

Development of a Web-Based eChecklist System for Final Product Inspection in a Coachbuilding Company

Gehan Gibran Fatahillah^{1*}, Muhammad Rizq Dzaki Asyam², Egi Al Fansyah², Dwi Fahira Alsyah³, Abdul Hakim Satria Nusantara², Rayhan Zahwan Saleh²

¹ Information Technology Study Program, Faculty of Computing, President University, address, Jababeka Education Park, Jl. Ki Hajar Dewantara, Kota Jababeka, Cikarang Utara, Bekasi, 17550, Jawa Barat, Indonesia.

² Master of Information System Study Program, School of Industrial Engineering, Telkom University, Main Campus (Bandung Campus), Jl. Telekomunikasi no. 1, Bandung 40257, West Java, Indonesia.

³ Information System Study Program, School of Industrial Engineering, Telkom University, Main Campus (Bandung Campus), Jl. Telekomunikasi no. 1, Bandung 40257, West Java, Indonesia.

DOI : 10.62123/enigma.v3i2.146

ABSTRACT

Received : March 16, 2026

Revised : April 17, 2026

Accepted : April 22, 2026

Keywords:

Coachbuilding, eChecklist, Final Inspection, Quality Control, Web-based System

This final project develops a web-based eChecklist system to support the final product inspection process in a coachbuilding company. The system is designed to replace manual checklist recording, which is often inefficient, prone to human error, difficult to track, and lacks real-time accessibility. The objective of this project is to improve the consistency and accuracy of inspection records, streamline communication among inspectors, supervisors, and administrators, and provide reliable documentation for each inspected unit. The system was developed using PHP with the Laravel framework and implemented using the Rapid Application Development (RAD) method. The main features include user authentication, role-based access control, digital checklist input, checklist validation, inspection history tracking, and report export. Testing was conducted on core functionalities such as login, role-based access, checklist submission, validation, status tracking, user and role management, and report generation. The results show that all tested functionalities worked properly and achieved the expected outcomes. It can be concluded that the web-based eChecklist system is able to support final inspection activities more effectively, improve data traceability, and enhance operational reliability in the coachbuilding quality control process.

1. INTRODUCTION

The manufacturing sector plays an important and vital role in driving national economic growth [1], particularly in industries such as automotive and vehicle body manufacturing, commonly referred to as a coachbuilding company. The coachbuilding company is responsible for preparing, planning, designing, and producing vehicle bodies, with the long and critical processes from assembly to production process and the final inspection of vehicle units. One of the most vital steps is the final checklist inspection, which ensures that each product leaving the factory meets quality and safety standards before the product reaches the customer's hands [2]. Standardized procedures are also essential in organizational processes because "SOP as quality standards to perform the process of the instruction and executing tasks and activities are important for an organization" [3].

There are still many coachbuilding companies that perform final checklist inspection manually [4]. Meanwhile, manual documentation of these checklists often leads to inefficiencies. These include missing records, inconsistent data entry, or delays in verification [5]. As production output increases, the risks of managing physical and manual records also increase, leading to potential quality control failures or product recalls [6]. To address this, many coachbuilding companies are beginning to adopt digital systems to automate and streamline their internal process. However, digital system adoption in organizations does not always proceed without challenges, since "ERP system implementation is not always running smoothly" [7]. A web-based eChecklist application will offer a useful solution to help coachbuilding company manage and monitor final inspection activities before products are released and reach the customer's hand. By transitioning from manual to digital checklists, a coachbuilding company can ensure accountability, improve accuracy, and maintain consistent standards across all units.

Based on this condition, several problems have been identified that the final product inspection process in the coachbuilding companies are often manual, time-consuming, sensitive to loss, and prone to human error. Current manual practices in inspection recording lack data consistency, security vulnerability, and real-time accessibility, which can hamper efficiency resulting in slowing down and decreasing worker performance and operational accountability. Furthermore, the absence of a centralized digital system makes it difficult for various stakeholders, such as inspectors, administrators, and supervisors, to access, update, and track checklist data promptly. This also poses challenges in maintaining reliable and retrievable inspection records for future audits or other needs. Therefore, there is a critical need to explore how digital technology can be leveraged to streamline the inspection process, reduce

*Corresponding Author Email: egialfansyah@student.telkomuniversity.ac.id

errors, ensure secure data management, and support real-time multi-role access. This final project proposes a web-based eChecklist system specifically designed for coachbuilding companies to support the final inspection process before a vehicle unit is released from the factory and reached the customer's hands. This system is expected to enhance the consistency and accuracy of inspection records that fits with the standard, minimize manual or human errors, and streamline the communication process between inspectors, administrators, and supervisors. By implementing this web-based eChecklist system, coachbuilding companies can possibly improve their operational efficiency, effectiveness of workers' performance, ensure better compliance with quality standards, and maintain reliable documentation for every unit inspected. Additionally, this application will assist in real-time data access, making it easier for decision-makers and other stakeholders to monitor the inspection process and evaluate performance over time.

