THE UNIVERSITY OF DANANG

UNIVERSITY OF SCIENCE AND TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING

CAPSTONE PROJECT

MAJOR: MECHATRONICS ENGINEERING

DESIGN OF PREDICTIVE MAINTENANCE SYSTEM USING IOT

Supervisor	: **Ph.D NGO THANH NGHI**
Co-Supervisor	: **Mr. Tran Quang Phu**
Approved	: **Ph,D Pham Anh Duc**
Students	: **Nguyen Viet Khanh**
	Nguyen Thanh Phuong
Student ID	: **101200398**
	101200407
Class	: **20CDTCLC3**

Da Nang, 6/2025

SUMMARY

Title: Design of a Predictive Maintenance System Using IoT

Student: Nguyen Viet Khanh Student ID: 101200398

Class: 20CDTCLC3 Industry: Mechatronics Engineering

Student: Nguyen Thanh Phuong Student ID: 101200407

Class: 20CDTCLC3 Industry: Mechatronics Engineering

Supervisor: Ph.D Ngo Thanh Nghi

Approved by: Ph.D Pham Anh Duc

1. Practical necessity: In the context of increasing industrialization, manufacturing enterprises are under growing pressure to enhance operational efficiency and minimize downtime. Traditional maintenance methods are no longer effective in predicting and preventing equipment failures. Therefore, the research and implementation of a predictive maintenance system based on IoT is essential. Such a system enables real-time monitoring of equipment conditions, early detection of potential faults, and optimization of maintenance planning—ultimately reducing costs and improving system reliability.

2. Scope of the study: The research focuses on the design and development of a predictive maintenance system for industrial equipment based on the IoT platform. The scope is limited to collecting sensor data (such as vibration, temperature, etc.) from machinery, processing and analyzing this data using machine learning algorithms to detect early signs of malfunction. The system is implemented in a simulated or small-scale real-world environment to evaluate its applicability in intelligent industrial maintenance.

3. The content of the topic has been done

Number of drawings: 5

Project Report: 01

4. Results: The project simulates a CMMS (Computerized Maintenance Management System) software to collect sensor data, apply machine learning algorithms for failure prediction, and generate early warnings. The system is also capable of automatically sending email alerts when abnormal conditions are detected.

THE UNIVERSITY OF DANANG
University of Science and Technology
Faculty of Mechanical Engineering

SOCIALIST REPUBLIC OF VIETNAM
Independent – Freedom - Happiness

GRADUATION PROJECT TASK

TT	Student name	Student ID	Class	Major
1	Nguyen Thanh Phuong	101200407	20CDTCLC3	Mechatronics Engineering
2	Nguyen Viet Khanh	101200398	20CDTCLC3	Mechatronics Engineering

1. *Project title: Design of predictive maintenance system using iot*
2. *Project Category:* ☐ *Subject to an Intellectual Property Agreement regarding the results achieved.*
3. *Contents of Explanation and Calculations:*

a. General part:

TT	Student name	Content
1	Nguyen Thanh Phuong	Search for sources, references, illustrations, glossaries; Check spelling, formatting, content and design of annotated model images. Explain, operate and test. Present reports, opening and closing.
2	Nguyen Viet Khanh	

b. Private part:

TT	Student name	Content
1	Nguyen Thanh Phuong	Chapter 1: Overview of the research topic
2	Nguyen Viet Khanh	Chapter 2: Analysis of design options
3	Nguyen Thanh Phuong	Chapter 3: Components of the system
4	Nguyen Viet Khanh	Chapter 4: CMMS Maintenance Software
5	Nguyen Thanh Phuong	Chapter 5: Maintenance software design
6	Nguyen Viet Khanh	Chapter 6: Conclusion

4. *Drawings, graphs (specify types and sizes of drawings):*

a. General part:

TT	Student name	Content
1	Nguyen Thanh Phuong	1 A0 process drawing

2	Nguyen Viet Khanh	1 A0 controller drawing 1 A0 overview drawing

b. Private part:

