

Original Article

Design and Development of a Centralized Asset Maintenance System and Reporting Downtime Web-Based

Tirwanda Edho D.¹, Maridiana², Wicaksono Kodrat W.³, Setiawan Yanto⁴

^{1,2,3}Computer Science Department, BINUS University, Jakarta, Indonesia.

⁴Corresponding Author : yanto.setiawan@binus.ac.id

Received: 07 January 2023

Revised: 17 March 2023

Accepted: 01 April 2023

Published: 25 April 2023

Abstract - The rapid development of information technology encourages companies in various fields to develop to stay caught up. One of the things that must be done is digitization, where manual processes can be automated using the system. The use of technology in a company is crucial because it can save time and increase profits, including in manufacturing companies. This research was conducted at a manufacturing company engaged in the assembly and distribution of bicycles. Currently, some processes are still done manually and not integrated, causing problems and not maximizing the company's performance. In the last six months, 377 minutes of machine downtime were counted, of which 75% were machine repair processes, 19% were corrective maintenance submission processes, and 6% were spare part search processes. This research aimed to design and build a system that can facilitate companies in asset management and automate several processes, such as submitting corrective maintenance, finding spare parts, and handling machine downtime. The SDLC (System Development Life Cycle) method was used in developing this system with the waterfall model and OOAD (Object Oriented Analysis Design) analysis. The result of this research is a web-based system that can reduce machine downtime, thereby increasing production and company performance by reducing the corrective maintenance submission process time by 50%, reducing spare part search time by 28%, and reducing data error.

Keywords - Asset management, Downtime, Manufacture, OOAD, SDLC, Waterfall.

1. Introduction

Rapid technological advancements have heavily influenced the world, including the manufacturing industry. The traditional way of managing asset maintenance and reporting downtime has become outdated and no longer effective in meeting the demands of modern businesses. This has resulted in numerous challenges the industry faces, such as manual and unintegrated processes, the potential for data writing errors, and incorrect indicator data. The high downtime of machines is a common issue faced by many manufacturing companies, and the causes include machine repair, corrective maintenance, and spare part replacements. The manufacturing industry requires a centralized solution to address these challenges that can effectively manage asset maintenance and report downtime in real-time. This is where the "Design and Development of Centralized Asset Maintenance System and Reporting Downtime Web-Based" paper comes into play, aiming to provide a solution to these pressing problems faced by businesses.

PT XYZ Manufacturing is a manufacturing company engaged in the assembly and distribution of bicycles, with a total production of 1.3 million units per year. The high

number of production is supported by good performance in each Division, especially the Production and the Engineering Division. The Production Division is responsible for the main production process, namely assembling components or parts to become one unit of bicycles ready for distribution. The Engineering Division is responsible for production support, namely adjusting and controlling all assets needed in the bicycle assembly process, including production equipment and machines. In addition, the Engineering Division is also responsible for the performance of the production machines used in bicycle assembly by carrying out Preventive Maintenance, Corrective Maintenance, and Reporting Production Machine Downtime. Preventive maintenance is conducted at regular intervals according to a predetermined schedule, during which various maintenance tasks are performed, including but not limited to inspections, repairs, replacements, cleaning, lubrication, and adjustments to ensure proper functioning and balance of equipment or systems [1]. Corrective maintenance is one of the treatments in which maintenance actions such as repair or replacement are carried out on the machine after the machine is damaged [2].



OEE measurement is based on three main factors: Availability, Performance, and Quality. Each factor represents a different perspective of how closely a manufacturing process can achieve production optimization. Availability is the percentage of time during which a process (or equipment) is available to run [3]. Availability Loss includes all events that stop production, either planned or unplanned, for a considerable amount of time. The causes of unplanned downtime recorded in this company are machine failure, unplanned maintenance, and spare part replacement [5].

The main issue faced by PT XYZ Manufacturing in the last six months is the high downtime of the machines, with 377 minutes recorded. 75% of the downtime is due to machine repair, 19% is due to the submission of corrective maintenance, and 6% is due to searching for spare parts. The high downtime is caused by manual and unintegrated processes, such as the submission of corrective maintenance using physical forms by the Production Division and the downtime reporting using Microsoft Excel by the Engineering Division, without any asset management system in place. These manual processes have the potential for data writing errors, leading to incorrect data and causing errors in engine indicator data, such as Mean Time Between Failure (MTBF) and Mean Time to Repair (MTTR). The average error in data recording in the last six months has been 9%.

The objective of this research is to provide businesses with a solution that improves their asset maintenance and downtime reporting processes through the use of a centralized asset maintenance system and reporting downtime web-based solution. The system is expected to increase the accuracy of data recording and reduce errors in engine indicator data such as MTBF and MTTR. It also provides a framework for other manufacturing companies to improve their asset management and downtime reporting processes by implementing a centralized web-based solution.

However, it is important to note that the research has limitations, such as only focusing on implementing the system within one manufacturing company, PT XYZ Manufacturing, and only covering the reporting of downtime and maintenance of production machines and not including other assets such as buildings and infrastructure. The methodology used in this research includes analysing business requirements, designing and implementing the system, and evaluating the results. The research presents the results of designing and developing the centralized asset maintenance system and reporting downtime web-based solution. The proposed solution offers several unique features, including real-time data recording, improved accuracy and efficiency, and increased transparency in the asset maintenance and downtime reporting processes. These features significantly improve the manufacturing industry's asset management and downtime reporting processes, providing businesses with a centralized, web-based solution that streamlines processes, reduces errors, and enables more effective decision-making.

2. Related Works

In addition to the existing literature, this research presents a comprehensive solution for asset management and reporting downtime in manufacturing companies. While previous studies have focused on specific aspects such as monitoring, predictive maintenance, or information management, this study offers a complete solution to improve maintenance efficiency and reliability. By implementing a digitized asset management system, organizations can streamline their maintenance operations, increase visibility, and make more informed decisions based on data analysis. This research compares and analyzes the gap between previous studies and presents a clear roadmap for future research in this field. Ultimately, this study contributes to the growing body of literature on the benefits of digitizing asset management in manufacturing.

Table 1. Gap Research

| Topics | Design and Development of a Centralized Asset Maintenance System | Cloud-Based Asset Monitoring | Information as a Key Dimension to Develop Industrial Asset Management |
|---------------------|----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Focus | Designing a system to manage assets and reporting maintenance activities for better efficiency and visibility. | Predictive maintenance and cloud-based monitoring of assets in an industrial IoT system to improve reliability and productivity | The importance of information management and data analysis for improving maintenance efficiency and reliability. |
| Gap Research | May not address predictive maintenance and cloud-based monitoring. | May not focus on the design and development aspect of the system. | May not focus on the design and development of the system or the practical challenges involved. |
| Implication | Can provide insights for organizations to improve the efficiency and visibility of maintenance activities. | Can provide insights for predictive maintenance and cloud-based monitoring. | Can provide insights for information management and data analysis. |

The presentation of related work in this paper will be presented in the form of a gap research table, comparing it with previous research. This method of presenting related work will provide a clear and concise comparison between the current study and previous studies, allowing for a thorough understanding of the existing gaps in the field. By presenting the related work in the form of a gap research table, the authors aim to provide a comprehensive overview of the current state of research and highlight areas that have yet to be explored, thereby offering a roadmap for future research in the field. The paper will compare “Cloud-Based Asset Monitoring and Predictive Maintenance in an Industrial IoT System”[6] with "Design and Development of Centralized Asset Maintenance System and Reporting Downtime Web-Based” and “Information as a key dimension to develop industrial asset management in manufacturing”[7].

This research has the unique feature of reporting downtime and asset management. This provides a comprehensive solution for organizations in terms of maintenance management. Unlike previous research, which may have focused on a specific aspect, such as monitoring or predictive maintenance, this research provides a complete solution for organizations to manage and report downtime and keep track of their assets. This feature helps organizations to streamline their maintenance operations, increase visibility and efficiency, and ultimately improve their overall maintenance management.

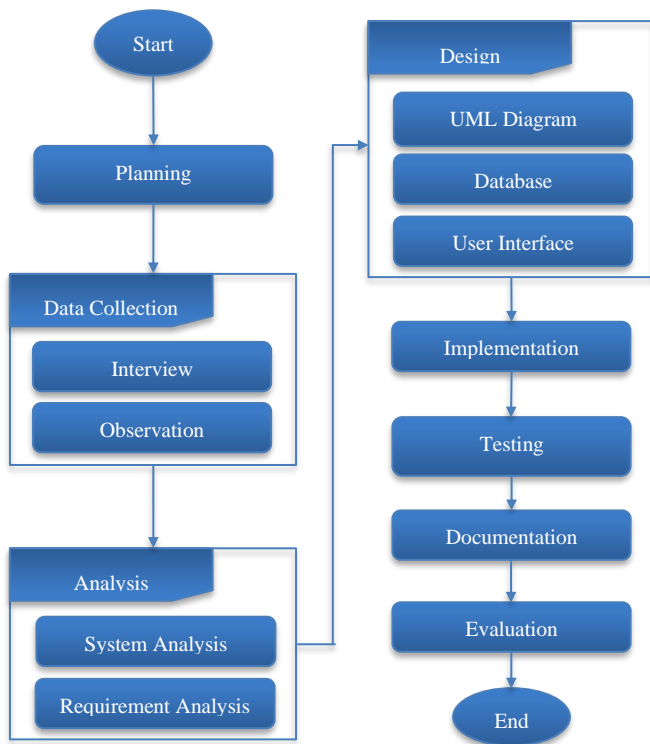


Fig. 1 Research Methodology

3. Methodology

The methodology used in the Design and Manufacturing of a Centralized Asset Maintenance System and Web-Based Machine Downtime Reporting is the Waterfall SDLC model. The waterfall model is a basic software development model where all SDLC phases will run in sequential order [9]. This method is used because user requirements have been identified, and system flow is certain, so there are no changes in the requirements during the development [10], [11,12]. The waterfall methodology flowchart of this research is shown in Figure 1.

The following are the 8 SDLC stages in designing and developing a web-based system using the waterfall method.

3.1. Planning

The planning stage is determining the tasks or activities, the estimated time for completion, the resources needed, the scheduling of activities to be carried out and monitoring the work process.

3.2. Data Collection

Data collection is one of the most important stages in a study. The quality of research, both quantitative and qualitative research, is largely determined by the quality and completeness of the data collected by the researcher [13], [14].