This final project covers the development of a web-based eChecklist system that allows the users, namely inspectors, administrators, and supervisors, to input inspection data, attach photos and videos, write notes, and view inspection history in a structured and user-friendly interface. The system supports multiple user roles with different access levels to maintain data security and workflow efficiency. However, this final project also has several limitations. The web-based eChecklist system only supports exporting data in spreadsheet format and does not yet include features such as mobile application integration or customizable checklist templates. Additionally, the system is designed primarily for internal use within the factory and does not currently support external integration with other enterprise systems. In developing this system, the methodology used is Rapid Application Development (RAD), which is a software process model that focuses on quick, iterative development and shortens the software development cycle through rapid prototyping and continuous feedback [8]. Rapid Application Development (RAD) emerged as an alternative to the traditional linear waterfall model, offering a faster and more flexible approach by building software in modular components and increments [9]. It combines structured methods with Joint Application Development (JAD) and prototyping techniques to speed up system and application development [10].

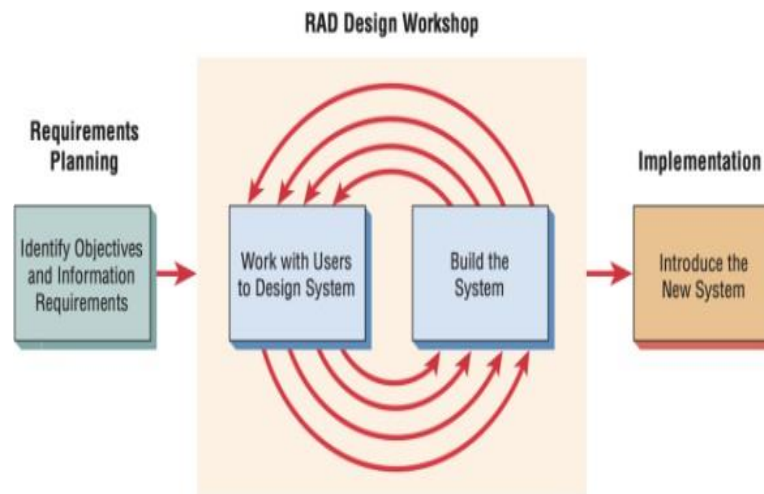


Figure 1. RAD Diagram

The RAD methodology in this final project consists of requirements planning, design workshop, and implementation. At the requirements planning stage, users and analysts come together to align the goals of the application or system and to share the information needed to achieve them. At the design workshop stage, the design process takes place, and any differences between the users' expectations and the analysts' work need to be addressed and refined through active participation and direct feedback. At the implementation stage, once the system design is approved by both the user and the analyst, the programmer begins turning that design into a working application, followed by testing and user evaluation to ensure that the system meets their needs. The main contribution of this study lies in the development and implementation of a lightweight web-based eChecklist system specifically tailored for final product inspection in a coachbuilding company. Unlike many existing digital inspection solutions that are complex, costly, or designed for other industrial contexts, this work emphasizes a practical role-based system with centralized inspection records, validation workflow, historical tracking, and report export that can be implemented more accessibly in medium-scale manufacturing environments.

2. LITERATURE REVIEW

2.1 Quality Control in Manufacturing

Quality control is a vital and important aspect of the manufacturing industry to ensure products meet the required standards before they are delivered to customers [11]. In the context of a coachbuilding company, an organization specializing in vehicle body manufacturing, quality control plays an important role in maintaining product safety, customer satisfaction, and regulatory compliance [12]. Given that these companies often handle customized designs and modifications on commercial vehicles, every unit must pass through a series of stringent inspections. These inspections usually cover welding accuracy, material durability,

installation correctness, paint finishing, electrical systems, and many other critical elements. If any aspect is overlooked, it may lead to product failure, increased warranty claims, or even customer dissatisfaction [13].

Traditionally, quality control procedures in such companies are carried out manually using paper checklists and visual confirmations. This process is often prone to human error, inconsistent reporting, and a lack of traceability. Furthermore, when defects are found after the product leaves the production floor, it becomes difficult to track back to the responsible party or specific production stage. Consequently, this manual method can result in inefficiencies, unnecessary production costs, and reputational damage [14].

In response to these challenges, many modern manufacturing companies are transitioning toward digital solutions, such as electronic checklists (e-Checklists). These systems help streamline the inspection process, enforce consistency, and maintain a historical record of every quality check performed. By implementing a web-based eChecklist system specifically tailored for coachbuilding company final product inspections, the company can improve accountability, data accuracy, and quality assurance [15]. It also allows supervisors and management to monitor inspections in real time and take corrective actions promptly. Therefore, incorporating a digital quality control tool is not just a technical upgrade but a strategic move toward smarter manufacturing practices

2.2 Database Management System

A Database Management System (DBMS) is a vital component in modern software applications, particularly for systems that rely heavily on structured data and require accurate, secure, and efficient data handling. In the context of a web-based eChecklist system for a coachbuilding company, a DBMS is responsible for storing and organizing all inspection-related data, including unit information, checklist entries, inspection results, user logs, timestamps, and corrective action records. The DBMS ensures that this data is not only persistently stored but also easily accessible, modifiable, and analyzable in real-time, facilitating better decision-making across the organization [16]. The use of a DBMS in a manufacturing inspection application helps reduce data redundancy and inconsistency while promoting data integrity. For instance, when a supervisor logs in to review the results of a final inspection, the system must retrieve data from multiple related tables, such as user records, unit details, and checklist logs [17]. This process would be highly inefficient and error-prone without a properly normalized and relational database design. Systems such as MySQL, PostgreSQL, and SQLite are widely used open-source relational DBMS options suitable for this kind of application. These systems support complex querying, indexing, and transactional operations, all of which are essential for maintaining reliable inspection workflows.