TT	Student name	Content
1	Nguyen Viet Khanh	1 A0 electrical drawing
2	Nguyen Thanh Phuong	1 A0 Algorithm flowchart drawing

6. Supervisor name:	*Section/ Content:*
Ph.D Ngo Thanh Nghi	Entire topic
Mr. Tran Quang Phu	Entire topic

Da Nang, 6/2025

Supervisor

Dr. Ngo Thanh Nghi

PREFACE

In the process of industrialization and modernization, automation plays a crucial role. Due to industrial automation, factories have become more efficient in utilizing energy, materials, and human resources. Industrial automation involves using management systems such as computers, robots, and information technology to control various machines and production processes in the industry. After mechanization, automation is the second step in industrialization.

In the current economic context, both the world and our country are gradually developing.

After a period of diligent study and research, and under the dedicated supervision of Ph.D Ngo Thanh Nghi, we have successfully completed our graduation thesis titled: *"Design of a Predictive Maintenance System Using IoT".* By applying the knowledge acquired throughout our academic journey, consulting relevant literature, and conducting practical field surveys, we have made our utmost effort to fulfill the objectives of this thesis. We also strived to minimize errors that may arise due to our limited practical experience and knowledge. Therefore, we sincerely welcome and appreciate any constructive feedback and valuable guidance from our instructors to further improve the quality of this work.

In addition, we would like to express our heartfelt gratitude to East Sea Electrical Automation Technology Co., Ltd. (ESTEC) for their support and for providing favorable conditions that enabled us to gain practical insights and collect essential information to complete this project.

Da Nang, 6/2025

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the graduation thesis titled: “Design of Predictive Maintenance System Using IoT” is the result of my own serious learning and research process, carried out independently under the supervision of Ph.D Ngo Thanh Nghi.

I affirm that I have not copied, used, or presented any part of others’ works or documents without proper citation. If any academic dishonesty or violation of integrity regulations is discovered, I take full responsibility before the Faculty Council and the University.

I sincerely thank all lecturers who have guided and supported me during the process of completing this thesis.

Da Nang, 6/2025

Phuong	Khanh
Nguyen Thanh Phuong	Nguyen Viet Khanh

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full Term
PLC	Programmable Logic Controller
CMMS	Computerized Maintenance Management System
MRO	Maintenance, Repair, and Overhaul
WL	Working Layer
API	Application Programming Interface
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Network

CHAPTER 1 INTRODUCTION

1.1 About the company

Information[1]:

ESTEC is a leading solution provider in the field of Automation and Digitalization for various industries in Vietnam. The company offers end-to-end services including engineering design, software programming, system integration, equipment and material supply, construction, installation, commissioning, training, and technology transfer.

BIEN DONG AUTOMATION TECHNOLOGY CO., LTD

+ Head Office: 61 Le Duc Tho Street, Ward 7, Go Vap District, Ho Chi Minh City, Vietnam
+ Factory: Lot A14, Street No. 7, Da Nang Hi-Tech Park, Hoa Lien, Hoa Vang District, Da Nang City, Vietnam
+ Representative Office: 87 To Hieu Street, Nghia Do Ward, Cau Giay District, Hanoi City, Vietnam
+ Hotline: (+84) 28 5446 4649
+ Fax: (+84) 28 5446 4648
+ Email: info@biendongco.vn

Figure 1.1: Estec Company

1.2 General introduction to automation

Automation is one of the key fields driving the development of modern science and technology. It refers to the application of automatic control systems to operate machinery, equipment, and production processes with minimal or no human intervention. Automation has increasingly proven its vital role in various sectors of life and industrial production.

The emergence and advancement of automation have fundamentally transformed traditional production methods, shifting from manual operations to flexible, intelligent, and efficient manufacturing systems. By integrating mechanics, electrical and electronic engineering, automatic control, and information technology, modern automation systems can operate continuously, with high precision, reduced errors, improved product quality, and optimized operational costs.