Data collection was carried out by observing and interviewing staff directly involved in managing assets and handling corrective maintenance at the company.

This process is carried out to collect supporting data so the author can understand user requirements. From this data collection process, information is obtained regarding the flow of asset management processes, corrective maintenance processes and constraints from the company.

3.3. Analysis

At the analysis stage, it is explained about the current condition of the system based on the data that has been collected. The data will be used as a reference in analyzing needs and proposing solutions to problems that occur in the company. The analysis process can determine the quality of the software [15]. A good analysis can produce good software and vice versa.

3.4. Design

The design process consists of making UML Diagrams which contain Use Case Diagrams, Activity Diagrams, Sequence Diagrams, and Class Diagrams. User Interface Design provides an overview of the system to be developed. Database design in the form of an ERD (Entity Relationship Diagram), which describes the relationships between tables in the system.

3.5. Implementation

In this phase, the code was written and compiled as an operational application in which the database and text file was created based on the design made [17]. This software development uses Java on the back end and ReactJS on the front end, and REST API is used in the integration process to connect the back end and front end.

3.6. Testing

The system that has been built will be tested to ensure that the system will work properly with all specified operations [16,18]. Testing consists of black box testing and User Acceptance Testing (UAT). Black box testing to review the input and output of the system. User Acceptance Testing (UAT) to ensure that the system functionally meets user requirements.

3.7. Documentation

Documentation is carried out to provide information to the user regarding how to implement and operate the system. It can be used as a reference for future system development. Practitioners identify quality requirements documentation as important and perceive it as contributing to ensuring quality, clarifying quality requirements, and facilitating decision-making[19]. Documentation consists of software specifications, hardware, and procedures for using the system.

3.8. Evaluation

Evaluation is carried out to get feedback on the system that the user has tested. This process is an identification to measure and assess system performance functionally and ease of system operation by the user.

4. Results and Discussion

This research was conducted at PT XYZ Manufacturing, a manufacturing company engaged in the assembly and distribution of bicycles. Based on the observations and interviews, it was found that several processes were still being carried out manually and had not been integrated, causing problems and not maximizing the company's performance. The process includes submitting corrective maintenance by the Production Division using a physical form on paper which is then sent to the Engineering Division; the Engineering division does not yet have an asset management system to facilitate the search for spare parts; and downtime reporting by the Engineering Division still using Microsoft Excel based on the corrective maintenance submission form. In addition, the manual data reporting process can result in data writing errors between the corrective maintenance submission form and downtime reporting. Incorrect data can cause errors in engine indicator data such as MTBF (Mean Time Between Failure), MTTR

(Mean Time to Repair), Reliability, and Availability. The average data recording error in the last 6 months was 9%, and machine downtime was 277 minutes. Of these downtime machines, 75% were machine repair processes, 19% were corrective maintenance submission processes, and 6% were spare part search processes. This research was conducted to reduce high machine downtime and facilitate companies in asset management by utilizing technology.

The developed information system is web-based, with primary functions for asset management and machine downtime reporting. In the system to be built, there are three user roles, including the role of the production staff who has permission to submit corrective maintenance of machines to the engineering division; the role of engineering staff who is responsible for following up on corrective maintenance submissions from the production division and are responsible for managing company asset data; and role of manager or coordinators who are responsible for the process of approving corrective maintenance data and monitoring machine downtime data.

4.1. Use Case Diagram

Use cases are described textually in the form of use case scenarios to explain the interactions that occur between actors and the system [8], [21], [22]. The following is a Use Case Diagram of a centralized asset maintenance system application and website-based machine downtime reporting in Figure 2.

4.2. Class Diagram

The class diagram is a diagram that is useful for describing system structures that contain classes, attributes, methods, and relationships between the object [4], [24], [26]. The class diagram for this system can be seen in Figure 3.

4.3. Result

The centralized asset maintenance system and machine downtime reporting is a web-based system built with a simple interface and features that meet user requirements. The user interface is part of the system that acts as an intermediary between users and facilitates users to interact with the system efficiently [27], [28]. A suitable user interface helps users efficiently operate the system. The following is a user interface of the system that was successfully developed.

The login page is shown in Figure 4. The login page is the first page that appears when the user accesses the website link. To enter the web application, the user must fill in the username and password registered and stored in the system database.