Additionally, a DBMS enhances data security by providing role-based access control and permission management. This is particularly important in a quality control setting where different users, inspectors, supervisors, and admins should only have access to data and functions relevant to their roles [18]. A secure DBMS ensures that sensitive data, such as inspection failure logs or supervisor comments, are protected from unauthorized access and tampering. This aligns with prior research emphasizing that “The success and failure of electronic voting implementation depend on the fulfilment of the voter needs on privacy and personal data protection” [19]. Moreover, the use of a DBMS makes it easier to generate historical reports, conduct audits, and comply with internal or external quality standards. By integrating the DBMS with the Laravel framework, developers can leverage Eloquent ORM for seamless interaction with the database, minimizing the risk of SQL errors and simplifying code maintenance. Therefore, the DBMS serves as the backbone of the eChecklist system, enabling smooth, secure, and scalable operation throughout the coachbuilding production quality assurance process [20]

2.3 Model View Controller (MVC)

Model View Controller (MVC) is a widely used architectural pattern that divides an application into three main components: the Model, the View, and the Controller. By separating these concerns, MVC helps keep the code organized and makes development, testing, and maintenance more manageable. MVC is especially effective in web applications like the eChecklist system for coachbuilding companies, where multiple user roles (admin, supervisor, inspector) interact with structured data through various interfaces. Laravel, the framework used for this project, fully supports the MVC pattern and promotes its use as a core structure. The Model’s role is to communicate with the database and oversee the business logic. In the context of this system, models such as Checklist, User, Unit, and Inspection Result represent the core entities. They handle data validation rules, database queries, and relationships between tables. By isolating data handling in the Model layer, the application becomes more reliable and easier to maintain.

The View is the user interface component that renders data for users to see and interact with. Using Laravel Blade templating, views display forms for checklist input, tables of inspection history, and dashboards with statistics. Views in MVC are kept separate from logic, which makes them more manageable and easier to redesign or adapt for different devices and user needs. The Controller serves as the link between the Model and the View. It processes user input, invokes business logic from the model, and determines which view to return as a response. For instance, when a supervisor requests to see inspection history, the Checklist Controller fetches data from the database through models and sends it to the appropriate view for display.

Overall, the MVC structure enhances modularity, scalability, and maintainability. In a quality control system for a coachbuilding company, where frequent updates and multiple stakeholders are common, MVC ensures that changes in the interface, business rules, or data structure can be made independently without disrupting the entire application. This promotes faster development cycles and a cleaner codebase, crucial for supporting production-critical software.

2.4 Role-Based Access Control (RBAC)

Role-Based Access Control (RBAC) is a security approach that limits or grants access to system features based on a user's role within an organization. Instead of assigning permissions directly to each user, RBAC assigns them to roles, and users are then linked to those roles [21]. This approach simplifies the management of user permissions, especially in sophisticated systems with various user types and levels of access requirements. In a manufacturing context, particularly in a coachbuilding company that requires precise quality control before products leave the facility, RBAC ensures that only authorized personnel can perform certain actions such as inputting checklist data, verifying inspection results, or modifying critical information [22].

For instance, inspectors may only have access to fill out checklists, while supervisors can review and validate the entries, and administrators manage user roles and settings. Laravel provides a robust way to implement RBAC through the Spatie Laravel Permission package, which allows developers to define roles and permissions flexibly within the application. By leveraging RBAC, the eChecklist system can enhance data integrity and ensure accountability across user activities. Furthermore, RBAC improves system security by minimizing the risk of unauthorized access and maintaining an audit trail of who performed what actions and when. In addition to technical controls, user awareness and organizational evaluation are also important, since "two important factors need much consideration in raising awareness are how organization influences significantly of end user's attitude and how the organization has the regular assessment or evaluation" [23]. In this final project, RBAC plays a vital role in structuring user interactions and ensuring that the application adheres to organizational hierarchy and data confidentiality, which is particularly important in regulated or quality-sensitive industries like vehicle manufacturing.

2.5 Role-Based Access Control (RBAC)

In many coachbuilding manufacturing companies, particularly those that assemble vehicle bodies, the final inspection process before a product exits the factory is still handled manually using printed checklists or spreadsheets. These manual procedures are prone to human error, difficult to audit, and inefficient when it comes to tracking historical data. Technicians or inspectors often carry out their tasks without a centralized system, making it hard to monitor progress, validate entries in real-time, or ensure consistent adherence to quality standards. Additionally, when defects are found or actions need follow-up, they are sometimes recorded on paper and manually relayed, increasing the risk of miscommunication and delays. Supervisors and management typically must consolidate this information manually to generate reports, which adds to the administrative burden [24][25].