In the context of the Fourth Industrial Revolution (Industry 4.0), automation goes beyond mere control and monitoring. It is now integrated with advanced technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), and Big Data to create intelligent systems capable of learning, analyzing, and making real-time decisions.

In Vietnam, automation has become a significant trend in major industries such as mechanical manufacturing, electronics, food and beverage, pharmaceuticals, logistics, and energy. Investing in automation technology not only helps enterprises increase productivity and quality but also fosters the development of a highly skilled technical workforce, ready to compete both domestically and internationally.

Given its profound importance, the research and application of automation solutions in production and equipment maintenance is a practical direction that contributes to technical innovation and operational efficiency. It also motivates engineering students to approach new technologies and prepare for integration into the modern industrial landscape.

Figure 1.2: Introduction to Automation[2]

1.3 Introduction to the topic

In modern industrial systems, maintaining the stability and reliability of machinery is a crucial factor in ensuring production efficiency, product quality, and overall competitiveness. Unexpected equipment failures not only result in downtime and costly repairs but also disrupt production schedules and compromise operational safety.

Traditionally, industries have relied on two main maintenance strategies:

- Corrective Maintenance: Repairs are performed only after equipment failure has occurred.
- Preventive Maintenance: Maintenance is scheduled at fixed intervals or based on equipment usage time.

However, both approaches present significant drawbacks. Corrective maintenance can lead to unplanned downtimes and serious operational risks, while preventive maintenance may cause unnecessary resource consumption by replacing parts that are still functional.

To address these limitations, Predictive Maintenance (PdM) has emerged as a modern and efficient solution. PdM involves real-time monitoring of machine conditions using sensors, combined with data analysis techniques to detect anomalies and predict the likelihood of equipment failure within a future time window. The key advantage of PdM lies in its ability to determine the "optimal timing" for maintenance—striking a balance between minimizing risk and maximizing equipment life.

With the advancement of the Internet of Things (IoT), implementing predictive maintenance systems has become more feasible and powerful. IoT technologies—such as smart sensors, microcontrollers, wireless communication, and cloud platforms—enable continuous monitoring of key parameters like vibration, temperature, current, and pressure. These devices can:

- Collect and store operational data in real time;
- Detect abnormal patterns and warning signs;
- Forecast the equipment's health status for a defined period (e.g., the next 3, 5, or 7 days);
- Send early alerts to maintenance personnel or management for timely intervention.

This project, titled “Design of Predictive Maintenance System using IoT,” aims to develop a practical and scalable solution that integrates sensor data acquisition, anomaly detection, and short-term failure prediction for industrial equipment. The system will include:

- IoT sensors to monitor vibration, temperature, and equipment status;
- A microcontroller or industrial PC to collect and transmit data;
- A real-time dashboard with visual charts;
- Threshold-based alerts for abnormal conditions;
- A prediction module capable of forecasting failures within a specific future timeframe;
- Automatic email or interface-based notifications to relevant personnel.
- By implementing this system, the project seeks to:
- Reduce machine downtime and maintenance costs;

- Improve operational efficiency and equipment longevity;
- Provide proactive, data-driven decision support for maintenance planning;
- Support the development of smart factory systems in line with Industry 4.0.
- This project gives us the opportunity to apply interdisciplinary knowledge—from electronics, embedded systems, and signal processing to networking and IoT—and to develop systems aligned with the future of industrial automation and digital transformation.