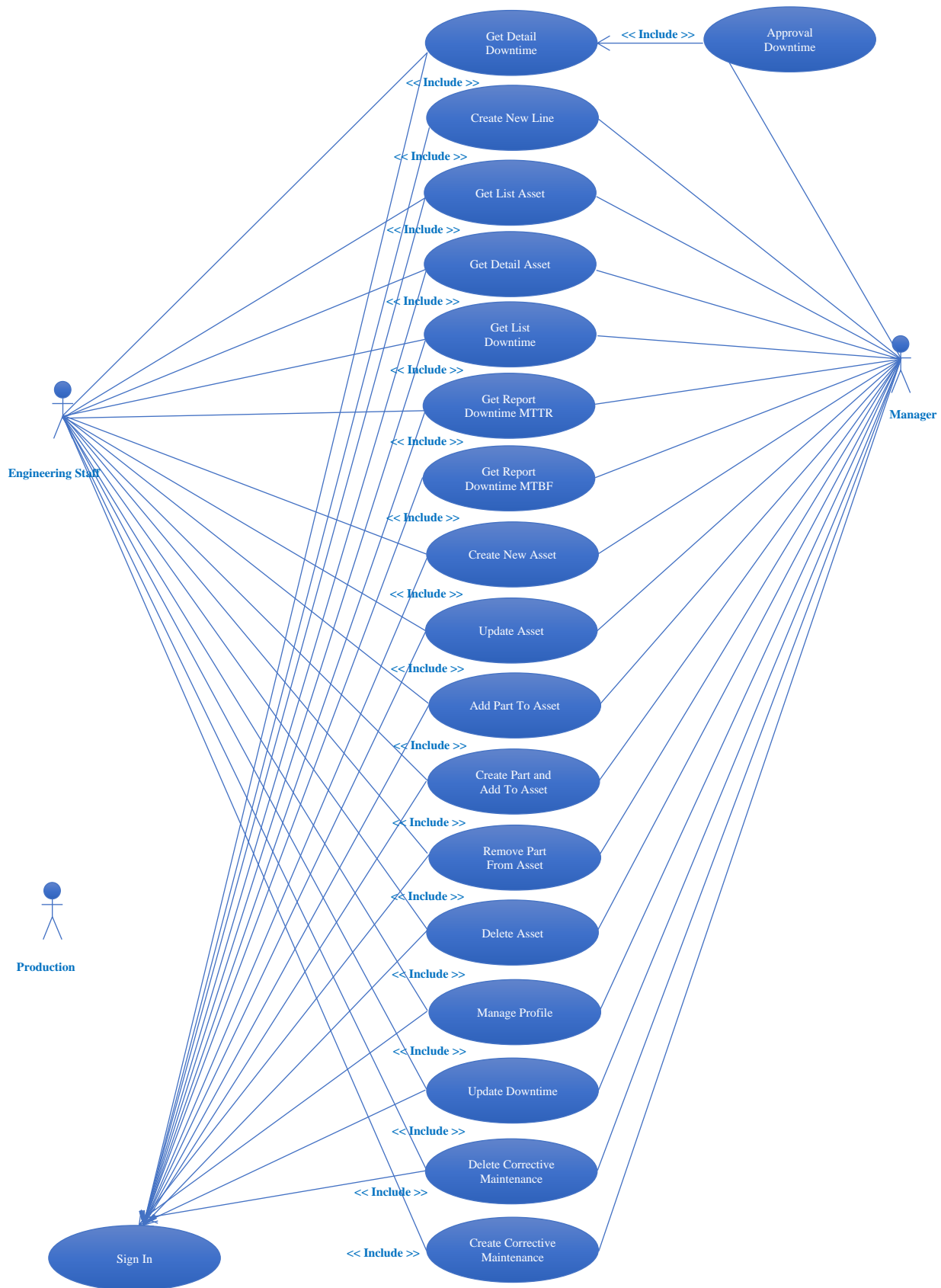


Fig. 2 Use Case Diagram

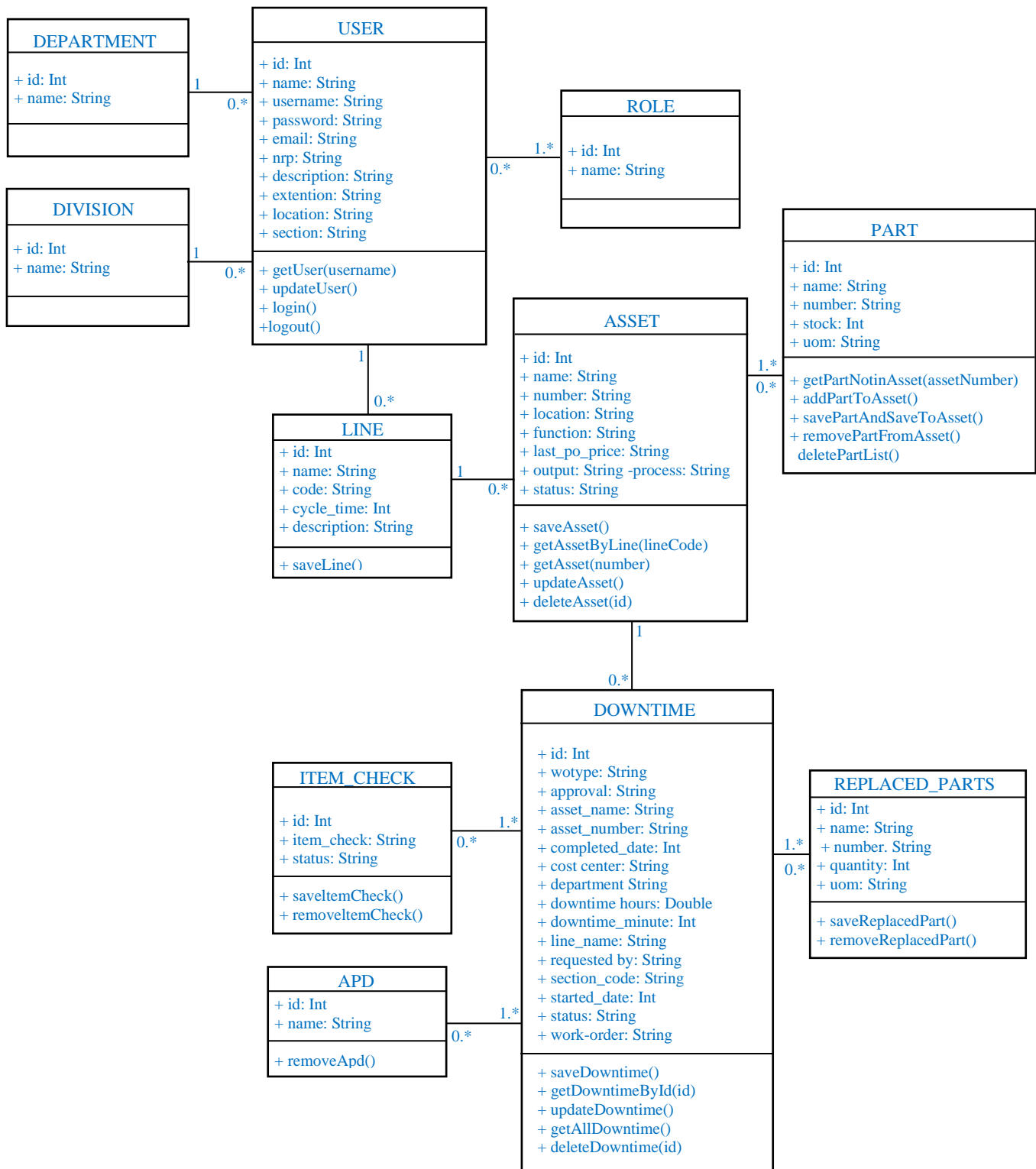


Fig. 3 Class Diagram

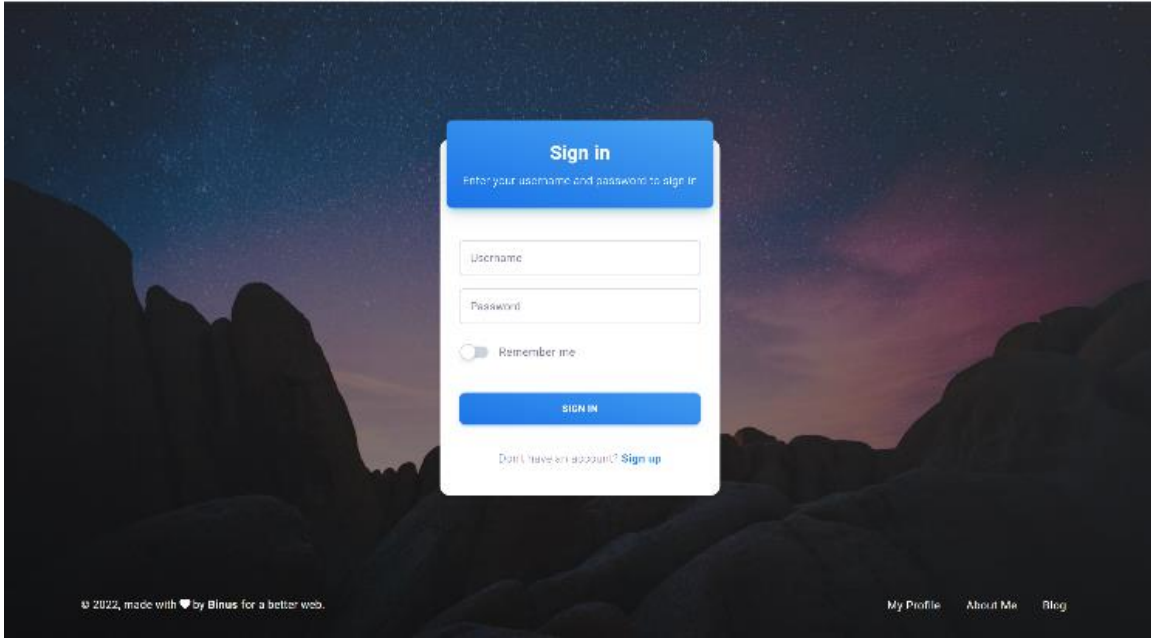


Fig. 4 Login Page

The dashboard page is shown in Figure 5. On the dashboard page, the user can see a summary of information about the number of assets in each line, the total downtime in each line and the 5 machines with the highest downtime. On

the left side of the screen, there is information on what menus can be accessed by the user, namely the application, report, and assets menus.

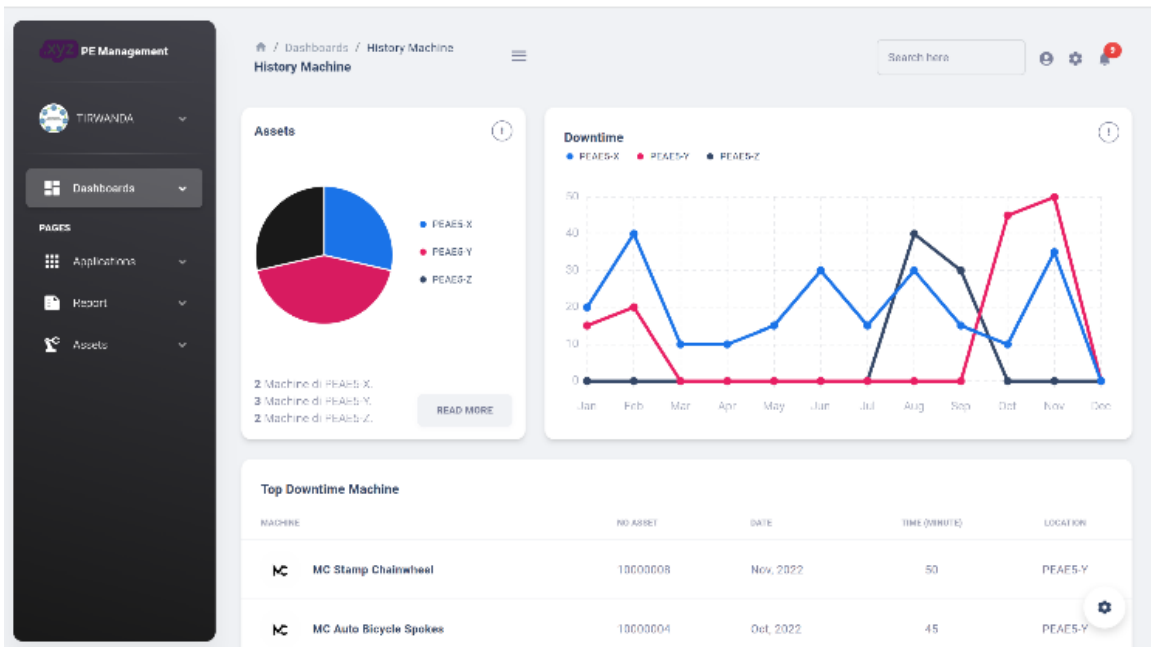


Fig. 5 Dashboard Page

The "Create New Line" page is shown in Figure 6. On the "Create New Line" page displays a form for filling in

new production line data. The user is asked to fill in all the data in the form.

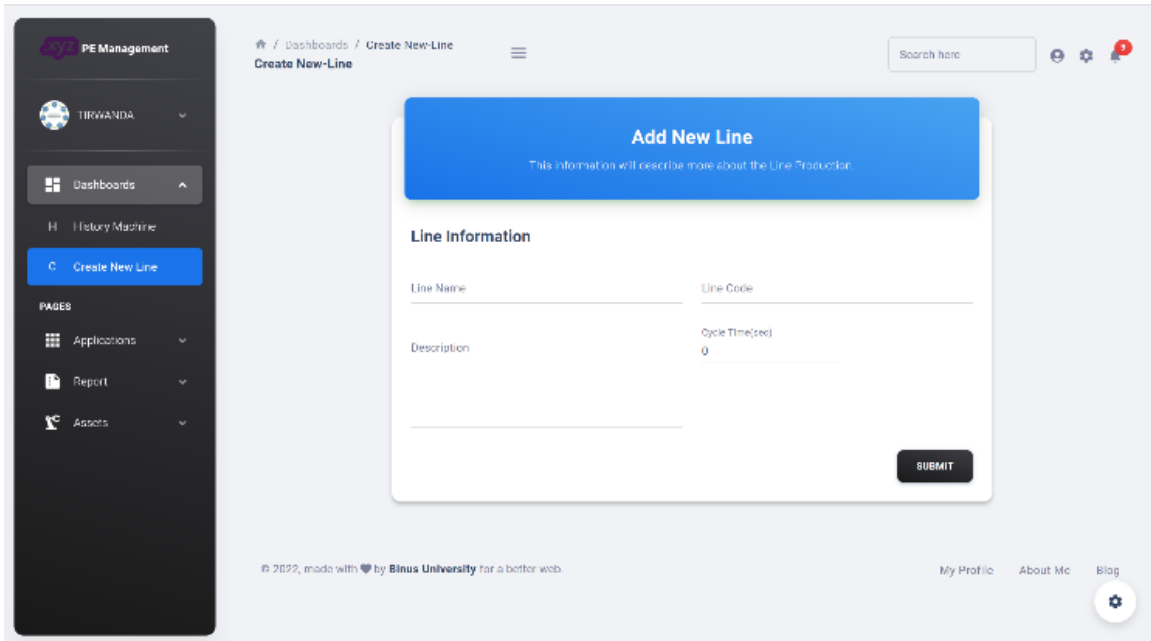


Fig. 6 Create New Line Page

The "Add New Asset" page is shown in **Figure 7**. On the "Add New Asset" page displays a form for filling in new assets that will be added to the production line; in this form, the user is asked to fill in new asset data that will be added to

the production line; these data include asset number, asset name, status, asset location, asset function, process, output, and last PO price.