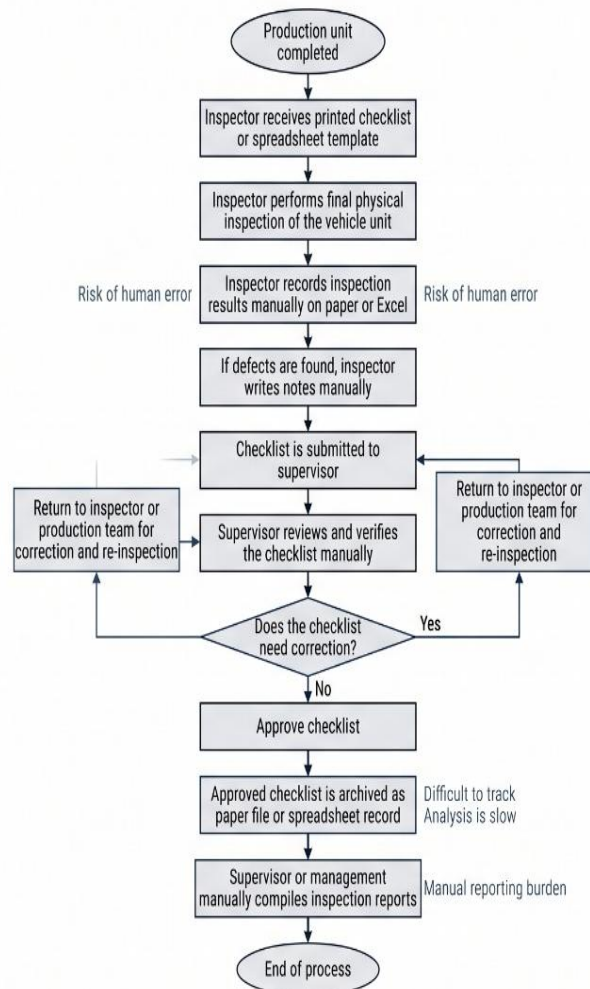


Figure 2. Current General Process of eChecklist

2.6 Current GAP and Problem

Despite the growing need for digital transformation in the manufacturing sector, many coachbuilding companies still rely on outdated, manual methods for performing and recording final product inspections. This outdated process leads to several critical issues. First, manual checklists lack standardization, meaning that different inspectors might interpret checklist items differently, resulting in inconsistencies in product quality. Additionally, because the checklist is often on paper or scattered across multiple Excel files, it becomes difficult to track accountability, validate the timing of inspections, or ensure that every required step was completed. This creates blind spots in the quality assurance process, potentially allowing faulty units to be delivered to customers.

Another problem is the lack of real-time data access. Supervisors and quality control managers cannot immediately monitor inspection progress or quickly identify which units have passed or failed specific checks. This results in delayed responses to quality issues and often requires double work to reconcile inspection records with production reports. In the long run, these inefficiencies can reduce customer satisfaction and increase operational costs due to rework or product recalls [26][27][28].

Furthermore, the absence of centralized data storage makes it difficult to retrieve historical records for audits or performance analysis. Without digital records, it becomes almost impossible to track trends, identify recurring defects, or implement effective continuous improvement initiatives. There is also a lack of role-based access control, meaning sensitive inspection data might be accessible to unauthorized users, posing a security risk.

This final project aims to bridge these gaps by creating a centralized, role-based eChecklist system that automates inspection workflows, reduces human error, enhances data traceability, and improves overall quality control processes in coachbuilding manufacturing companies.

2.7 Related Work

In the field of quality control and inspection systems, various studies and implementations have been conducted to support manufacturing processes through digital solutions. One of the most well-known approaches is the implementation of digital checklists in sectors such as automotive manufacturing, aircraft maintenance, and electronics assembly. These systems are designed to help organizations ensure that each production step is completed according to standard operating procedures. Research has shown that the use of digital checklists significantly improves consistency and reduces human error compared to traditional paper-based systems.

For example, in the automotive industry, Toyota and BMW have integrated tablet-based checklists into their assembly lines to help technicians perform inspections more efficiently. These systems often include real-time monitoring dashboards, barcode scanning for unit verification, and automated logging of completed tasks. Similarly, in aviation, companies like Boeing and Airbus utilize electronic checklist systems that are tightly integrated with their maintenance management platforms. These systems improve traceability and ensure regulatory compliance by automatically storing inspection records for auditing purposes.

In academic research, various models have been proposed to enhance quality control through software-based tools. Many of these studies emphasize the importance of role-based access, data centralization, and mobile accessibility. Some systems also leverage technologies such as RFID, IoT, or cloud computing to provide even greater automation and insight into the production environment.

However, most of these solutions are either highly complex, costly to implement, or tailored to specific industries. There is a gap in simpler, more accessible tools designed for medium-sized manufacturing sectors like coachbuilding companies. This final project aims to build on the lessons from existing research and systems by offering a lightweight, Laravel-based web application that fulfills core inspection and quality control needs with user roles, historical tracking, and data export features tailored for coachbuilding production environments.

2.8 Comparison Overview

Table 1. Comparison between eChecklist Web App for coachbuilding Final Inspection with related work

Feature	Excel-based Checklist	eChecklist Web Application
Dynamic Form Display & Auto Calculations	✓	✓
Real-Time Data Update and Display	✗	✓
Centralized Database Storage	✗	✓
Data Visualization (e.g., Inspection Summary)	✗	✓
History Tracking (Created/Updated Logs)	✗	✓
Login and Register Account	✗	✓
Send Data to Email	✗	✓
Role-Based Access (Admin, Supervisor, Inspector)	✗	✓
Export Data	✗	✓
Change Password Account	✗	✓
Forgot Password Account	✗	✓

Compared with prior digital inspection systems that often emphasize enterprise-scale integration, mobile devices, or advanced automation, the proposed system in this study focuses on a more accessible implementation for a coachbuilding manufacturing context. Its distinction lies in combining role-based access control, centralized inspection storage, validation workflow, historical

tracking, and report export within a lightweight web-based architecture using Laravel and RAD. This makes the proposed approach more suitable for organizations that need practical inspection digitalization without the overhead of more complex industrial platforms