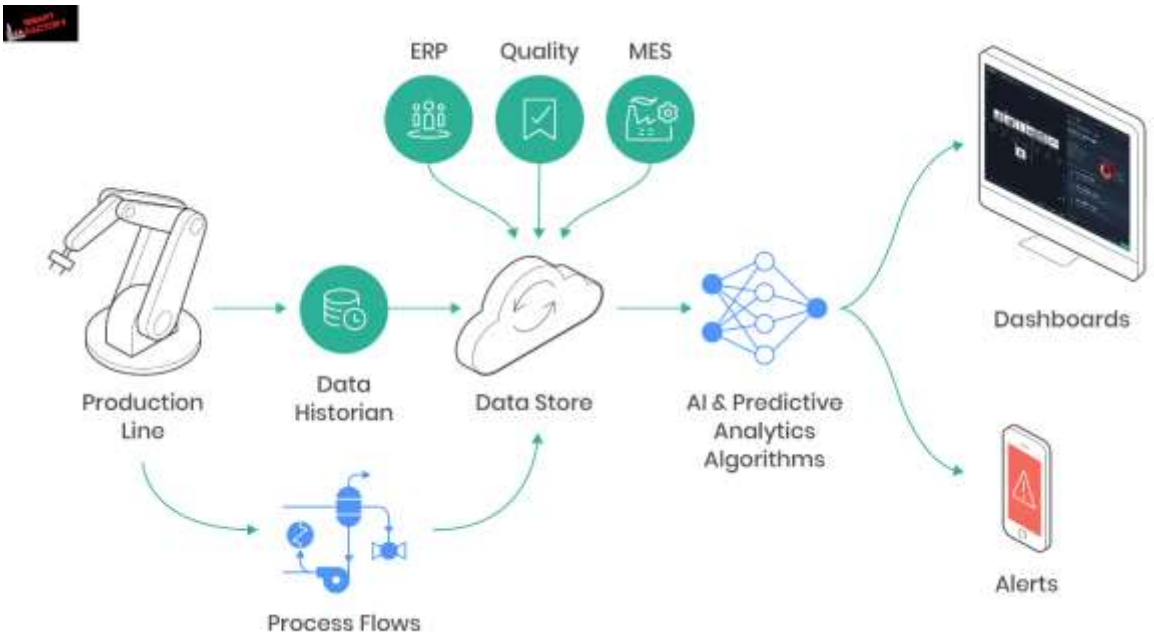


Figure 1.3: Predictive maintenance[3]

1.4 Urgency of the topic

In the current era of rapid industrialization, modernization, and digital transformation, manufacturing systems demand a higher degree of automation, reliability, and optimization than ever before. One of the critical challenges faced by production enterprises is maintaining the stable operation of machinery and equipment while minimizing unplanned downtime. Unexpected failures not only result in significant economic losses but also affect product quality, production schedules, and the company's reputation.

Traditional maintenance strategies, such as scheduled preventive maintenance or corrective maintenance after failure, show considerable limitations in high-precision and continuous production environments. Preventive maintenance may lead to resource

waste by replacing components that are still functioning, while corrective maintenance often results in serious downtime and costly repairs.

In this context, Predictive Maintenance (PdM) has emerged as an essential and effective solution for improving equipment reliability. The application of modern technologies such as the Internet of Things (IoT) enables real-time condition monitoring, early anomaly detection, and the forecasting of potential failures within a defined future timeframe, thereby allowing maintenance to be planned proactively, reducing risks, and optimizing maintenance operations.

The urgency and relevance of this project become even more apparent as businesses transition toward smart factories, where all devices are interconnected and decisions are made based on data-driven insights. Researching and implementing predictive maintenance systems using IoT not only delivers high practical value but also aligns with the development trend of Industry 4.0. Furthermore, it provides engineering students with the opportunity to engage with advanced technologies and apply theoretical knowledge to real-world industrial challenges.

Therefore, the project titled "Design of Predictive Maintenance System using IoT" is both timely and necessary, addressing the urgent needs of modern industrial production and contributing to the development of intelligent maintenance solutions in Vietnam.

1.5 Problem requirements

Objective:

To develop a machine learning model that predicts the likelihood of mechanical failure within the next 7 days, based on real-time sensor data.