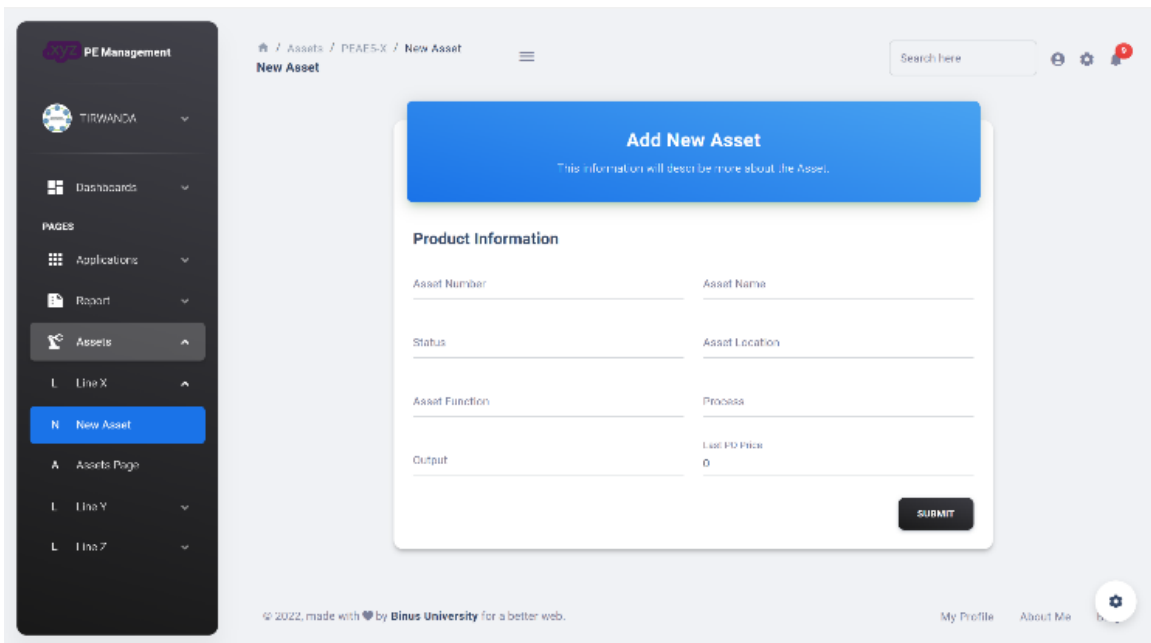


Fig. 7 New Asset Page

The "List Asset" page is shown in Figure 8. The "List Asset" page displays a table listing assets contained in the production line. Each asset displays information on the asset name, asset number, status and actions in the form of three buttons for delete, edit and view. Each button has a different

function, namely the delete button functions to delete assets, the edit button functions to change information or data on these assets and the view button functions to view detailed information on these assets.

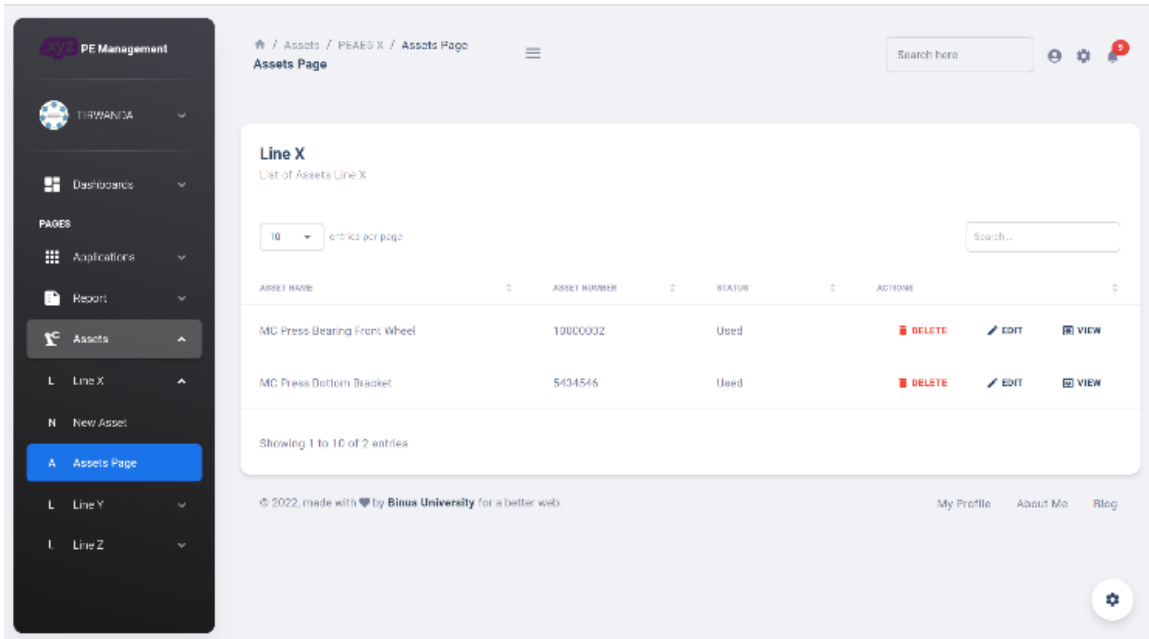


Fig. 8 List Asset Page

The "Edit Asset" page is shown in Figure 9. To be able to go to the "Edit Asset" page, the user must click the "edit" button on the "List Asset" page. On the "Edit Asset" page displays an asset edit form that has been filled with asset information, namely asset name, asset number, asset

location, asset function, machine process, machine output, asset status and last PO price. On the form contained on this page, the user can change the asset information with the latest information and can save the latest information by clicking the "Update Asset" button.

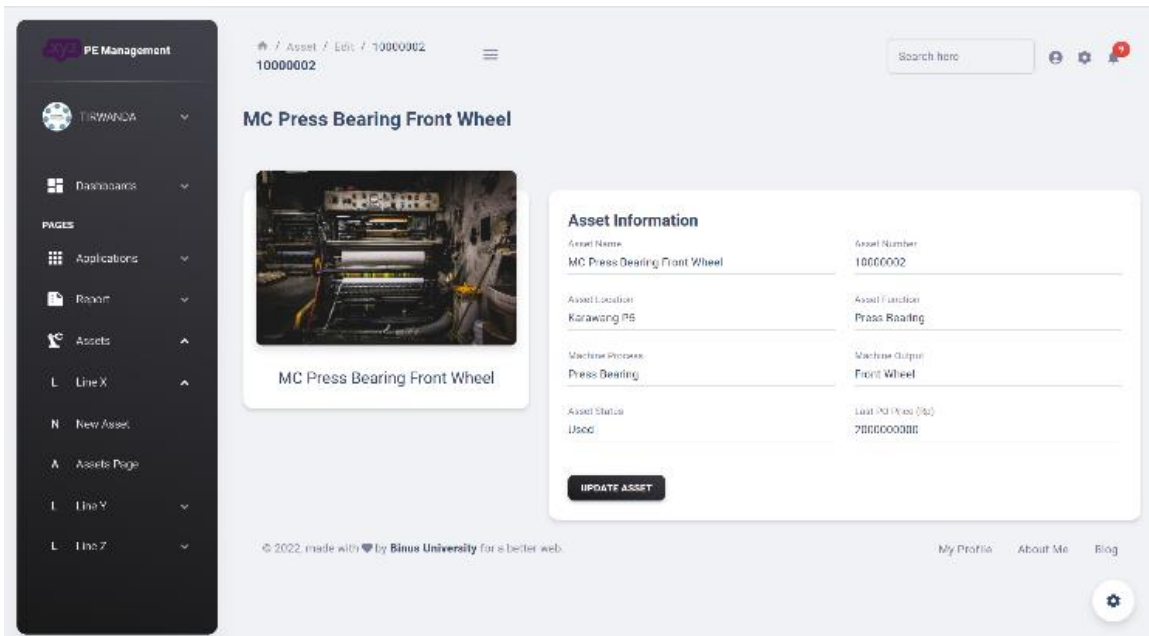


Fig. 9 Edit Asset Page

The "Asset Details" page is shown in Figure 10. The "Asset Details" page displays detailed asset information,

namely the asset image, name, price, number, location and a list of parts the asset uses.

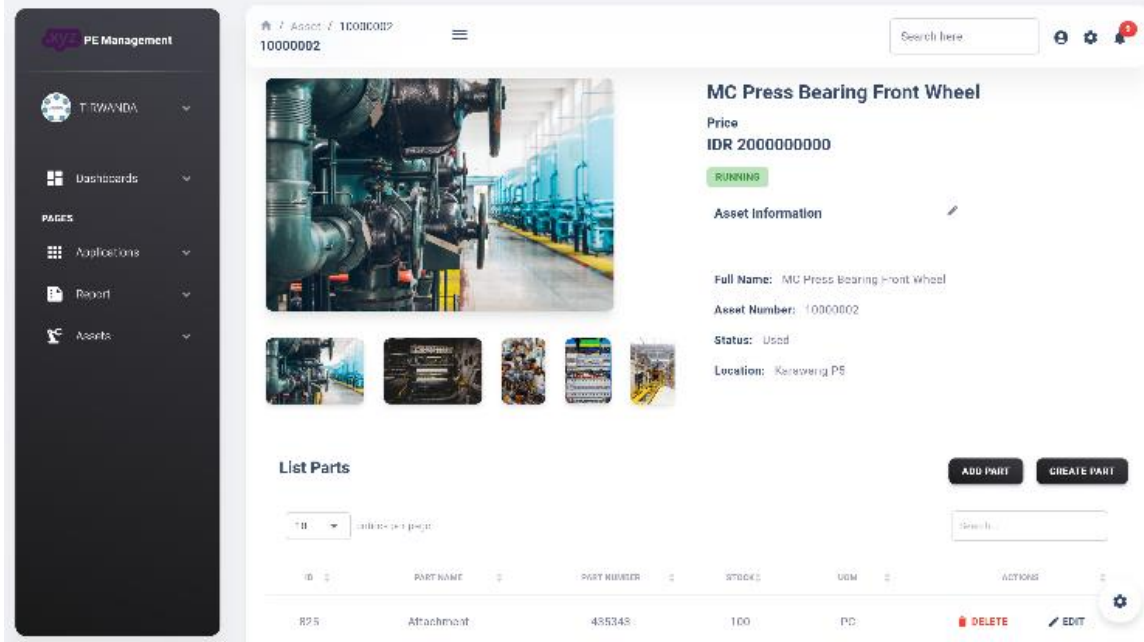


Fig. 10 Asset Detail Page

In the part list section on the "Asset Detail" page shown in Figure 10, there are two buttons ", add part" and "create part", and there is an information table listing the parts used

by the asset in a table containing information such as part ID, part name, part number, stock, uom. There are two buttons, namely, the delete button and the edit button.

