



## Developing A Web Based OEE Monitoring System for Pharmaceutical Industry

Hery Hamdi Azwir<sup>1</sup>, Kevin Desmalilo<sup>2</sup>, Herwan Yasmira<sup>3</sup>

<sup>1,2,3</sup> Faculty of Engineering, Industrial Engineering Department, President University  
Jl. Ki Hajar Dewantara

Kota Jababeka, Cikarang, Bekasi - Indonesia 17550

Email: <sup>1</sup>[hery.azwir@president.ac.id](mailto:hery.azwir@president.ac.id), <sup>2</sup>[kevin.desmalilo@student.president.ac.id](mailto:kevin.desmalilo@student.president.ac.id),  
<sup>3</sup>[herwan.yasmira@president.ac.id](mailto:herwan.yasmira@president.ac.id)

### ABSTRACT

PT. XYZ is a leading biopharmaceutical company in Indonesia that produces and sells health products. A critical aspect of its operations is calculating Overall Equipment Effectiveness (OEE) to ensure optimal production and minimize downtime. Currently, XYZ conducts OEE calculations manually, leading to significant challenges. Manual data collection and processing are time-consuming, taking up to a week to complete. This delay hinders timely decision-making and increases the risk of errors, including miscalculations and inaccuracies in data analysis. Such inefficiencies negatively impact production planning and resource utilization. To address this problem, this research proposes the development of a web-based application to automate the OEE calculation process. Using the Object-Oriented System Design (OOSD) approach and Object-Oriented Programming (OOP) methodology, the application streamlines OEE data input, automates the calculation of Availability, Performance, and Quality components, and presents results through interactive visualizations. By transitioning from a manual to an automated system, the new application significantly reduces the time required for OEE calculations, providing users with fast and accurate results to support better operational decisions.

**Keywords:** Overall Equipment Effectiveness (OEE), Pharmaceutical Industry, Web-Based Application, Production Efficiency, Object-Oriented Programming, Object Oriented System Design.

### ABSTRAK

*PT. XYZ adalah perusahaan biofarmasi terkemuka di Indonesia yang memproduksi dan menjual produk kesehatan. Salah satu aspek krusial dalam operasinya adalah perhitungan Overall Equipment Effectiveness (OEE) untuk memastikan produksi berjalan optimal dan meminimalkan waktu henti. Saat ini, PT. XYZ melakukan perhitungan OEE secara manual, yang menyebabkan berbagai tantangan. Pengumpulan dan pemrosesan data secara manual memakan waktu lama, hingga satu minggu untuk diselesaikan. Keterlambatan ini menghambat pengambilan keputusan yang tepat waktu dan meningkatkan risiko kesalahan, termasuk kesalahan perhitungan dan ketidakakuratan dalam analisis data. Ketidakefisienan ini berdampak negatif pada perencanaan produksi dan pemanfaatan sumber daya. Untuk mengatasi masalah ini, penelitian ini mengusulkan pengembangan aplikasi berbasis web untuk mengotomatiskan proses perhitungan OEE. Dengan menggunakan pendekatan Object-Oriented System Design (OOSD) dan metode Object-Oriented Programming (OOP), aplikasi ini akan menyederhanakan proses input data OEE, mengotomatiskan perhitungan komponen Availability, Performance, dan Quality, serta menampilkan hasil melalui visualisasi yang interaktif. Dengan beralih dari sistem manual ke sistem otomatis, aplikasi ini secara signifikan mengurangi waktu yang diperlukan untuk menghitung OEE, memberikan hasil yang cepat dan akurat guna mendukung pengambilan keputusan operasional yang lebih baik.*

**Kata Kunci:** Overall Equipment Effectiveness (OEE), Industri Farmasi, Aplikasi Berbasis Web, Efisiensi Produksi, Object-Oriented Programming, Object-Oriented System Design.