Input Data:

- Vibration sensor data: (vibration amplitude)
- Shaft misalignment sensor data: (misalignment in mm)
- Temperature sensor data: (temperature in °C)
- Current sensor data: (current in amperes - A)
- Measurement timestamp: [timestamp]
- Machine condition label: [label]
  + 0 = Normal

+ 1 = Failure expected within 7 days

Output:

- Binary prediction:
  + 0 = Normal
  + 1 = Warning – Possible failure soon

Target Fault Conditions to Detect:

- Mechanical imbalance:
  + Gradual increase in vibration amplitude, especially at rotating frequency
- Worn or damaged bearings:
  + Irregular or directional vibration increases, especially along the shaft
- Bent or misaligned shaft:
  + Periodic increase in shaft misalignment

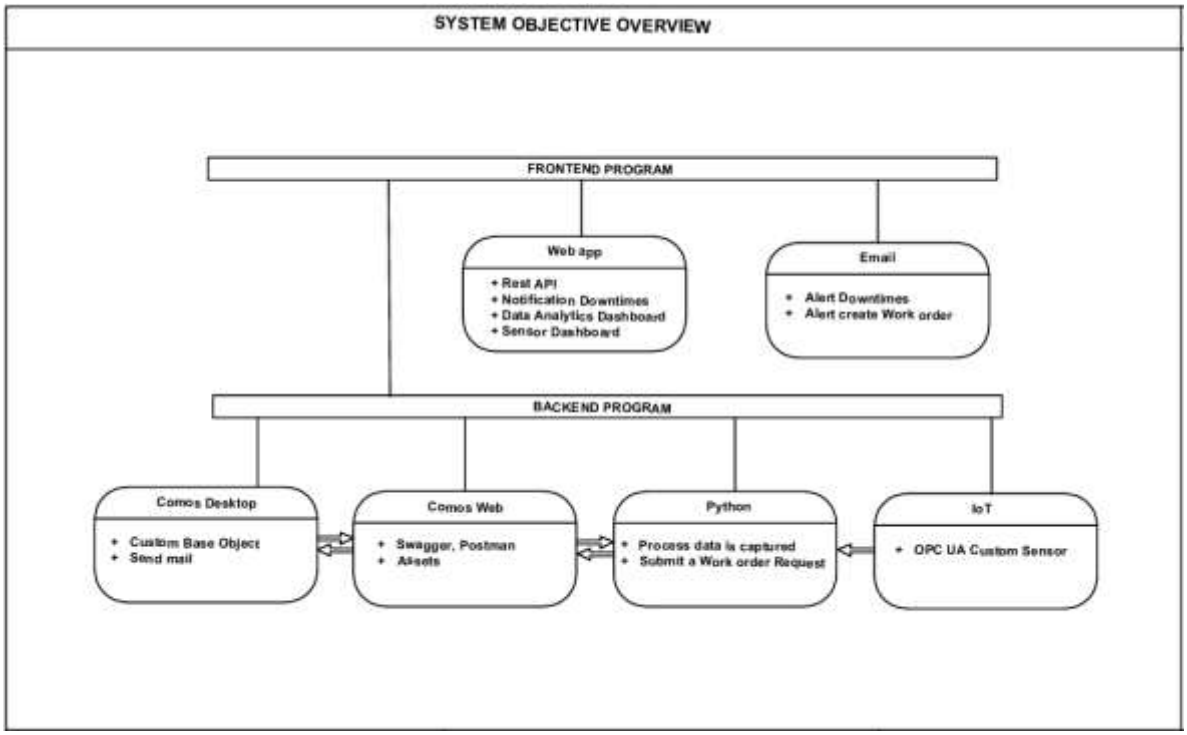


Figure 1.4: System Objective Overview

1.6 Content of Project

The predictive maintenance system using IoT operates based on a closed-loop process—from real-time data acquisition to automated alerting and maintenance

recommendations. The system integrates both hardware and software components to monitor the operational status of equipment, analyze data, and forecast potential failures within a defined future time window. The operating principle consists of the following key stages:

- Real time date acquisition
- Data Processing and Storage
- Equipment condition Analysis
- Forecasting future equipment status
- Alert threshoulds and automated response
- User Interaction

1.6.1 Real-time Data Acquisition

IoT sensors (such as vibration, temperature, and current sensors) are directly installed on industrial equipment to monitor critical parameters. These sensors continuously collect real-time data and transmit it to a local processor (microcontroller, edge gateway, or industrial PC) via wired (e.g., UART, RS485) or wireless communication (e.g., Wi-Fi, ZigBee, LoRa).