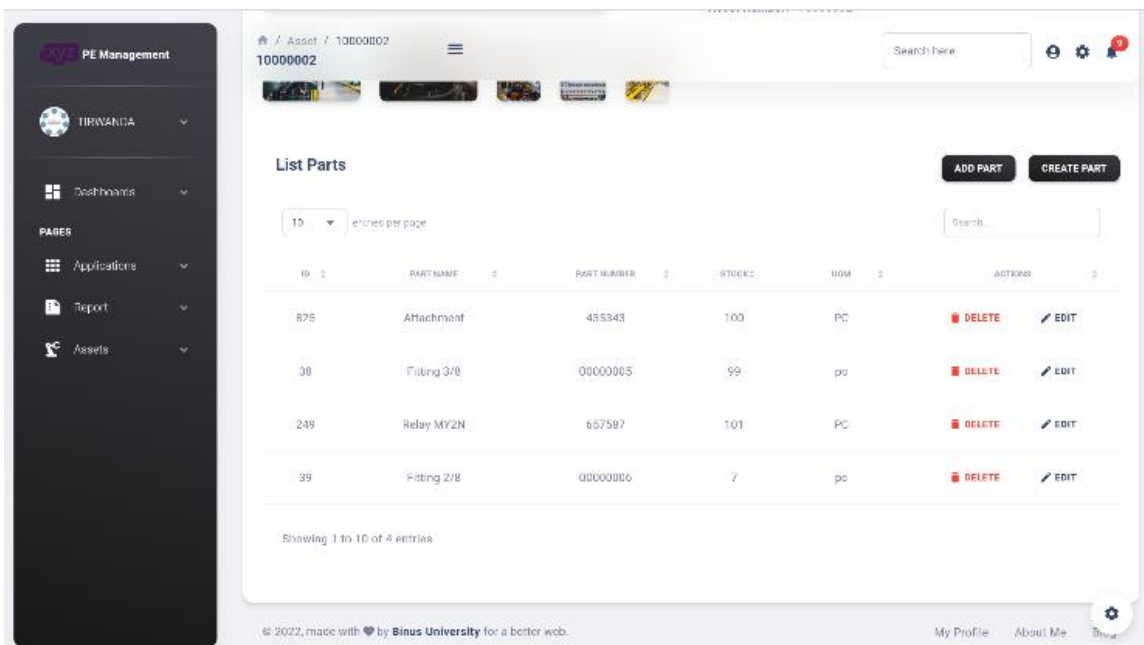


Fig. 11 List Part

In the "List Part" view, there are two buttons, namely the "Add Part" button and the "Create Part" button, each of which has a different function; the "Add Part" button

functions to add a part from the master part (Figure 12) to the part list, and the "Create Part" button functions to create a new part that does not yet exist in the master part.

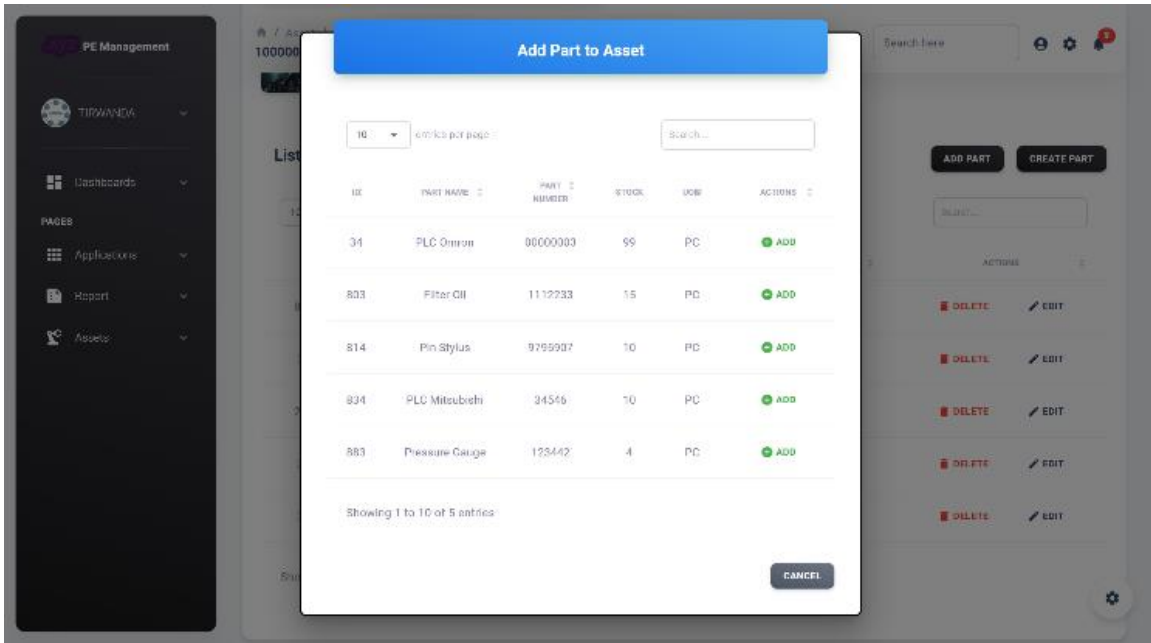


Fig. 12 Master Part Modal

The "Add Part To Asset" modal shown in Figure 14 displays a master part table containing information on ID, part name, part number, stock, uom, and action in the form of

an "Add" button. At the bottom of the modal, it displays a "Cancel" button.

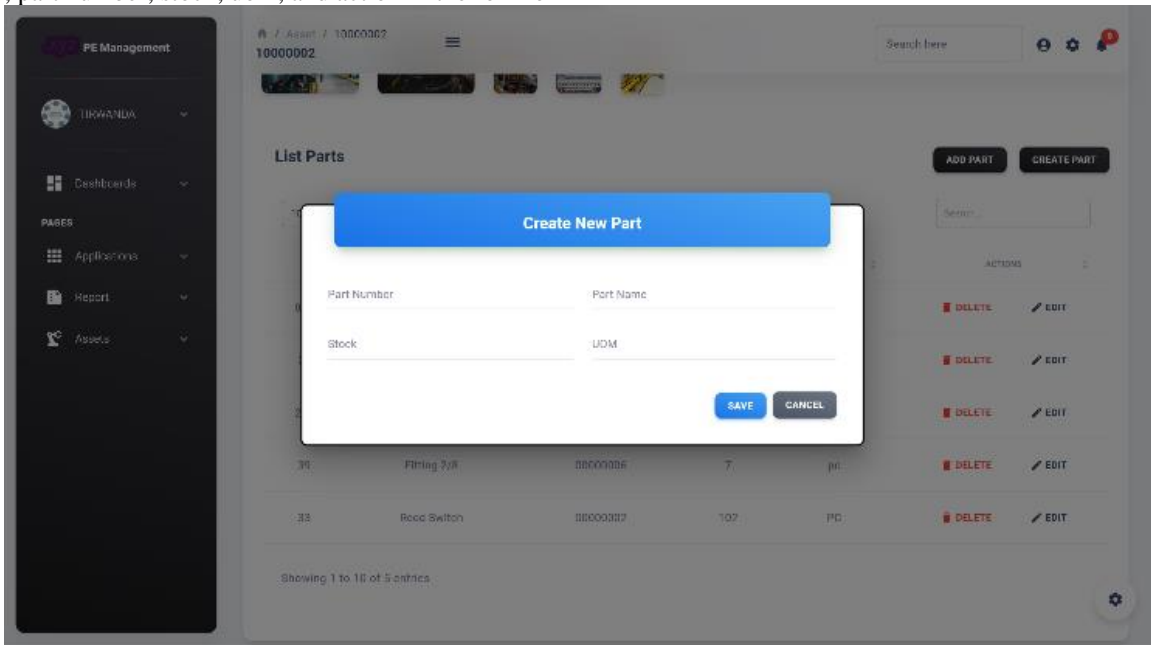


Fig. 13 Create Part Modal

The "Create New Part" modal shown in Figure 13 displays a form for filling in new part information that will be added to the part list; in this form, the user is asked to fill in the new part data; the data includes the part number, part name, stock, and UOM. At the bottom of the "Create New Part" modal, there are two buttons, namely the "Save" button

and the "Cancel" button; the "Save" button serves to save the new part data into the part list, and the "Cancel" button serves to close the modal.

The "Create CM" page is shown in Figure 14 and Figure 15. The "Create CM" page displays the corrective

maintenance creation form; in this form, the user is asked to fill in information related to downtime that occurs, including line, asset name, started date, completed date, work order, status, department, cost center, section code, request by. The information that may not be filled in includes the part list,

item check, and apd. At the bottom of the corrective maintenance form is a submit button that functions to save the downtime information that has been filled in on the corrective maintenance form.