## 1. Introduction

Every company requires an effective system to measure the efficiency of its equipment, production output, and machine performance. The Overall Equipment Effectiveness (OEE) system is crucial for ensuring optimal production and minimizing downtime. Information systems play an essential role for both large and small companies in supporting various processes such as production monitoring, machine performance tracking, and identifying issues faced by machines. [12]

PT. XYZ is a leading biopharmaceutical company in Indonesia that produces health products categorized into two types: Drug Products and Substance Products. The production process at PT. XYZ is complex due to the nature of the health-related products, requiring strict sterility and precision at each stage. Despite the company's sophisticated systems, some equipment still relies heavily on manual processes. Currently, PT. XYZ calculates its OEE manually. Data from production machines is collected directly by operators or staff, and calculations are performed using tools such as spreadsheets or calculators to determine the Availability, Performance, and Quality components of OEE. This manual method poses several disadvantages, including a high risk of errors, such as data entry mistakes, miscalculations, and negligence in recording downtime. Consequently, the resulting data is often inaccurate and difficult to analyze promptly. Furthermore, manual processes are time-consuming, delaying critical analysis and decision-making. The limited data visibility further hampers the company's ability to respond swiftly and effectively to production issues. [12]

Manual calculation of OEE presents numerous challenges, such as increased risk of human error in data entry and calculations, delayed data processing causing slower decision-making, and poor data visualization, making trend analysis and pattern identification difficult. Several alternative solutions exist to improve OEE calculations. For example, Excel-based automated templates can assist in calculating OEE metrics. However, such templates still require manual data entry, which retains the risk of human error. Additionally, commercial OEE software solutions provide advanced features but may be costly, complex to integrate, and require extensive user training, for example: Evocon (<https://evocon.com/oeo-software/>) and Vorne (<https://www.vorne.com/solutions/applications/oeo-software/>). [3] Developing customized information systems tailored to specific organizational needs is often more effective and ensures better adaptability [12]. Customized systems allow companies to integrate their unique production processes and workflows seamlessly [11]. Furthermore, leveraging Object-Oriented Programming (OOP) principles ensures better system scalability and maintainability. [2]

The development of an automated OEE system is essential to overcome these limitations. By adopting Object-Oriented System Design (OOSD) and Object-Oriented Programming (OOP), the proposed system can efficiently manage data processing, ensuring greater flexibility for future feature enhancements [8]. Additionally, integrating real-time data collection features with database systems such as MySQL enhances data accuracy and speeds up calculation processes [10] Incorporating web development technologies like HTML and CSS ensures a user-friendly interface [5] while frameworks like Flask or Laravel provide robust backend support for data handling and security [9]

Furthermore, implementing the system with interactive dashboards powered by JavaScript libraries will improve data visualization and enable easier interpretation of OEE metrics [7]. By utilizing visual tools, PT. XYZ will gain comprehensive insights into machine performance trends, bottlenecks, and potential inefficiencies. This capability aligns with [6] who emphasize the importance of visual analytics in improving organizational decision-making processes.

The primary objectives of this study are to develop an automated web-based application for OEE calculations, integrate machine data collection for real-time processing, minimize calculation errors through automated computations, and enhance data visualization to improve decision-making processes. The proposed system is expected to reduce the time required for calculations, provide real-time data insights, and support better production planning and resource utilization. By automating the OEE calculation process, PT. XYZ can achieve improved production efficiency, reduced downtime, and enhanced decision-making capabilities. This comprehensive system is designed to address the shortcomings of manual processes, ensuring accuracy, efficiency, and better data management, thus supporting PT. XYZ in maintaining its leading position in the pharmaceutical industry.

## 2. Methods

This section describes the techniques and flow stages of the research completion process. The process flow in this chapter will be used as a guide for doing research to ensure that the study objectives are met. This adjustment reflects a more focused approach on analyzing data to identify issues. The following figure 1 shows methodology structure.