3. RESEARCH METHODS

3.1 System Overview

The web-based eChecklist System at the coachbuilding Company is a web-based software application developed using the PHP programming language and built with the Laravel framework. This system is intended to facilitate the digitalization of the final inspection process before a vehicle body product leaves the company, improving inspection accuracy, traceability, and efficiency in quality control operations. The system supports user authentication and role-based access control to ensure that only authorized personnel, such as inspectors, supervisors, and administrators, can access specific features and perform related actions in the system. This layer of security ensures proper segregation of duties and accountability. Inspectors can submit product inspection checklists digitally through the system.

These checklists include various criteria based on company standards, such as completeness of parts, installation accuracy, cleanliness, and paint quality. Supporting evidence, such as photos, can also be attached. The system also provides features for inputting new data, updating previously created data, and viewing a list of recorded data in order to improve data accuracy and data management. In addition, the system routes submitted checklists to supervisors for review. Supervisors are responsible for verifying and approving or rejecting the submissions, while notifications are sent automatically to relevant users to ensure timely follow-up. All checklist submissions are stored in the system's database and can be accessed to review historical inspections, allowing users to monitor trends, maintain audit trails, and compare results from previous inspections. The system also includes a dashboard to visualize key metrics, such as the number of completed inspections, pending approvals, and rejected items. Admin users have access to generate reports that can be used for internal evaluations and continuous improvement.

3.2 Functional Analysis

This system showed the functions listed in Table 3.1. The main functionalities include register, login, new checklist input and submission, list data, update data, data history, and data export. The register function allows the user to create an account in the system, while the login function allows users to access the system using email and password. The new checklist input and submission feature allows users to fill, edit, and submit eChecklists during final inspections, including attaching photos and notes. The list data feature allows the user to view the list of recorded data, while the update data feature allows the user to edit the data. The data history feature allows the user to see the time when the data was created and recorded. In addition, the data export feature allows admin users to export checklist records and reports in spreadsheet format for documentation or analysis. These functions are summarized in Table 2.

Table 2. Functionality Table

No	Section	Function Description
1.	Register	Allow user to register account in the system.
2.	Login	Allow user to access the system using email and password.
3.	New Checklist Input & Submission	Allow fill, edit, and submit eChecklists during final inspections, including attaching photo and notes.
4.	List Data	Allow the user to view the list of recorded data.
5.	Update Data	Allow the user to edit the data.

3.3 Hardware and Software Requirements

The hardware and software used in the development process consist of a web browser, code editor, database server, and laptop. To run this project, Microsoft Edge is used to view the code that is run on the code editor. To write the code for the program, Visual Studio Code is used with PHP as the programming language and Laravel as the framework. To store data, the system uses MySQL as the database with phpMyAdmin interface and XAMPP to access it. A laptop with at least 8GB RAM is used as the development device.

3.4 Use-Case Diagram

A use-case diagram is a visual representation that illustrates the interactions between actors and a system. It identifies who will use the system and what functions the system will provide. Use-case diagrams are helpful for understanding the behaviors of the system from the user's perspective and are commonly used during system analysis to gather requirements. The use-case diagram for the eChecklist System demonstrates the interaction flow when users utilize the web-based application for final product inspection in a coachbuilding company. The system involves three primary actors: Inspector (User), Supervisor, and Admin. Each actor has specific access and responsibilities that reflect their role within the inspection process. The Inspector is responsible for logging in to the system, selecting the assigned unit, and filling out the final inspection checklist. They can input item status, add notes, and attach photos to support the inspection results. The Supervisor reviews the submitted checklists from inspectors. If the data is complete and correct, the supervisor approves the checklist; otherwise, they return it with revision notes. The admin has full access to manage users and roles, update checklist templates, and monitor system-wide activity. The admin can also export inspection data and generate reports for decision-making. The system also enables authentication, role-based access, and real-time

data storage to ensure accuracy and accountability. Additionally, users with sufficient privileges can export checklist results in spreadsheet format for reporting or documentation purposes. The detailed use-case diagram of the eChecklist system is shown in Figure 3.

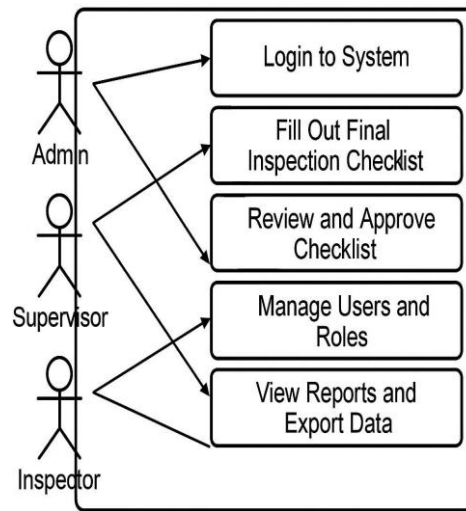


Figure 3. Use-Case Diagram of echecklist

4. RESULTS AND DISCUSSIONS

4.1 User Interface Design

The user interface design is an important component of an application that greatly affects the user experience because the user will directly use the application. A well-designed and attractive user interface will make it easier for users to be interested in using the application, and conversely, a poorly designed and unattractive user interface will make it difficult and uninteresting for users to use the application. Therefore, a good interface design not only serves to make the website look better but also helps users when using the website.