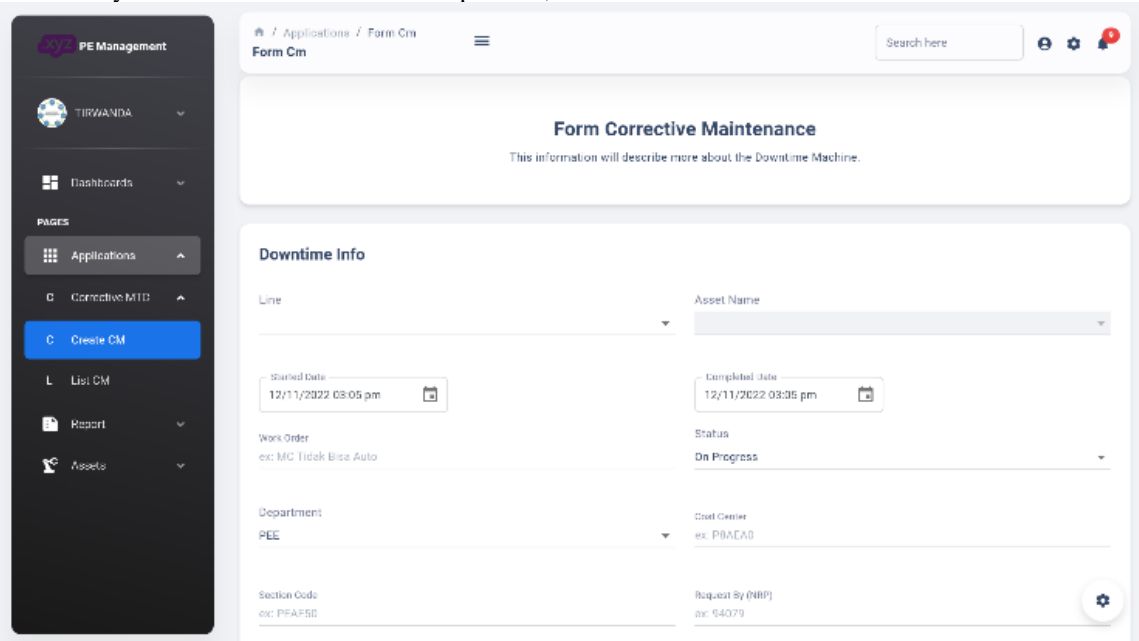


Fig. 14 Create CM Page Downtime Info Section

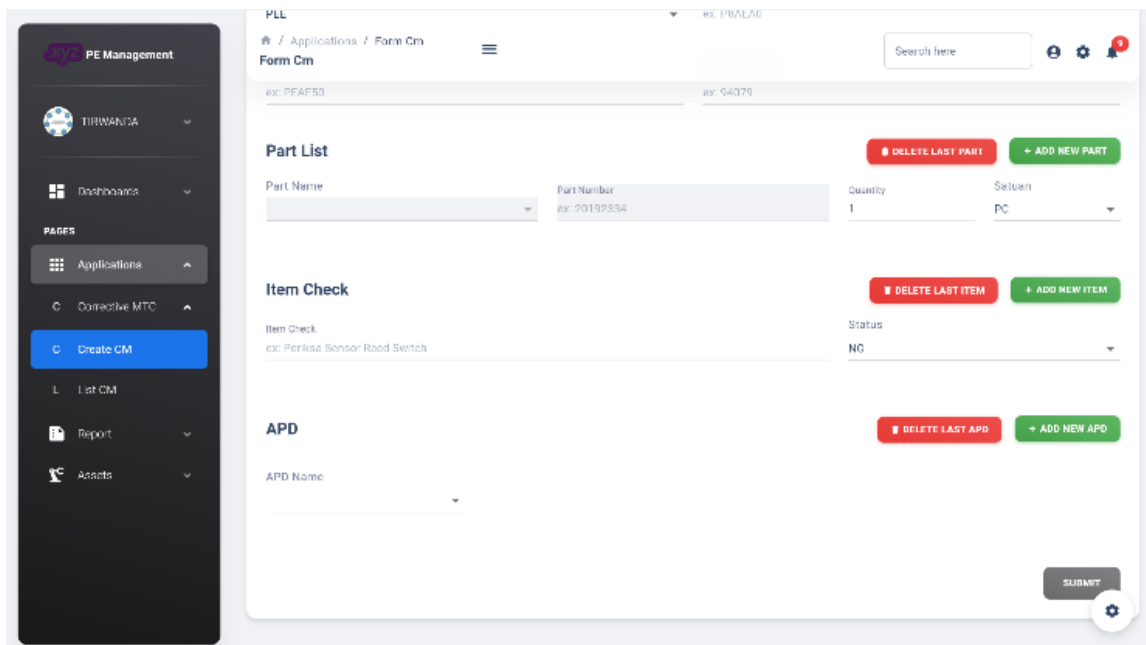


Fig. 15 Create CM Page List Part Section, Item Check Section and APD Section

The "List Corrective Maintenance" page is shown in Figure 16. The "List Corrective Maintenance" page displays a table containing all corrective maintenance created. This table displays corrective maintenance data, including asset

name, line name, date, minutes, status, approval and action, which has three buttons, namely the edit, view and delete buttons.

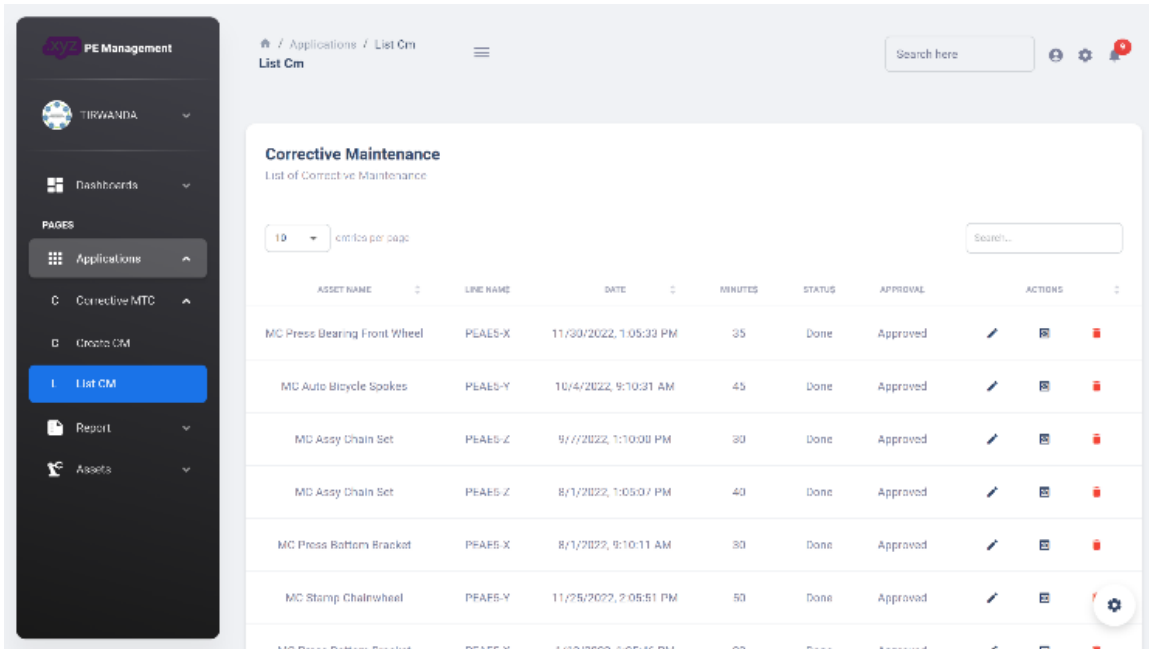


Fig. 16 List Corrective Maintenance Page

On each corrective maintenance, there are three buttons: edit, view, and delete. The edit button functions to change the downtime information that occurs on the production machine; when the user clicks the edit button, the user will be directed to the corrective maintenance edit page (Figure 17 and Figure 18). The view button functions to view

detailed information from the corrective maintenance; if the user clicks the view button, the user will be directed to the corrective maintenance information detail page (Figure 19 and Figure 20). Button delete serves to delete corrective maintenance.

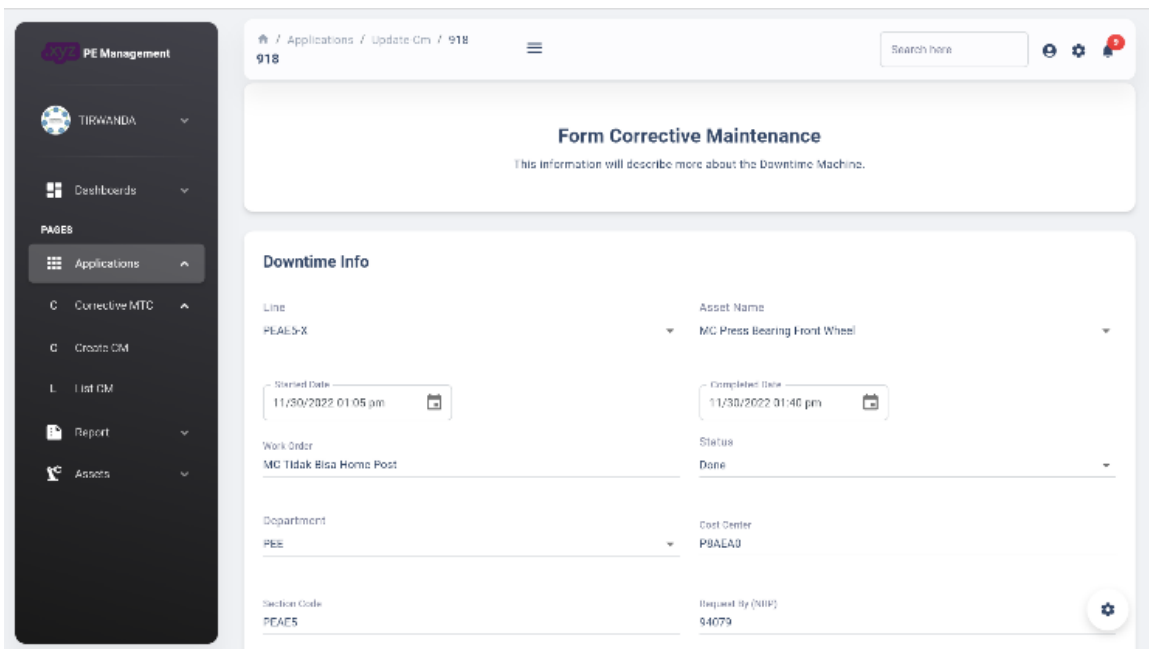


Fig. 17 Edit CM Page Downtime Info Section

The "Edit Corrective Maintenance" page is shown in Figure 17 and Figure 18. On the "Edit Corrective Maintenance" page displays a corrective maintenance form

that has been filled with previous downtime information; in this form, the user is asked to change information related to downtime that occurs, including line, asset name, started

date, completed date, work order, status, department, cost center, section code, request by, part list if any parts are replaced, item check and apd used when carrying out corrective maintenance. If the changed information is

appropriate, the user is asked to click the update button at the bottom of the form to save the changes to the corrective maintenance form.