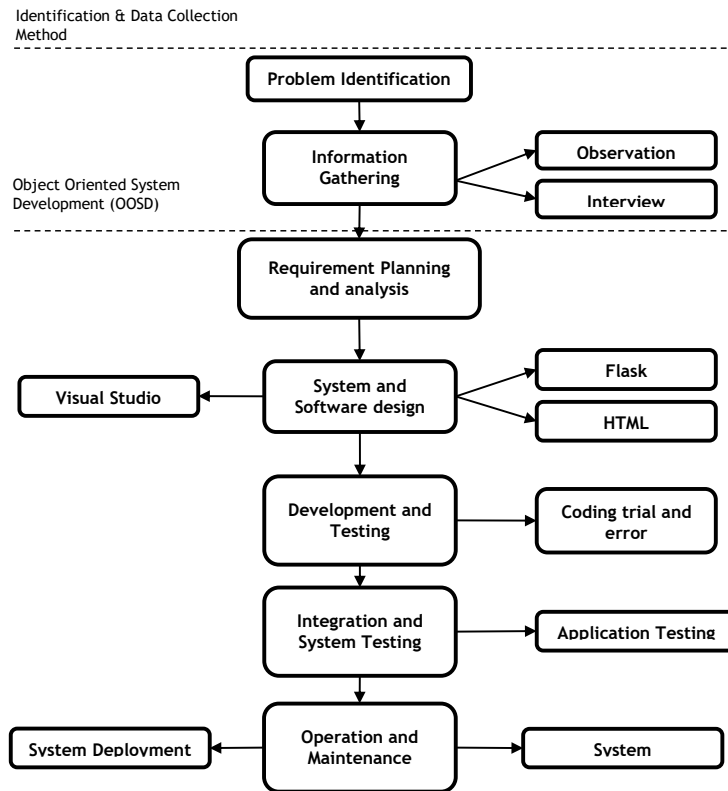


Figure 1. Methodology Structure

## 2.1 Requirement Planning and Analysis

Requirement planning and analysis play a crucial role in the successful development of any system by establishing a solid foundation for the entire project. This phase is essential because it bridges the gap between the initial idea and the technical implementation, ensuring that the developed system meets user expectations and business objectives. The process begins by gathering information from various stakeholders, including end-users, project managers, developers, and clients. This data collection phase may involve conducting interviews, distributing questionnaires, or organizing brainstorming sessions to understand the organization's current challenges, objectives, and expectations.

Once the data is collected, it is carefully analyzed to identify key requirements. Functional requirements describe the specific behaviors and capabilities the system must have to fulfill user needs. Examples include user authentication, data input, data processing, and report generation. These requirements outline what the system should accomplish to deliver its intended functionality. Non-functional requirements, on the other hand, define system attributes such as performance, reliability, scalability, security, and maintainability. For instance, a web-based system may require response times below two seconds to ensure optimal user experience, or the system may need to handle simultaneous requests from hundreds of users during peak hours.

## 2.2 System and Software Design

The design phase is a critical stage in the software development lifecycle, where the conceptual framework of the system is translated into detailed architectural blueprints. In this phase, Object Oriented System Design (OOSD) principles are applied to ensure that the system's structure is modular, scalable, and maintainable. OOSD emphasizes core concepts such as encapsulation, inheritance, polymorphism, and abstraction, each of which contributes to the efficiency and flexibility of the system. Encapsulation ensures that each object manages its own data and behavior, reducing dependencies between components and improving security by restricting direct access to internal data. Inheritance allows developers to build new classes based on existing ones, promoting code reuse and simplifying updates, while polymorphism enables objects to behave differently depending on their context, enhancing system adaptability. Abstraction, on the other hand, helps designers

focus on essential system features while concealing unnecessary complexities, ensuring a cleaner and more organized codebase.

In addition to focusing on the structural aspects, OOSD also helps in organizing the system's data flow, determining how data will be passed between different objects or modules. By utilizing UML diagrams such as Use Case Diagrams, Class Diagrams, Sequence Diagrams, and designers can visualize the interactions between objects and identify potential bottlenecks in the system's design. This ensures that critical aspects like performance and data integrity are taken into account from the outset. Additionally, adopting the MVC (Model-View-Controller) framework enhances separation of concerns, promoting organized development and simplified code maintenance. The following figure 2 shows the OOSD phase.

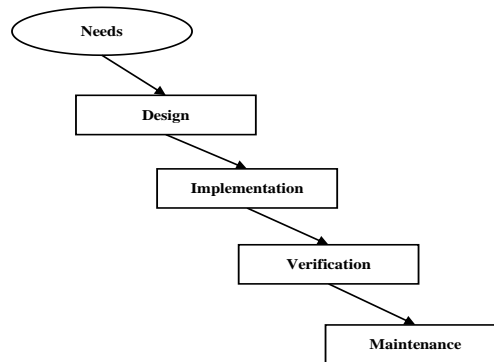


Figure 2. OOSD phase

### 2.3 Coding and Trial Error

After writing the code in Visual Studio Code, the implementation phase involves rigorous testing using a trial-and-error approach. This method is essential for identifying and resolving issues that may arise during development. Trial and error allow developers to experiment with different solutions, identify potential bugs, and refine the code until the desired functionality and performance are achieved. The process is iterative, meaning developers continuously test, analyze results, and improve their code through multiple cycles.

### 2.4 Integration and System Testing

Integration and system testing are crucial stages in the software development lifecycle, designed to validate the overall functionality and performance of the system as a complete unit. After individual components have been developed and tested independently, integration testing ensures that these components work harmoniously when combined. This phase is essential because even well-functioning modules may encounter issues when interacting with one another, especially when data flows across different layers such as frontend, backend, and database systems.

