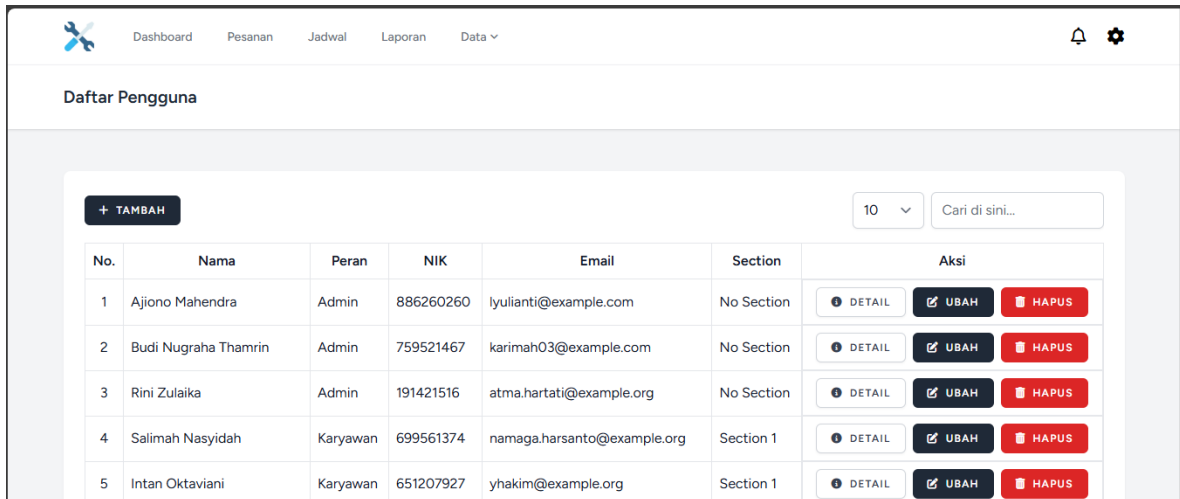


Figure 19. Dashboard Page

○ User List Page

On the user list page, there is a table displaying the list of users, an "Add" button to add new users, a search bar to find users, an "Edit" button to modify user data, and a "Delete" button to remove users. As shown in the figure 20.

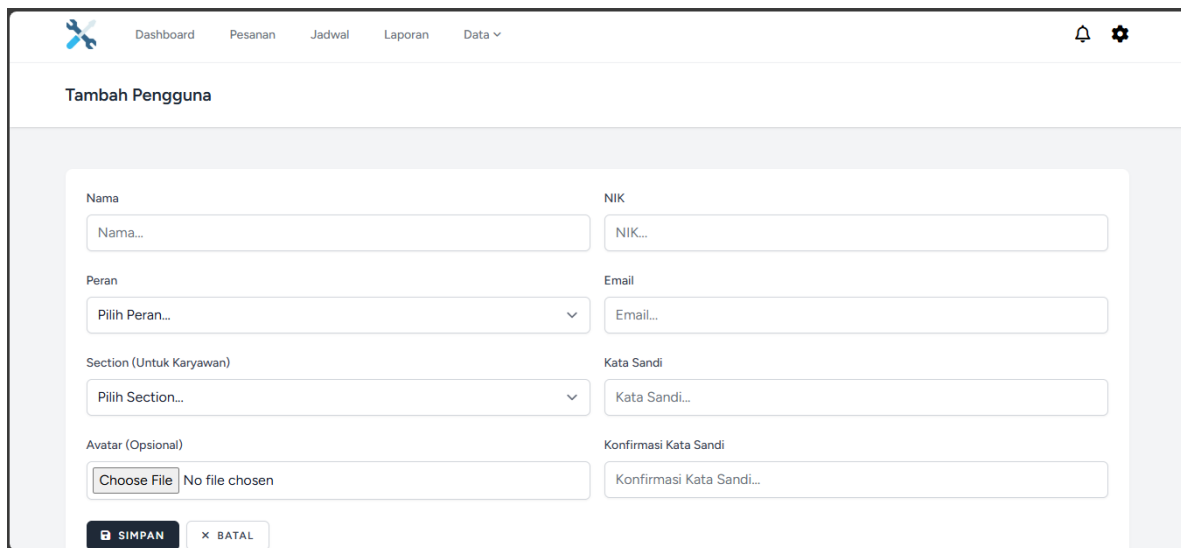


No.	Nama	Peran	NIK	Email	Section	Aksi
1	Ajiono Mahendra	Admin	886260260	lyulianti@example.com	No Section	DETAIL UBAH HAPUS
2	Budi Nugraha Thamrin	Admin	759521467	karimah03@example.com	No Section	DETAIL UBAH HAPUS
3	Rini Zulaika	Admin	191421516	atma.hartati@example.org	No Section	DETAIL UBAH HAPUS
4	Salimah Nasyidah	Karyawan	699561374	namaga.harsanto@example.org	Section 1	DETAIL UBAH HAPUS
5	Intan Oktaviani	Karyawan	651207927	yhakim@example.org	Section 1	DETAIL UBAH HAPUS