The eChecklist system consists of several main interfaces. The login page is the first page accessed by users and consists of images, text, and form components such as the logo, login page title, email text box, password text box, and button for account registration. The register page provides several text boxes for inputting user data such as name, email, password, and confirm password.

The admin dashboard provides full control over system configuration and user management. From this interface, the admin can manage user accounts, assign roles, create or update checklist templates, and monitor overall system usage. The layout includes quick access panels for managing users, viewing reports, and system statistics. The supervisor dashboard focuses on managing and validating submitted checklists. Supervisors can see pending inspection reports, filter submissions by date or status, and access detailed checklist forms. From this dashboard, they can approve or return inspection results to inspectors with feedback.

The inspector or user dashboard is centered around inspection tasks. Inspectors can view the list of units assigned to them, initiate the inspection process, and submit completed checklists. The final inspection checklist form is used by inspectors to perform the final inspection of assigned units. It displays a checklist form containing items that must be filled with status options such as "Pass", "Fail", or "Not Applicable", and inspectors can also add supporting notes or upload images for each item. Meanwhile, supervisors use the checklist validation interface to review checklists submitted by inspectors and either approve them or return them with comments for revision.

In addition, the admin panel includes a user and role management interface that enables admins to add new users, update user data, assign roles, or remove users if necessary. There is also a checklist template management interface that allows admins to create or update checklist templates by defining item names, categories, and expected conditions. Supervisors and admins can access a report and export interface where inspection data can be filtered based on parameters such as date, status, or inspector, and exported in Excel format for documentation or further analysis.

4.2 Database Design

Entity Relationship Diagram is very important in assisting the development process because it helps identify the tables used in this project. Entity Relationship Diagram has also been widely used in structured analysis and conceptual modeling. In this system, the ERD is used to describe the relationship between the data entities that support the eChecklist application.

Several databases are needed by the eChecklist system. The database is used to store all project data, including user accounts, checklist records, validation data, and inspection history. This design supports the system's operational processes by ensuring that all data can be stored, connected, and retrieved properly when needed. The Entity Relationship Diagram of the system is shown in Figure 4, while the database structure and related information are presented in the tables included in this section.

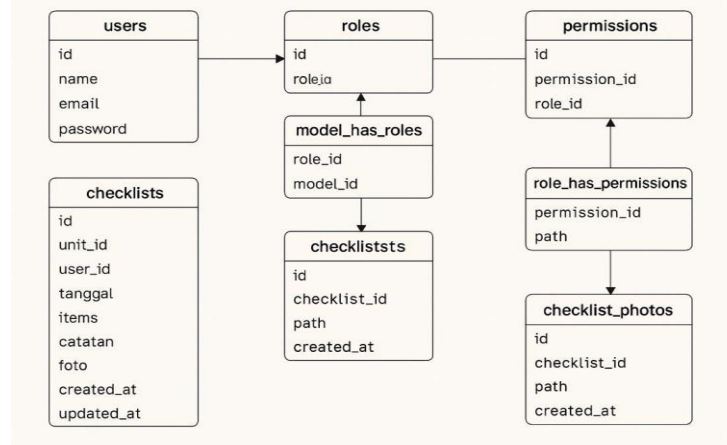


Figure 4. Entity Relationship Diagram

4.3 Testing Environment

The Entity Relationship Diagram of the system and the database design are shown in the figures and tables included in this section. The Web-Based eChecklist System for Final Product Inspection was tested using the following environment: Laptop HP Pavilion Gaming 15 dk1064tx, Windows 11, RAM 8 GB, and Google Chrome as the web browser. This environment was used to run and evaluate the system during implementation and testing.

4.4 Testing Scenario and Result

Testing scenarios were conducted to check whether all the features of the application were working properly and appropriately. The tests covered several main functionalities of the system, including user login, role-based access, checklist form submission, checklist validation by supervisor, checklist status tracking, user and role management, and report generation. The test results show that login with valid credentials successfully redirected users to their respective dashboards. Access restriction based on user role worked as expected, where unauthorized access was denied. Checklist forms could be filled and submitted successfully, including pass or fail results for each item. Supervisors were also able to validate submitted checklists properly using the validation interface provided by the system. The checklist status tracking feature displayed the correct validation status. Admin users were able to manage users and roles successfully, and the report generation feature produced downloadable reports as expected. Overall, all tested functionalities produced the expected outcomes. The system demonstrated consistent behavior and reliability, confirming that it is ready to support the operational needs of final product inspection in the coachbuilding company.

Testing scenarios were conducted to check whether all the features of the application were working properly and appropriately. The tests covered several main functionalities of the system, including user login, role-based access, checklist form submission, checklist validation by supervisor, checklist status tracking, user and role management, and report generation. The detailed testing scenarios and results are presented in Table 3.