Database testing is a vital part of system testing, particularly for web applications that rely heavily on data storage and retrieval. This involves validating data integrity, ensuring database queries are optimized, and confirming that transactions are processed correctly. For example, testing may involve verifying that records are inserted, updated, and deleted as intended, with proper rollback mechanisms in place to handle failed transactions. Developers also test data migrations, ensuring seamless data transfer when updating database schemas or deploying new features. Integration and system testing involve combining all the components and modules developed during the implementation phase and testing them as a whole to ensure they work together seamlessly. The following figure 3 shows database management system.

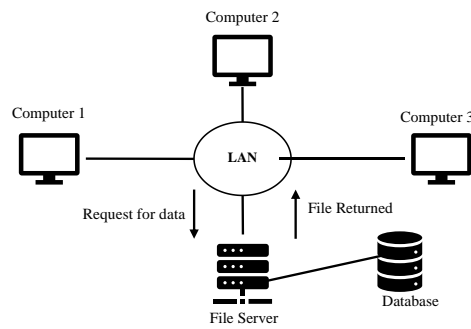


Figure 3. Database Management System

## 2.5 Application Testing

Application testing is a critical phase in the software development lifecycle, aimed at ensuring that the application functions as intended, meets user requirements, and performs efficiently under various conditions. This process involves multiple testing strategies designed to identify defects, improve functionality, and enhance the overall user experience. By thoroughly testing the application, developers can minimize the risk of errors in the production environment and ensure the system is stable, secure, and reliable.

## 2.6 Operation Maintenance

After the system is deployed, it enters the operation and maintenance phase. This phase involves monitoring the system's performance, providing support to users, and making necessary updates to ensure the system continues to meet the organization's needs.

## 2.7 System Deployment

System deployment during the maintenance phase includes adding new features, optimizing existing ones, and ensuring the system remains compatible with new technologies and changing requirements. The System Deployment Phase is a critical stage that involves the transition of the developed system from the testing environment to the live production environment. This phase not only focuses on the initial launch but also encompasses ongoing enhancements, updates, and optimizations to ensure the system remains effective and relevant. Deployment activities include finalizing configurations, performing data migration, and ensuring system compatibility with the target environment.

## 3. Result and Discussion

At PT. XYZ, the Overall Equipment Effectiveness (OEE) calculation is a critical part of the production system. This calculation is conducted manually, involving several steps that require human input and observation. The process begins with the collection of data from the production floor, which includes machine uptime, downtime, production output, and quality metrics. These data points are manually recorded by operators or supervisors on a daily basis. The operators gather data such as machine uptime, downtime, and production output. This data is recorded manually and entered into spreadsheets. The calculation involves three key metrics: Availability, Performance, and Quality.

PT. XYZ is exploring the development of a web-based OEE system to automate data collection, calculation, and reporting, aiming to improve accuracy, efficiency, and decision-making capabilities in their production environment. The following figure 4 shows current OEE calculation system.

Theoretical BS	Actual Output	Good Output	Rate of Quality (Good Output / Actual Output)	Std output per mins	Actual Machine Operating Time (mins)	Performance %	Filling Time (mins)	Planned Working Time	Availability	OEE
Jul										
75,000	71,840	65,907	91.74%	75	2966	32.29%	958	1,000	95.79%	67.34%
50,000	48,160	46,957	97.50%	75	701	91.60%	642	667	96.32%	
18,750	17,503	15,288	87.35%	40	555	78.84%	438	469	93.35%	
18,750	17,505	15,251	87.12%	40	603	72.57%	438	469	93.36%	
18,750	17,404	14,747	84.73%	40	562	77.42%	435	469	92.82%	
75,000	72,180	68,608	95.05%	75	812	118.52%	962	1,000	96.24%	
			90.58%			78.54%			94.65%	

ration on MVHMOJ23003 due to Liquid in RS was found since tub 110 caused by unstable vacuum stoppering in the middle of filling process (DR-2024.06.06 There was found liquid

Figure 4. Current OEE calculation system

In the OEE manual system at PT. XYZ, the process involves three main entities, namely Operators, Supervisors, and Managers. The system is run in several stages, and it usually takes anywhere from one to two weeks to complete. The following figure 5 shows the current flow OEE calculation.