Figure 20. User List Page

○ Add User Page

On the add user page, there are several fields that must be filled in, including **name**, **role**, **section**, **image**, **NIK**, **email**, and **password**. Additionally, there are "Save" and "Cancel" buttons to add the user or cancel the action and return to the user list page. As shown in the figure 21.



Tambah Pengguna

Nama:

NIK:

Peran:

Email:

Section (Untuk Karyawan):

Kata Sandi:

Avatar (Opsional): No file chosen

Konfirmasi Kata Sandi:

Figure 21. Add User Page

○ Edit User Page

On the edit user page, there are several fields that must be filled in, including **name**, **role**, **section**, **image**, **NIK**, **email**, and **password**. Additionally, there are "Save" and "Cancel" buttons to update the user or cancel the action and return to the user list page. As shown in the figure 22.

Ubah Pengguna

Nama: Ajiono Mahendra

NIK: 886260260

Peran: Admin

Email: lyulianti@example.com

Section (Untuk Karyawan): No Section

Kata Sandi: Kata Sandi...

Konfirmasi Kata Sandi: Konfirmasi Kata Sandi...

Avatar (Opsional): Choose File No file chosen

PERBARUI BATAL

Figure 22. Edit User Page

○ Orders Page

On the orders page, users can view a list of orders created by employees for damaged machines. Additionally, the order status can be seen, whether it is **waiting**, **accepted**, or **completed**, as confirmed by the technician. As shown in the figure 23.

Daftar Pesanan

+ TAMBAH

10 Cari di sini...

No.	Nama	Pemesan	Mesin	Jenis Perbaikan	Status	Aksi
1	Perbaikan 68	Karyawan	Machine 10	Perbaikan Bulanan	Menunggu	DETAIL UBAH HAPUS
2	Perbaikan 58	Belinda Hasna Puspita	Machine 11	Perbaikan Harian	Menunggu	DETAIL UBAH HAPUS
3	Perbaikan 43	Kemal Ramadan	Machine 3	Perbaikan Mingguan	Diterima	DETAIL UBAH HAPUS
4	Perbaikan 26	Belinda Hasna Puspita	Machine 1	Perbaikan Harian	Diterima	DETAIL UBAH HAPUS
5	Perbaikan 25	Ida Nasyiah	Machine 1	Perbaikan Harian	Diterima	DETAIL UBAH HAPUS
6	Perbaikan 9	Martani Dalimin Setiawan	Machine 6	Perbaikan Harian	Diterima	DETAIL UBAH HAPUS

Figure 23. Orders Page

- **Request Repair Page**

On this page, employees can submit a repair request for a damaged machine. Several fields must be filled in, including **repair name, machine name, breakdown time, type of repair, start time, end time, and notes**. Additionally, there are "Save" and "Cancel" buttons to submit the repair request or cancel the action. As shown in the figure 24.

The screenshot shows the 'Ajukan Perbaikan' (Request Repair) page. The form includes the following fields:

- Nama Perbaikan**: Text input field.
- Waktu Mulai Perbaikan**: Datetime picker field.
- Nama Mesin**: Dropdown menu.
- Waktu Selesai Perbaikan**: Datetime picker field.
- Waktu Rusak**: Datetime picker field.
- Catatan**: Text area for notes.

At the bottom of the form are two buttons: **SIMPAN** (Save) and **BATAL** (Cancel).

Figure 24. Request Repair Page

- **Edit Repair Request Page**

On this page, employees can modify a submitted repair request for a damaged machine. Several fields must be filled in, including **repair name, machine name, breakdown time, type of repair, start time, end time, and notes**. Additionally, there are "Save" and "Cancel" buttons to update the repair request or cancel the action. As shown in the figure 25.