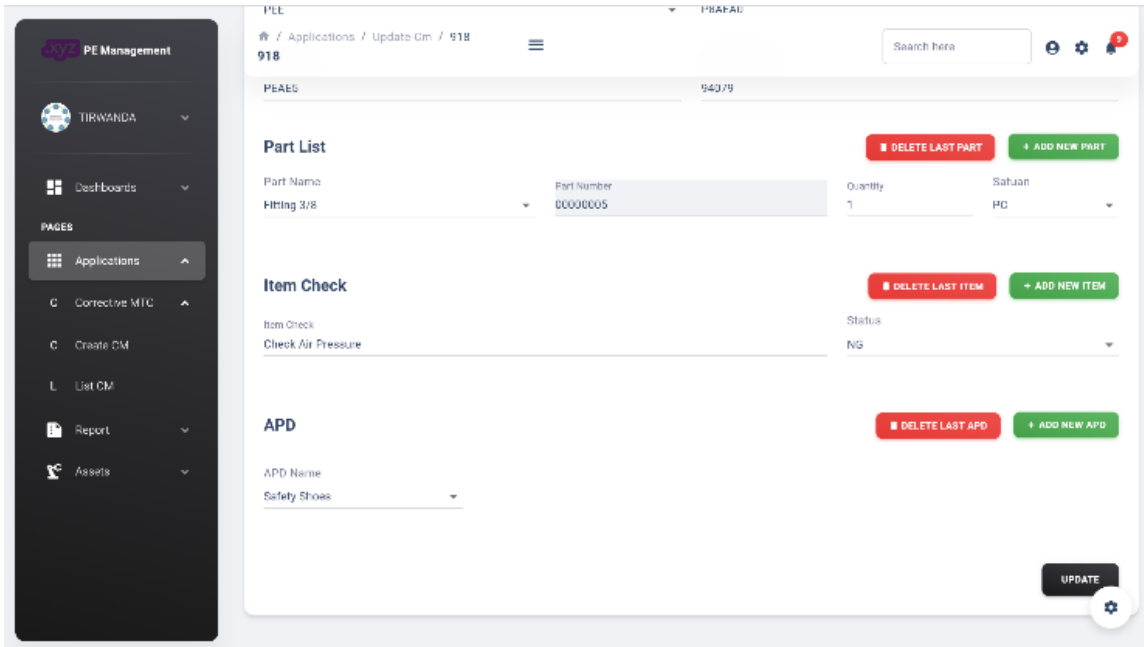


Fig. 18 Edit CM Page List Part Section, Item Check Section and APD Section

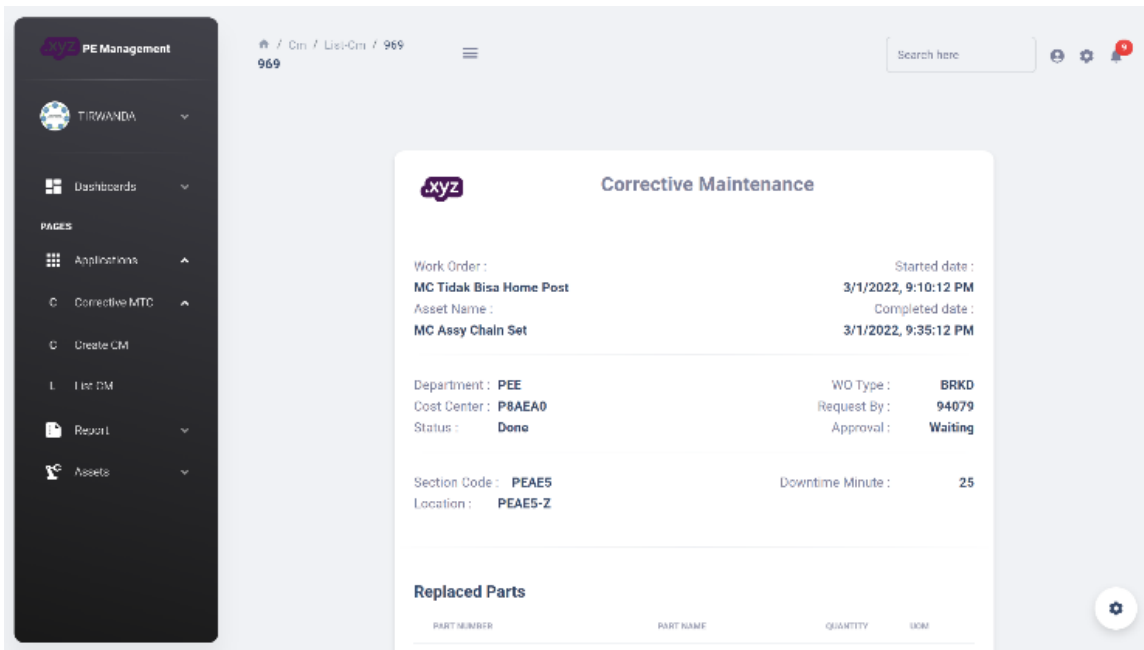


Fig. 19 View Corrective Maintenance Page Downtime Info Section

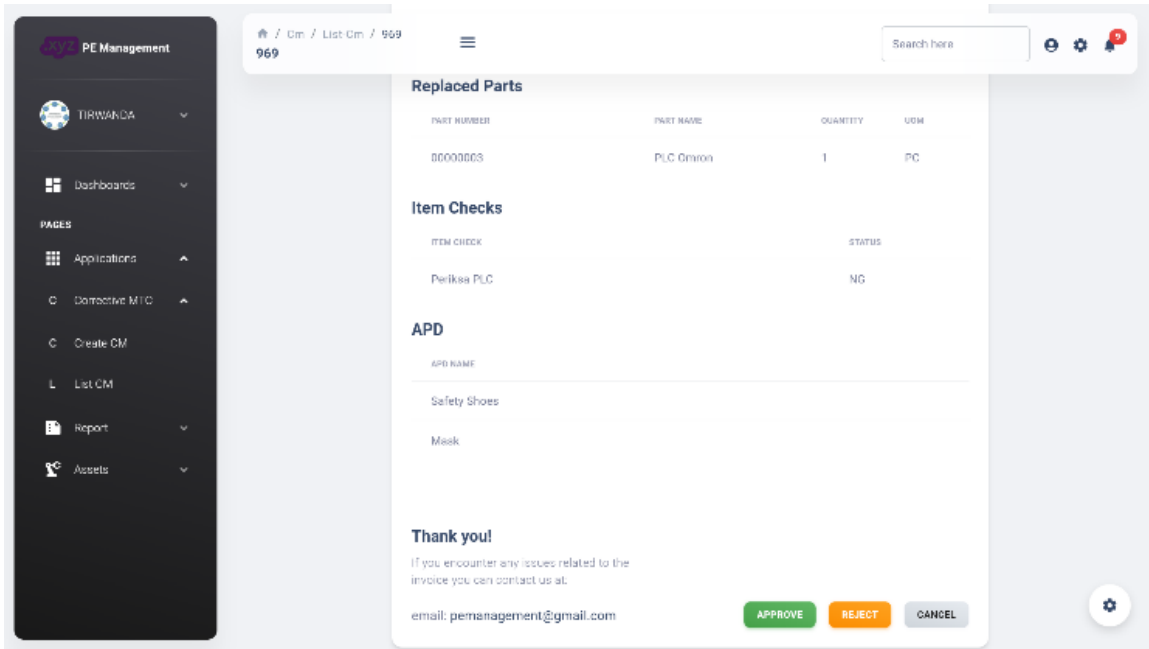


Fig. 20 View Corrective Maintenance Page Replaced part Section, Item Check Section and APD Section

The "View Corrective Downtime" page is shown in Figure 19 and Figure 20. On this page displays detailed corrective maintenance information, and at the bottom, there are three buttons, namely the approve button, eject button and cancel button. The approve button functions to change the corrective maintenance status from "Waiting" to "Approved", the reject button functions to change the corrective maintenance status from "Waiting" to "Rejected", and the "Cancel" button functions to return to the "List Corrective Maintenance" page.

The "MTTR-MTBF" page is shown in Figure 21. On the "MTTR-MTBF" page displays machine performance reports in the form of MTTR (Mean Time to Repair) and MTBF (Mean Time Before Failure), which are presented every month on each production line at the top of the "MTTR-MTBF" page there are two menus, the month selection menu and the year selection, this menu serves to determine the time period of the machine performance report that will be displayed on the "MTTR-MTBF" page.

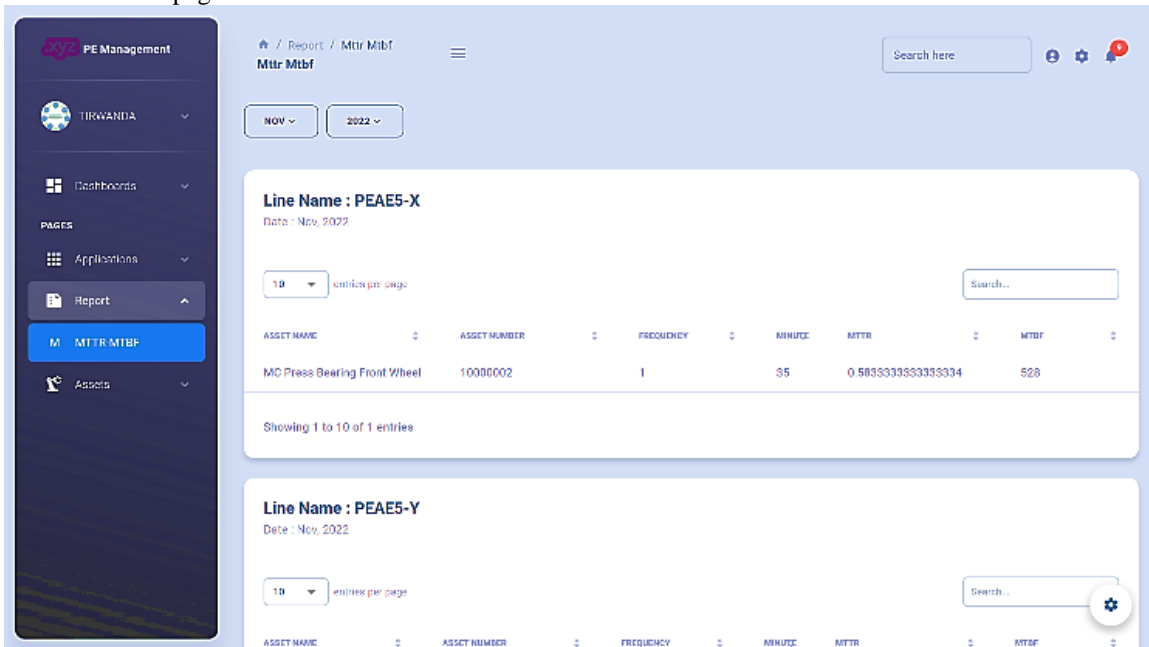


Fig. 21 MTTR and MTBF page

Table 2. Testing Result

| No | Feature | Scenario | Expected Result | Status |
|----|------------------------------------|--------------------------------------------------------------------------------|--------------------------------------------------------|--------|
| 1 | Login | Enter the registered username and password | Login success and display the dashboard page | Pass |
| 2 | Add Line | Complete the form to create a new line | Successfully created a new line | Pass |
| 3 | Add Asset | Complete the form to create a new asset | Successfully created a new asset | Pass |
| 4 | Edit Asset | Change asset detail | Asset updated successfully | Pass |
| 5 | Delete Asset | Delete data from the asset list | Asset deleted successfully | Pass |
| 6 | Create part | Complete the form to create a new part | Successfully add a new part to the list | Pass |
| 7 | Edit Part | Change part detail | Part updated successfully | Pass |
| 8 | Delete Part | Delete data from the part list in asset detail | Successfully delete a part from the list | Pass |
| 9 | Add Part | Add part that has been registered in the database | Successfully add a part to the list | Pass |
| 10 | Add Corrective Maintenance | Complete the form to create a new corrective maintenance | Successfully created a new corrective maintenance | Pass |
| 11 | View Detail Corrective Maintenance | Click the view button on the list row to display corrective maintenance detail | Display data on the corrective maintenance detail page | Pass |
| 12 | Approve Corrective Maintenance | Corrective maintenance validation. Click the approve button | Corrective maintenance status approved | Pass |
| 13 | Reject Corrective Maintenance | Corrective maintenance validation. Click the reject button | Corrective maintenance status rejected | Pass |
| 14 | Edit Corrective Maintenance | Change corrective maintenance detail | Corrective maintenance was updated successfully | Pass |
| 15 | Delete Corrective Maintenance | Delete data from the corrective maintenance list | Corrective maintenance was deleted successfully | Pass |
| 16 | Report | Select the month and year of machine downtime data displayed on the report | Display machine downtime report in tabular form | Pass |
| 17 | Logout | Logout from the application | Logout success and display the login page | Pass |