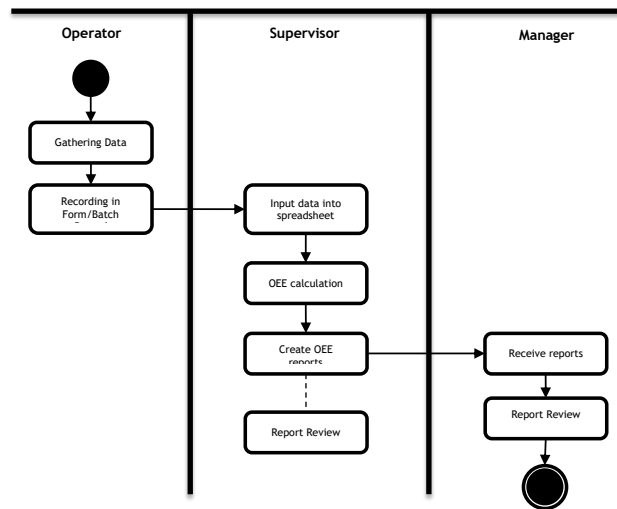


Figure 5. Current flow OEE calculation

In this manual system, it was explained previously in the current flow manual calculation of OEE where the operator must write and record all data on the form listed. Then here is an example of a form that needs to be written and recorded by the operator. The following figure 6 shows OEE manual form.

No Batch	Jam Awal	Jam Selesai	Total Jam	Output (kg)	Target (kg)	% Achievement	% Output	Keterangan	Status	Status Operator	Unit Operator
230001	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230002	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230003	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230004	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230005	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230006	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230007	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230008	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230009	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...
230010	17:30	18:45	1 1/4	1.800	1.800	100%	100%	...	...	...	...

Figure 6. OEE manual form

In the manual inputting form above, it is clear that this form is not efficient because there are a lot of human errors. Unclear writing, unfilled data and incorrect writing. In the form above, it shows human error, the time used is quite long and inefficient. The following is the error data on the form that the operator wrote in a month. The following table 1 shows the error data in a month.

**Table 1.** Error data in a month

Name of Errors	Frequency
Incorrect writing	8
Writing cannot be read	11
Form not filled out	5
Torn form data	3

The table above shows the errors that are often experienced in the OEE calculation tool that are currently occurring. From the data above, it was found that incorrect writing 8, writing cannot be read 11, form not filled out 5 and torn form data is 3 from 30 data. That means that out of the 38 data in the form, there are 27 data that have related issues.

### 3.1 Requirement Planning and Analysis

In this web-based application, we need some requirements Planning and Analysis in OEE Server that include the following steps:

#### 1. Identify System Needs

Identify the features and functionality expected of the system, such as machine management, OEE calculations, shift management, and others.

#### 2. User Needs Analysis

Understand what users (operators, production managers, technicians) need in terms of interfaces, workflows, and access to real-time data.

#### 3. System Architecture Definition

Defines the structure of the database, the required modules, and integration with other components such as Redis, SQL Alchemy, or Flask-Login.

### 3.2 Identify System Needs

The functional requirements of the OEE Server system may include some of system that will be explain on following table. Table 2 shows the System Needs.

**Table 2.** System Needs

System	Explanation
Machine Management	Features to add, edit, and remove (CRUD) machines to be monitored in the system.
OEE calculations	The system must be able to calculate the three components of OEE, namely Availability, Performance, and Quality, and provide a report of the calculations.
User and Shift Management	Handles the data of users involved in operations, as well as their shift schedules.
Real-Time Data Collection	The system needs to collect and store machine performance data in real-time or semi-real-time to analyse efficiency

### 3.3 User Need Analysis

In this web-based application, the priority of this web-based application is to make it easier for users to analyze OEE. Here is a list of user needs:

#### 1. Machine Operator

Enters data related to machine performance, downtime, and production quality.

#### 2. Production Manager or Supervisor

View machine performance reports in real-time or historically, and monitor production efficiency.