Table 3. Testing Scenario and Result

No	Functionality	Scenario	Expected Result	Actual Result	Status
1.	User Login	Login with valid credentials for admin/user/supervisor	User is redirected to respective dashboard	User successfully redirected to correct dashboard	Passed
2.	Role-Based Access	Access restricted pages based on role	Unauthorized roles are denied access	Access denied as expected	Passed
3.	Checklist Form Submission	Fill and submit final inspection checklist	Checklist saved and submitted successfully	Checklist saved with pass or fail per item	Passed
4.	Checklist Validation by Supervisor	Supervisor views and validates submitted checklist	Validation interface with accept/reject options appears	Supervisor can validate checklist properly	Passed
5.	Checklist Status Tracking	View status (✓ or ✗)	Status is displayed accordingly	Status displayed correctly based on validation	Passed
6.	Manage Users and Roles	Admin adds, edits, or deletes users and roles	Users and roles updated accordingly	Admin successfully manages users and roles	Passed
7.	Report Generation	Export checklist reports for inspection history	Report is generated and downloadable	Report downloaded as expected	Passed

The fact that all tested functionalities achieved the expected results indicates that the system is functionally reliable for its intended operational scope. In practical terms, these results mean that the application can consistently support the main inspection workflow, from checklist entry and supervisor validation to role-restricted access and report generation. Compared with manual inspection recording, this reduces the risk of incomplete documentation, unauthorized modification, and delays in retrieving inspection records, thereby improving process traceability and administrative efficiency.

However, the current testing mainly confirms functional correctness rather than long-term performance under heavy usage. Therefore, the results should be interpreted as evidence that the system is operationally feasible and reliable at the feature level, while broader testing for scalability and usability remains a direction for future work

5. CONCLUSION

This study developed a web-based eChecklist system to digitize the final product inspection process in a coachbuilding company. The main contribution of this work lies in providing a lightweight and role-based inspection system tailored to the operational needs of coachbuilding final inspection, integrating checklist input, validation workflow, centralized record storage, historical tracking, and report export within a single web-based platform. The system was implemented using PHP, Laravel, and the Rapid Application Development approach, enabling iterative design and practical deployment. Functional testing showed that the core features of the system, including login, role-based access control, checklist submission, supervisor validation, status tracking, user and role management, and report generation, all worked as expected. These results indicate that the system is operationally reliable for supporting the intended inspection workflow.

In practical terms, the developed application improves inspection data traceability, reduces the limitations of manual recording, and supports faster access to inspection records for inspectors, supervisors, and administrators. By replacing paper-based or fragmented inspection records with a centralized digital workflow, the system can contribute to better process consistency, reduced administrative burden, and improved accountability in quality control activities.

Nevertheless, this study has several limitations. The testing mainly confirms feature-level functional correctness and does not yet evaluate long-term scalability, usability, or integration with external enterprise systems. In addition, the current system only supports limited export and internal factory use. Future work may extend the platform through mobile support, broader export options, real-time notifications, analytics dashboards, and integration with other production or inventory systems.

Although the system has been successfully implemented, there are several potential improvements that can be developed to enhance the functionality and scalability of the eChecklist platform. The following are future development suggestions:

- Improve and modernize the user interface (UI) to offer a more intuitive experience for all types of users.
- Expand export capabilities to support additional file formats such as CSV, JSON, and print-friendly layouts.
- Add a built-in user guide or onboarding feature to help new users understand how to operate the system effectively.
- Integrate with inventory or production systems to connect inspection results with other departments, as “Smart City can be seen as a collection of collaboration and the integration of several systems.”
- Enable real-time notifications for pending approvals or incomplete checklists.
- Implement analytics or dashboards to visualize checklist trends and unit failure rates.
- Add multilingual support to accommodate a diverse user base.
- Improve system security and software maintenance, since “it is crucial to regularly update the operating system and all software associated with log4j”.

These improvements can help the system evolve into a robust enterprise-level solution, ensuring better coordination between teams and higher operational efficiency in the coachbuilding industry.

REFERENCES

- [1] Su, D., & Yao, Y. (2016, May). Manufacturing as the Key Engine of Economic Growth for Middle-Income Economies. Asian Development Bank Institute (ADBI) Working Paper, (573). <http://www.adb.org/publications/manufacturing-key-engine-economic-growthmiddle-income-economies/>
- [2] R, A. C. (2024, February 18). What is Karoseri: Here's The Retail. Retrieved May 13, 2025, from <https://voi.id/en/otoinfo/357098>
- [3] M. Lubis and H. H. R. Ananza, “Analysis and Design of Policy and Standard Operating Procedure (SOP) for Information Technology in the Communication and Information Services Department,” in 2020 6th International Conference on Education and Technology (ICET), 2020.
- [4] Wulandari, S. K., & Rahmad, N. (2023, November). Perancangan Sistem Pelaporan Kerusakan Peralatan Tambang Berbasis Website Pada PT. Inti Bara Nusalima. 1(1), 64. ISSN 3030- 8011
- [5] Grobe, D. (n.d.). With Digital Quality Control, You Minimize Three Key Quality Risks. <https://operations1.com/en/blog/digital-quality-control-minimize-quality-risks>
- [6] Kresnadi, M. N. D., & Hamdala, I. I. (2021). Perancangan Sistem Informasi Pencatatan Kegiatan Produksi dan Administrasi Pemesanan di CV. Ki Selo Redjo. Sarjana Thesis, Universitas Brawijaya, 1. <http://repository.ub.ac.id/id/eprint/190057>
- [7] T. Syafiera, M. Lubis, R. W. Witjaksono, and H. D. Anggana, “The Means of Engagement (MOE) Model of the Agreement towards the Enterprise Resource Planning (ERP) Implementation,” in 2019 International Conference on Information Management and Technology (ICIMTech), 2019.
- [8] Holivets, S. (2024, December 9). Rapid App Developments: Full RAD Methodology Overview and Practical Advice. Retrieved May 13, 2025, from <https://maybe.works/blogs/rapid-application-development-rad>
- [9] Schneider, A. (2024, November 25). What is Rapid Application Development (RAD)? The Complete Guide for 2025. Retrieved May 13, 2025, from <https://monday.com/blog/rad/rapid-application-development-rad/>
- [10] GEM. (2024, April 15). Rapid Application Development: A Ground-Breaking Approach to Software Development. Retrieved May 13, 2025, from <https://gem-corp.tech/software-development/rapid-application-development/>