Table 3. Machine Downtime Comparison

| | Without System | With System |
|-----------|----------------|---------------|
| Repair | 95 | 95 |
| Calling | 5 | 2.5 |
| Find Part | 7.2 | 5.18 |
| Total | 107.2 minute | 102.68 minute |

4.4. Testing

Acceptance testing aims to evaluate the system's compliance with the requirements and assess whether it is acceptable for delivery [14]. Testing is carried out on all features before users can send and use the application. The test case results are shown in Table 2.

4.5. Evaluation

The system trial was carried out at the company in one month, and the results compared the time needed to handle machine downtime reports. Table 3 shows the downtime data during the last month, which compares the time between manual processes and processes using the system.

After the company tested the system, interviews were conducted to get feedback on the applications that had been built. From the interview, the engineering division stated that the system has features that suit user needs, namely machine downtime reporting and asset management. The system also reduces machine downtime data reporting errors because it is done automatically, not with dual input. According to the production division, the process for submitting corrective maintenance is faster because it is unnecessary to deliver files to the engineering division, and it can reduce paper

usage. From the manager's point of view, the dashboard monitoring function is sufficient as evaluation material. In addition, the approval process for corrective maintenance and access to downtime reports can be carried out in real-time.

5. Conclusion

Based on the results of the analysis, design, and implementation of the Centralized Asset Maintenance System and Website-Based Machine Downtime Reporting at PT XYZ Manufacturing, it can be concluded that the system was able to reduce the time of the corrective maintenance submission process by 50% (based on a 1-month trial), reduce the spare parts search process time by 28% (based on a 1-month trial), and automate machine downtime reporting. The system was implemented to address the challenges faced by the company in the previous manual asset management process, downtime reporting, and corrective maintenance submission process, which resulted in a high downtime of 377 minutes (consisting of 19% corrective maintenance

submissions, 6% spare part search, and 75% machine repair) in the last 6 months. With the successful implementation of the Centralized Asset Maintenance System and Website-Based Machine Downtime Reporting, PT XYZ Manufacturing is expected to improve its asset management and maintenance processes significantly.

Additionally, it is suggested that future research focus on exploring the potential of using artificial intelligence and machine learning to enhance the centralized asset maintenance system and downtime reporting process. This could include developing algorithms predicting when machines may experience downtime and automatically triggering corrective maintenance processes to reduce downtime and improve asset performance. Future research could also consider integrating this system with other enterprise systems, such as the enterprise resource planning system, to create a comprehensive and integrated asset management solution.

References

- [1] Nur Fadilah Fatma, Henri Ponda, and Rizky Aditya Kuswara, "Analisis Preventive Maintenance Dengan Metode Menghitung Mean Time Between Failure (MTBF) dan Mean Time To Repair (MTTR) (studi kasus pt. gajah tunggal TBK)," *Heuristic*, vol. 17, no. 2, pp. 87–94, 2020. [[CrossRef](#)] [[Publisher Link](#)]
- [2] Ernie Ilyani Basri et al., "Preventive Maintenance (PM) planning: A Review," *Journal of Quality in Maintenance Engineering*, vol. 23, no. 2, pp. 114–143, 2017. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Temesgen Hailegiorgis Abebe, "Decreasing Machine Down Time by Enhancing Existing Preventive Maintenance System," *SSRG International Journal of Mechanical Engineering*, vol. 7, no. 6, pp. 33–40, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [4] M. Gopalakrishnan, Y. Vinay Kumar, and Arun Kumar Sangaiah, "A Method of Constraint-Based Tutor for Object-Oriented Analysis and Design Curriculum," *SSRG International Journal of Computer Science and Engineering*, vol. 1, no. 9, pp. 1–5, 2014. [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Adi Yermia Tobe, Denny Widhiyanuriyawan, and Lilis Yuliaty, "The Integration of Overall Equipment Effectiveness (Oee) Method and Lean Manufacturing Concept to Improve Production Performance (Case Study: Fertilizer Producer)," *Journal of Engineering and Management in Industrial System*, vol. 5, no. 2, pp. 102–108, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [6] Darian Daji et al., "Cloud-Based Asset Monitoring and Predictive Maintenance in an Industrial IoT System," *International Conference for Emerging Technology*, pp. 1–5, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Adalberto Polenghi et al., "Information as a Key Dimension to Develop Industrial Asset Management in Manufacturing," *Journal of Quality in Maintenance Engineering*, vol. 28, no. 3, pp. 567–583, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Ezikwa Tenas God'swill, and Maxwell Ibe Leo, "A Web-Based Computerized System for Effective Baby Gender Validation," *SSRG International Journal of Computer Science and Engineering*, vol. 9, no. 6, pp. 1–9, 2022. [[CrossRef](#)] [[Publisher Link](#)]
- [9] Neha Dwivedi, Devesh Katiyar, and Gaurav Goel, "A Comparative Study of Various Software Development Life Cycle (SDLC) Models Neha," *International Journal of Research in Engineering, Science and Management*, vol. 5, no. 3, pp. 141–144, 2022. [[Google Scholar](#)] [[Publisher Link](#)]
- [10] Sarang Shaikh, and Sindhu Abro, "Comparison of Traditional and Agile Software Development Methodology: A Short Survey," *International Journal of Software Engineering and Computer Systems*, vol. 5, no. 2, pp. 1–14, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [11] Madhup Kumar, and Ekbal Rashid, “An Efficient Software Development Life cycle Model for Developing Software Project,” *International Journal of Education and Management Engineering*, vol. 8, no. 6, pp. 59–68, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] N.Suresh, “An Overview of Object Oriented Software Testability,” *SSRG International Journal of Computer Science and Engineering*, vol. 1, no. 1, pp. 1-5, 2014. [[Publisher Link](#)]
- [13] Ahmad Fauzi, and Ika Wahyu Pradipta, “Research Methods and Data Analysis Techniques in Education Articles Published by Indonesian Biology Educational Journals,” *Jurnal Pendidikan Biologi Indonesia*, vol. 4, no. 2, pp. 123–134, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Hamed Taherdoost, “Data Collection Methods and Tools for Research ; A Step-by-Step Guide to Choose Data Collection Technique for Academic and Business Research Projects,” *International Journal of Academic Research in Management*, vol. 10, no. 1, pp. 10–38, 2021. [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Irfan Ahmad Khan, and Dipti Kumari, “The Role of Analysis Phase of SDLC for Small Scale Business Application-A Review,” *International Journal of Humanities, Engineering, Science and Management*, vol. 2, no. 1, pp. 63-75, 2021. [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Kalalali Roseline et al., “Mechanism for Detection of Software Design Defects,” *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 3, pp. 12-21, 2020. [[CrossRef](#)] [[Publisher Link](#)]
- [17] I. Friska Heriyanti, and Aulia Ishak, “Design of Logistics Information System in the Finished Product Warehouse with the Waterfall Method: Review Literature,” *IOP Conference Series: Materials Science and Engineering*, vol. 801, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Nirali Honest, “Role of Testing in Software Development Life Cycle,” *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 886–889, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] Woubshet Behutiye et al., “Towards Optimal Quality Requirement Documentation in Agile Software Development: A Multiple Case Study,” *Journal of Systems and Software*, vol. 183, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Minimol Anil Job, and S. Mustafa Alseba, “A Software Engineering Approach in Design and Development of a Mobile Application,” *International Journal of Engineering Trends and Technology*, vol. 69, no. 5, pp. 32-40, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Tri Astoto Kurniawan, “Use Case Modeling (UML): Evaluation of Multiple Errors in Practice,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 1, p. 77, 2018. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Rukiah Nasution, and Aninda Muliani, “Web-Based Inventory Data Processing Information System at the Regional Development Planning Agency (Bappeda) North Sumatra Province,” *Journal of Information Systems and Technology Research*, vol. 1, no. 1, pp. 32–41, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [23] Sikender Mohsienuddin Mohammad, “DevOps Automation Advances I.T. Sectors with the Strategy of Release Management,” *International Journal of Computer Trends and Technology*, vol. 67, no. 12, pp. 82-88, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [24] Willyam Chrisna Umbu Dagha, “Web Event, Spring Boot, Java Pembangunan Aplikasi Web Event Menggunakan Framework Spring Boot di PT XYZ,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 3, pp. 1457–1469, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [25] S.Kannan, and T.Pushparaj, “A Review: Software Security Testing,” *SSRG International Journal of Computer Science and Engineering*, vol. 4, no. 9, pp. 1-8, 2017. [[CrossRef](#)] [[Publisher Link](#)]
- [26] Sabah Al-Fedaghi, “Classes in Object-Oriented Modeling (UML): Further Understanding and Abstraction,” *International Journal of Computer Science and Network Security*, vol. 21, no. 5, 2021. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [27] Irfan Darmawan et al., “Design Thinking Approach for User Interface Design and User Experience on Campus Academic Information Systems,” *International Journal on Informatics Visualization*, vol. 6, no. 2, pp. 327–334, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [28] Jeff Johnson, *Designing with the Mind In Mind: Simple Guide To Understanding User Interface Design Guidelines*, 3rd Edition, ScienceDirect, 2020. [[Google Scholar](#)] [[Publisher Link](#)]