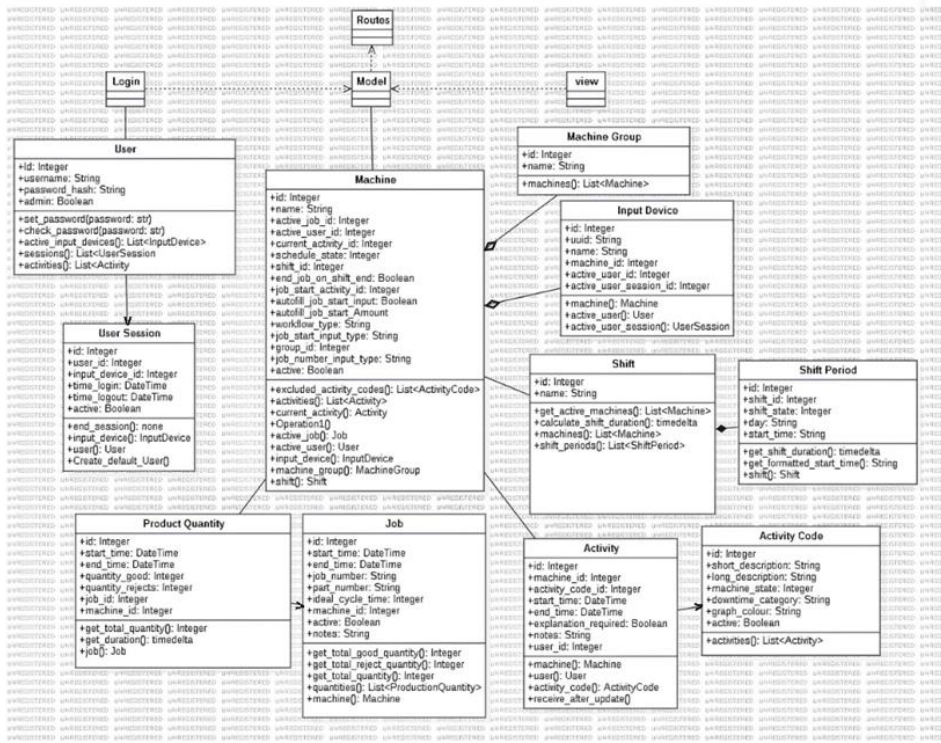


Figure 8. Class Diagram

**Cache (Redis):** Redis can be used to store temporary data and speed up data access, especially if real-time data processing is required.

Database design is depending on the ERD. The Entity-Relationship Diagram (ERD) that outlines the structure of a database, likely related to a production monitoring system. The design of the diagram can be seen in figure 9.

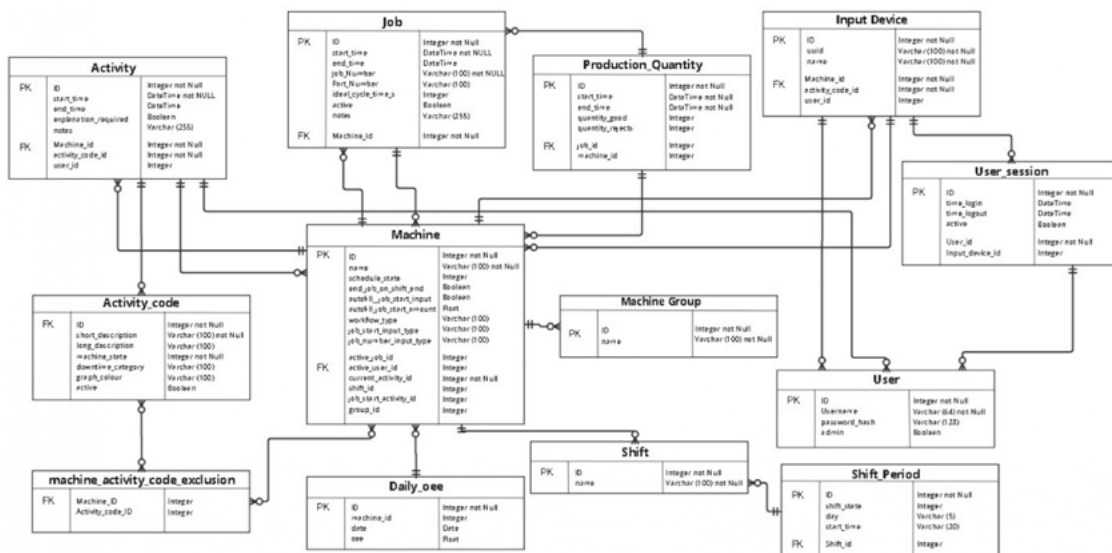


Figure 9. Entity Relationship Diagram

**Frontend:** UI/UX for users, both for operators and managers, that displays OEE metrics and enables interactions such as entering data or viewing reports. On the status display, it shows the status of the machine that is running. These statuses also include availability, performance and Quality.

In this view there are also some additions to clarify the usefulness of this status page, namely the date and also the outline of the machine that will change according to the status that occurs. The

following figure 10 shows the UI Design.