1.6.2 Data Processing and Storage

The collected data undergoes preprocessing to filter out noise, normalize values, and format the structure. Depending on system architecture, data may be stored locally or transmitted to a cloud platform for further processing and long-term storage. Time-series data is used for tracking historical trends and supporting predictive analytics.

1.6.3 Equipment Condition Analysis

The system applies statistical analysis or machine learning algorithms to assess the current condition of the equipment. By analyzing variations in vibration, temperature, and operating patterns, it identifies whether the equipment is operating normally, showing early signs of degradation, or approaching failure.

1.6.4 Forecasting Future Equipment Status

One of the system's core features is the ability to predict the condition of equipment over a defined period (the next 7 days). This is achieved by training AI or machine learning models on historical sensor data. The models identify abnormal trends and

predict possible failure points, enabling proactive maintenance planning before critical issues occur.

1.6.5 Alert Thresholds and Automated Response

Users can configure threshold values for each sensor parameter (e.g., maximum temperature of 85°C, maximum vibration amplitude of 5 mm/s). When any parameter exceeds its defined threshold, the system automatically triggers an alert, such as sending an email notification, displaying a real-time warning, or even generating a maintenance order. This ensures that potential problems are promptly addressed to prevent damage or downtime.

1.6.6 User Interaction

All sensor data, equipment statuses, and alerts are visualized through an interactive user interface. Users can monitor real-time charts, review historical data, track warning logs, and adjust system settings such as thresholds, prediction intervals, and maintenance schedules. The intuitive interface enhances usability and responsiveness in handling maintenance operations.

CHAPTER 2 ANALYSIS AND SELECTION OF DESIGN OPTIONS

2.1 Analysis of project requirements

2.1.1 General Overview

This project aims to develop a predictive maintenance system capable of monitoring industrial equipment using real-time sensor data (e.g., vibration, temperature, displacement). The system is designed to detect early signs of failure and provide timely alerts, thereby minimizing unplanned downtime and reducing maintenance costs.

2.1.2 Functional Requirements

System functional requirements

- Collect real-time sensor data from industrial equipment (via OPC UA or directly from PLCs).
- Store time-series data into a structured database (CSV, SQLite, or MySQL).
- Perform data preprocessing and feature extraction.
- Train and deploy machine learning models (e.g., LSTM, Random Forest Regressor) to predict equipment failure.
- Send early warnings through email, dashboard alerts, or messaging systems.
- Visualize real-time sensor data and prediction results via an intuitive interface.
- Manage devices, maintenance plans, and failure history records.

2.1.3 Non-Functional Requirements

Non-functional requirements of the system

- The system must run continuously and stably (24/7 operation).
- The user interface should be user-friendly and accessible to non-technical users.
- The system should be scalable to support additional devices and sensors.
- The alert latency should not exceed 10 seconds after anomaly detection.
- Ensure data security and user access control, especially in LAN or Internet-based deployments.