The screenshot shows the 'Ubah Pengajuan Perbaikan' (Edit Repair Request) page. The form includes the following fields:

- Nama Perbaikan**: Text input field with value "Perbaikan 68".
- Waktu Mulai Perbaikan**: Datetime picker field with value "03/11/2024 08:13 AM".
- Nama Mesin**: Dropdown menu with value "Machine 10".
- Waktu Selesai Perbaikan**: Datetime picker field with value "09/17/2025 03:01 AM".
- Waktu Rusak**: Datetime picker field with value "08/05/2024 12:48 PM".
- Catatan**: Text area for notes, containing placeholder text: "Voluptate nisi nobis est sint queraet exercitationem dolorum. Voluptatem qui velit ex. Distinctio non exercitationem fugit tenetur voluptatem ad."

At the bottom of the form are two buttons: **PERBARUI** (Update) and **BATAL** (Cancel).

Figure 25. Edit Repair Request Page

- **Schedule Page**

On this page, users can only view the schedule of technician and employee activities in handling machine damage cases. As shown in the figure 26.

No.	Nama Perbaikan	Jenis Perbaikan	Waktu Mulai	Waktu Selesai	Status	Aksi
1	Perbaikan 36	Perbaikan Bulanan	20-04-2025, 20:48:58	23-10-2025, 05:43:32	Belum Dimulai	DETAIL
2	Perbaikan 73	Perbaikan Bulanan	26-08-2024, 01:40:53	24-07-2024, 04:18:36	Selesai	DETAIL
3	Perbaikan 34	Perbaikan Harian	15-08-2024, 12:14:46	14-01-2025, 17:19:27	Selesai	DETAIL
4	Perbaikan 12	Perbaikan Bulanan	07-03-2025, 01:57:14	29-08-2024, 10:43:37	Sedang Dikerjakan	DETAIL
5	Perbaikan 2	Perbaikan Mingguan	09-09-2024, 12:21:34	18-08-2025, 06:51:32	Selesai	DETAIL
6	Perbaikan 2	Perbaikan Mingguan	09-09-2024, 12:21:34	18-08-2025, 06:51:32	Sedang Dikerjakan	DETAIL

Figure 26. Schedule Page

- **Report Page**

On this page, users can view **monthly and annual report transcripts**, which can then be exported to **PDF** for printing. As shown in the figure 27.

Laporan

Laporan Bulanan

Tahun

Pilih Tahun...

Bulan

Pilih Bulan...

CETAK

Laporan Tahunan

Tahun

Pilih Tahun...

CETAK

Figure 27. Report Page

3.2 System Testing

Black Box Testing is a software testing method that focuses on system functionality without examining the source code. Testers only observe the input and output to ensure that

the system works according to the specifications.[13]

The results of the system testing for the developed system are presented in Tables 3 to 8.

3.2.1 Login Page Testing

The following is the testing conducted on the login page:

Table 3: Login Page Testing

No	Testing	Result	Conclusion
1	Login using a registered account	User diarahkan ke halaman dashboard.	OK
2	Login using an unregistered account	Muncul pesan error pada form login dan user tetap berada di halaman login.	OK

3.2.2 Dashboard Page Testing

The following is the testing conducted on the dashboard page:

Table 4: Dashboard Page Testing

No	Testing	Result	Conclusion
1	Employee creates a maintenance technician order for machine repair.	There is a list of orders including the orderer's name, machine, type of repair, and status, with the status showing as "waiting" (not yet accepted).	OK
2	Employee creates a maintenance technician order for machine repair.	There is a list of orders including the orderer's name, machine, type of repair, and status, with the status showing as "accepted."	OK
3	Technician schedule list showing machines under repair.	The schedule list displays the name, type, time, and status (not started).	OK
4	Technician schedule list showing machines under repair.	The schedule list displays the name, type, time, and status (in progress).	OK
5	Technician schedule list showing machines under repair.	The schedule list displays the name, type, time, and status (completed).	OK

3.2.3 Order Page Testing

The following is the testing conducted on the orders page:

Table 5: Order Page Testing

No	Testing	Result	Conclusion
1	Employee creates a maintenance technician order for machine repair by pressing "Add."	The page will navigate to fill in the repair request data.	OK
2	Submit repair after filling in the details and pressing "Save."	The repair request will be saved and appear on the dashboard or schedule page.	OK
3	Submit repair after filling in the details and pressing "Cancel."	The repair request will be deleted and the page will return to the orders page.	OK

3.2.4 Schedule Page Testing

The following is the testing conducted on the schedule page:

Table 6: Schedule Page Testing

No	Testing	Result	Conclusion
1	The schedule page will display the repair schedule if the technician order is filled in by the employee.	The technician schedule will be available after the employee fills in the order.	OK