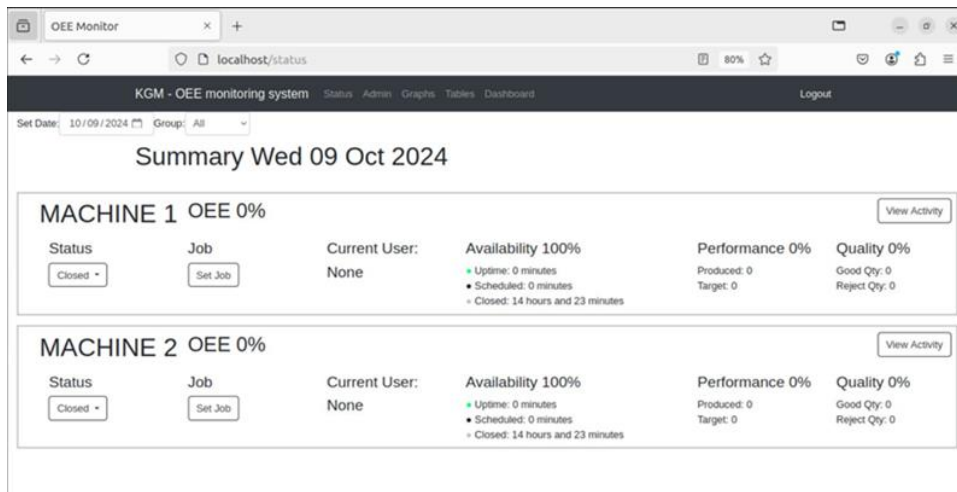


Figure 10. UI Design

### 3.5 Implementation of Program

The program implementation on the OEE monitoring system web application uses a flask framework with HTML and Python programming languages. In this OEE monitoring system web application, it uses ubuntu as the web server for the server development environment. The codes below including routes, models and views is taken from GitHub and DigitME2 then implemented and installed on this website. The following is a code snippet consisting of routes, models, and views.

The routes code is used to create a URL access and connect it to the View. That way, users can access the desired page directly without having to navigate from the home page. Figure 11 shows the route code.

```
94 @bp.route('/machine')
95 @login_required
96 def machine_report():
97     """ Show all a user's activities and allow them to be edited """
98     machine_id = request.args.get("machine_id")
99     machine = Machine.query.get_or_404(machine_id)
100     activity_codes = ActivityCode.query.all()
101     requested_date = request.args.get("date")
102     if requested_date:
103         requested_date = datetime.strptime(requested_date, "%Y-%m-%d").date()
104     else:
105         requested_date = datetime.now().date()
106     period_start = datetime.combine(date=requested_date, time=time(hour=0, minute=0, second=0, microsecond=0))
107     period_end = period_start + timedelta(days=1)
108
```

Figure 11. Route Code

This routes code was imports to bring in necessary modules and helper functions for handling dates, form processing, querying the database, user authentication, and rendering templates. The following figure 12 shows the import code from route code.



```
3 {% block content %}
4 <div class="container-flex mt-2">
5   <div class="card-group">
6
7
8     <div id="adminUserColumn" class="card border-0" style="min-width: 12rem;">
9       <div class="card-header" style="height: 6rem;">
10        <h2 class="adminListHeader">Users</h2>
11        <a id="create_new_user_btn" href="{{ url_for('admin.new_user') }}" class="">+ New User</a>
12      </div>
13      <div class="card-body overflow-auto">
14        <ul class="list-group">
15          {% for user in users %}
16            <li class="list-group-item">
17              <a id="edit_user_{{ user.id }}" href="{{ url_for('admin.change_password', user_id=user.id, action='reset_password') }}">
18                {{ user.username }} (ID: {{ user.id }})
19              </a>
20            </li>
21          {% endfor %}
22        </ul>
23      </div>
24    </div>
25  </div>
26 </div>
```

Figure 15. View Code

In the code snippet above, write `{% .... %}`. `{% extends "default/base.html" %}` is the syntax in the template engine used by Flask to build HTML templates.

#### 4. Conclusion

In conclusion, the current process of calculating Overall Equipment Effectiveness (OEE) at PT. XYZ is still performed manually, involving operators, supervisors, and managers in recording machine data and calculating OEE components such as Availability, Performance, and Quality. While this manual method has been in place for some time, it presents several significant drawbacks. The reliance on human input increases the risk of errors, including data entry mistakes, miscalculations, and negligence in recording downtime. These issues compromise the accuracy of OEE results, making it difficult for the company to obtain reliable insights into production performance. Moreover, the manual method is time-consuming, delaying data analysis and strategic decision-making.