- [11] SixSigma.us. (2025, January 17). Quality Control in Manufacturing: Overview and Best Practices - SixSigma.us. Six Sigma. Retrieved May 14, 2025, from <https://www.6sigma.us/manufacturing/quality-control-in-manufacturing/>
- [12] Hutchinson. (2025). Coachbuilding. Retrieved May 14, 2025, from <https://www.hutchinson-engineering.co.uk/sectors/coachbuilding>
- [13] Zeiss. (n.d.). ZEISS Car Body Solutions: Efficient production of car bodies | automated. ZEISS. Retrieved May 14, 2025, from <https://www.zeiss.co.uk/metrology/industries/automotive/car-body-solutions.html>
- [14] Ecubix. (n.d.). eChecklist (eLoogbook). Retrieved May 14, 2025, from <https://www.ecubix.com/echecklist/>
- [15] Qima One. (n.d.). Manual Quality Control Management is Dangerous for Your Margins. Retrieved May 14, 2025, from <https://www.qimaone.com/resource-hub/art/digitizing-traditional-qc>
- [16] Punit, Shrivastava, D. V., & Pandey, D. A. (2024, April). A Comprehensive Study on Modern Database Management Systems. 5(4), 2133-2136. ISSN 2582-7421
- [17] Morris, K. C., Mitchell, M., Dabrowski, C., & Fong, E. (1992, December). Database Management Systems in Engineering. <https://www.sciencedirect.com/science/article/abs/pii/036083529290165G>
- [18] Malla Reddy College of Engineering & Technology. (n.d.). DATABASE MANAGEMENT SYSTEM (DBMS). mrcet.ac. Retrieved May 14, 2025, from https://mrcet.com/downloads/digital_notes/ECE/III%20Year/DATABASE%20MANAGEMENT%20SYSTEMS.pdf
- [19] A. R. Ahlan and M. Lubis, "Information Security Awareness in University: Maintaining Learnability, Performance and Adaptability through Roles of Responsibility," in 2015 International Conference on Data and Software Engineering (ICoDSE), 2015.
- [20] Ghosh, S. P. (2015, September 14). An Application of Statistical Databases in Manufacturing Testing. An application of statistical databases in manufacturing testing
- [21] NIST Computer Security Resource Center. (n.d.). Role-based Access Control: an Overview. NIST Computer Security Resource Center. Retrieved May 14, 2025, from <https://csrc.nist.gov/CSRC/media/Presentations/Role-based-Access-Control-an-Overview/images-media/alvarez.pdf>
- [22] Zhou, K., Lv, M., & Ren, B. (2009, June). Access Control for Manufacturing Process in Networked Manufacturing Environment. 2, 107-116. https://www.scirp.org/pdf/JSSM20090200006_27740152.pdf
- [23] M. Lubis, M. Kartiwi, and S. Zuhuda, "Privacy and Personal Data Protection in Electronic Voting: Factors and Measures," TELKOMNIKA (Telecommunication Computing Electronics and Control), vol. 15, no. 1, pp. 512–521, Mar. 2017, doi: 10.12928/TELKOMNIKA.v15i1.3804.
- [24] Y. A. Prasetyo and M. Lubis, "Smart City Architecture Development Methodology (SCADM): A Meta-Analysis Using SOA-EA and SoS Approach," SAGE Open, vol. 10, no. 2, 2020, doi: 10.1177/2158244020919528
- [25] F. Maulana and H. Fajri, "Unmasking log4j's Vulnerability: Protecting Systems against Exploitation through Ethical Hacking and Cyberlaw Perspectives," in 2023 9th International Conference on Computer and Communication Engineering (ICCCE), 2023, doi: 10.1109/ICCCE58854.2023.10246082.
- [26] S. Nurindasari *et al.*, *Keselamatan Kerja Dan Kesehatan Mental*. Yayasan Kita Menulis, 2025
- [27] M. F. Safitra, M. I. Alhari, D. P. Putri, M. Lubis, H. Fakhurroja, and V. Satria Praditha, "Metaverse Trend: Definition, Application, Opportunities, Law, and Ethics," in 2023 IEEE International Conference on Computing (ICOCO), 2023, pp. 160–165. doi: 10.1109/ICOCO59262.2023.10397864.
- [28] M. and L. A. R. and A. M. I. Safitra Muhammad Fakhrol and Lubis, "The Need for Energy-Efficient Networks: A Review of Green Communication Systems and Network Architectures," in *Intelligent Sustainable Systems*, D. S. and M. D. and J. A. Nagar Atulya K. and Jat, Ed., Singapore: Springer Nature Singapore, 2024, pp. 127–136.