3.2.5 Report Page Testing

The following is the testing conducted on the Report page:

Table 7: Report Page Testing

No	Testing	Result	Conclusion
1	The report will be available once the employee's order is completed by the technician.	The report is saved and can be printed	OK

3.2.6 Data Addition Page Testing

The following is the testing conducted on the data addition page:

Table 8: Data addition Page Testing

No	Testing	Result	Conclusion
1	Section data is added by the admin.	New section data has been successfully added.	OK
2	Line data is added by the admin.	New line data has been successfully added.	OK
3	Machine data is added by the admin.	New machine data has been successfully added.	OK
4	User data is added by the admin.	New user data has been successfully added.	OK

4. Conclusions

The application streamlines the task scheduling process for maintenance employees, thereby enhancing the organization and transparency of work allocation [15]. As a web-based solution, it provides flexible access from various devices, thereby improving ease of use and operational efficiency within the maintenance division.

The Laravel framework was selected due to its robust security, ease of development, and extensive features, which support the creation of a scalable and reliable web-based system [14]. Overall, the developed application is anticipated to improve work efficiency, minimize the risk of errors in manual records, and aid the company in optimizing human resource management, particularly within the maintenance division.

REFERENCES

- [1] M. Savastano, C. Amendola, F. Bellini, and F. D’Ascenzo, “Contextual impacts on industrial processes brought by the digital transformation of manufacturing: A systematic review,” *Sustainability (Switzerland)*, vol. 11, no. 3, Feb. 2019.
- [2] M. G. S. Aboelmaged, “E-maintenance research: a multifaceted perspective,” *Journal of Manufacturing Technology Management*, vol. 26, no. 5, pp. 606–631, Jun. 2015.
- [3] K. Benmoussa, M. Laaziri, S. Khouliji, M. L. Kerkeb, and A. El Yamami, “A new model for the selection of web development frameworks: Application to PHP frameworks,” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 1, pp. 695–703, Feb. 2019,.
- [4] A. Hassanat Adepoju, B. Austin-gabriel, A. Eweje, and A. Collins, “Framework for Automating Multi-Team Workflows to Maximize Operational Efficiency and Minimize Redundant Data Handling,” 2022.
- [5] F. Heriyanti and A. Ishak, “Design of logistics information system in the finished product warehouse with the waterfall method: Review literature,” in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jun. 2020.
- [6] A. Hassanat Adepoju, B. Austin-gabriel, A. Eweje, and A. Collins, “Framework for Automating Multi-Team Workflows to Maximize Operational Efficiency and Minimize Redundant Data Handling,” 2022.
- [7] L. Tóth and L. Vidács, “Study of Various Classifiers for Identification and Classification of Non-Functional Requirements.”2019.
- [8] E. Ansari, “WEB BASED EMPLOYEE MANAGEMENT SYSTEM,” 2019.
- [9] M. Mukherjee and ; M Mukherjee, “Object-Oriented Analysis and Design,” 2016.

- [10] M. N. Arifin and D. Siahaan, "Structural and Semantic Similarity Measurement of UML Use Case Diagram," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 11, no. 2, p. 88, Jul. 2020.
- [11] H. Chen, J. M. Jiang, Z. Hong, and L. Lin, "Decomposition of UML activity diagrams," *Softw Pract Exp*, vol. 48, no. 1, pp. 105–122, Jan. 2018.
- [12] J. Rao *et al.*, "Data migration in the cloud database: A review of vendor solutions and challenges," ~ 96 ~ *International Journal of Computing and Artificial Intelligence*, vol. 3, no. 2, 2022.
- [13] Y. Irawan, S. Muzid, N. Susanti, and R. Setiawan, "System Testing using Black Box Testing Equivalence Partitioning (Case Study at Garbage Bank Management Information System on Karya Sentosa)," European Alliance for Innovation n.o., Feb. 2019.
- [14] M. Amini *et al.*, "MAHAMGOSTAR.COM AS A CASE STUDY FOR ADOPTION OF LARAVEL FRAMEWORK AS THE BEST PROGRAMMING TOOLS FOR PHP BASED WEB DEVELOPMENT FOR SMALL AND MEDIUM ENTERPRISES," 2020.
- [15] S. Bouzidi-Hassini, F. Benbouzid-, S. Tayeb, F. Marmier, M. Rabahi, and F. Marmier, "Considering human resource constraints for real joint production and maintenance schedules," 2015.