To address these challenges, this study proposes the development of a web-based application that automates the OEE calculation process. The proposed system will be designed using the Object-Oriented System Design (OOSD) approach and implemented with Object-Oriented Programming (OOP). By integrating machine data directly into the system, this solution aims to minimize the need for manual data entry, reducing the potential for human errors. Furthermore, the system will feature essential functionalities such as machine management, user and shift management, and automated report generation.

The system will utilize Flask as the backend framework, SQL Alchemy for database management, and Redis for real-time data caching. This combination of technologies will enable the system to collect data accurately and provide comprehensive insights into machine conditions and production performance. Interactive data visualizations will further enhance the company's ability to monitor performance trends, identify potential issues early, and support faster, data-driven decision-making.

Overall, the implementation of this automated system is expected to significantly improve the accuracy and efficiency of OEE calculations at PT. XYZ. By streamlining data collection and analysis, the system will help the company optimize production planning, reduce machine downtime, and improve resource utilization. Ultimately, this will contribute to increased productivity and enhance PT. XYZ's competitiveness in the pharmaceutical industry, where stringent production standards are critical.

#### References

1. Alter, S. (2013). Defining information systems as work systems: implications for the IS field. *European Journal of Information Systems*, 17(5), 448-469. <https://doi.org/10.1057/ejis.2008.37>
2. Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Connallen, J., & Houston, K. A. (2009). *Object oriented analysis and design with applications 3RD Edition*.
3. Cosmas, N. I. (2018). Transitions in system analysis and design methodology. *American Journal of Information Science and Technology*, 2(2), 50. <https://doi.org/10.11648/j.ajist.20180202.14>
4. D. Heller, "HTML and Web Development," [W3Schools], 2022. [Online]. Available: <https://www.w3schools.com/html/>
5. Duckett, J. (2014). *HTML and CSS: Design and build websites*. In Wiley Publishing eBooks.
6. Laudon, K. C., & Laudon, J. P. (1988). *Management Information Systems: Managing the digital firm*.

7. Learning web design: a beginner's guide to HTML, CSS, Javascript, and web graphics. (2013). Choice Reviews Online, 50(12), 50-6805. <https://doi.org/10.5860/choice.50-6805>
8. Pressman, R., & India, M. G. H. (2009). Software Engineering: A Practitioner's Approach, 7Th Edition.
9. Rankin, K., & Hill, B. M. (2017). The official Ubuntu Server Book.
10. Schwartz, B., Zaitsev, P., & Tkachenko, V. (2012). High performance MySQL.
11. Stevens, N. J., Salmon, P. M., Walker, G. H., & Stanton, N. A. (2018). Systems analysis and design methods. In CRC Press eBooks (pp. 291-332). <https://doi.org/10.1201/9781315587363-10>
12. Turban, E., Volonino, L., Sipior, J. C., & Wood, G. R. (2011). Information Technology for Management: Improving strategic and operational performance. <http://ci.nii.ac.jp/ncid/BB05018219>
13. Stallng, W. (2018). Operating Systems: Internals and Design Principles / William Stallngs. Operating Systems: Internals and Design Principles Seventh Edition, 2001(2001), 1-99.
14. Haselmann, T., & Vossen, G. (2011). Software-as-a-Service in Small and Medium Enterprises: An Empirical Attitude assessment. In Lecture notes in computer science (pp. 43-56). [https://doi.org/10.1007/978-3-642-24434-6\\_4](https://doi.org/10.1007/978-3-642-24434-6_4)
15. Marz, N., & Warren, J. (2015). Big Data: Principles and best practices of scalable realtime data systems.
16. Hacigümüő, H., Iyer, B., Li, C., & Mehrotra, S. (2002). Executing SQL over encrypted data in the database-service-provider model. Executing SQL Over Encrypted Data in the DatabaseServiceProvider Model. <https://doi.org/10.1145/564691.564717>
17. Connolly, T. M., & Begg, C. E. (2015). Database Systems: A Practical approach to design, implementation and management.
18. Elmasri, R. A., & Navathe, S. B. (2016). Fundamentals of database Systems.
19. Sobell, M. G. (2013). A practical guide to Linux commands, editors, and Shell programming.
20. Negus, C. (2005). Linux Bible.