2.1.4 System Inputs and Outputs

- Input:

+ Real-time sensor data (vibration, temperature, shaft displacement, etc.)
+ Device configurations and warning thresholds
+ Labeled historical data for training predictive models

- Output:
  + Real-time trend graphs and sensor charts
  + Failure predictions and Remaining Useful Life (RUL) estimations
  + Early warnings and maintenance logs

2.1.5 Hardware and Software

- Hardware: Siemens S7-1200 PLC CPU 1214C DC/DC/DC, vibration sensors, temperature sensors, shalf displacement sensors, pressure sensor, current sensor
- Software: Python, SQLite / MySQL, OPC UA Server
- AI Models: LSTM, Random Forest Regressor

2.2 Maintenance options

2.2.1 Maintence and repair

These are troubleshooting activities when equipment, assets suddenly fail or potential faults are discovered during routine maintenance without a pre-planned maintenance plan.

_ Implementation process

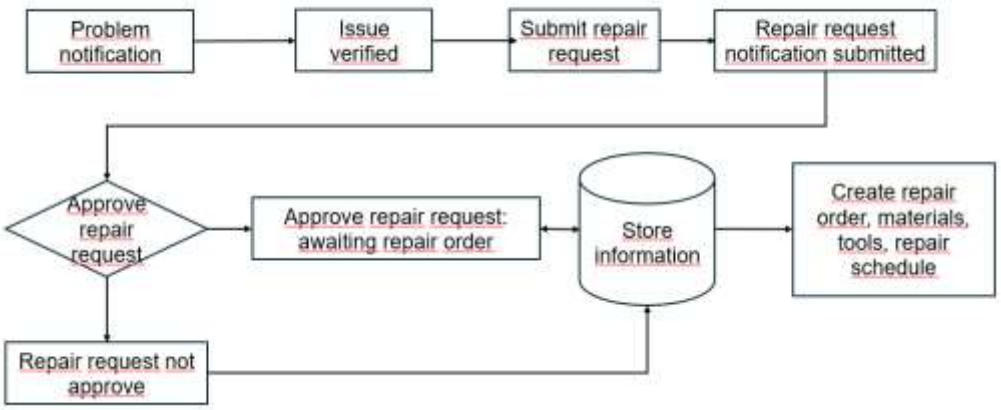


Figure 2.1: Reaction Maintenance Process

Notifications will be displayed via app notifications or sent via email.

Requests are defined by the approval flow according to the departmental structure of the unit.

Requests can change status, update information or be forwarded to a troubleshooting job. Data is aggregated to analyze equipment failures, calculate the operational KPI of each equipment (MTBF, MTTF, MTTR).

2.2.2 Preventice maintenance

Inspection, maintenance and repair activities are planned and performed periodically according to each type of equipment, ensuring stable operation of the equipment, minimizing damage and downtime, avoiding affecting operational efficiency.

- Implementation process

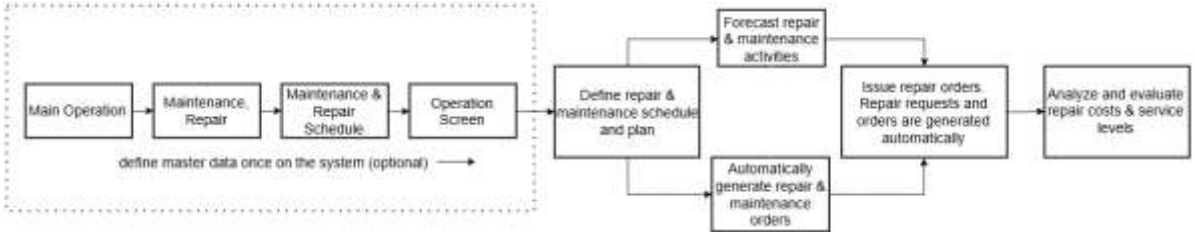


Figure 2.2: Preventive maintenance process

Based on the check sheet being monitored, schedule maintenance, servicing and repairs for groups of equipment and assets according to appropriate rules.

2.2.3 Predictive maintenance

It is the activity of monitoring operating parameters in real time, through measuring devices and sensors to monitor the status of equipment and assets. This activity can record and give warnings and predictions when the operating parameters of equipment and assets tend to deviate from standard parameters, thereby taking corrective actions to minimize damage to equipment.

- Implementation process

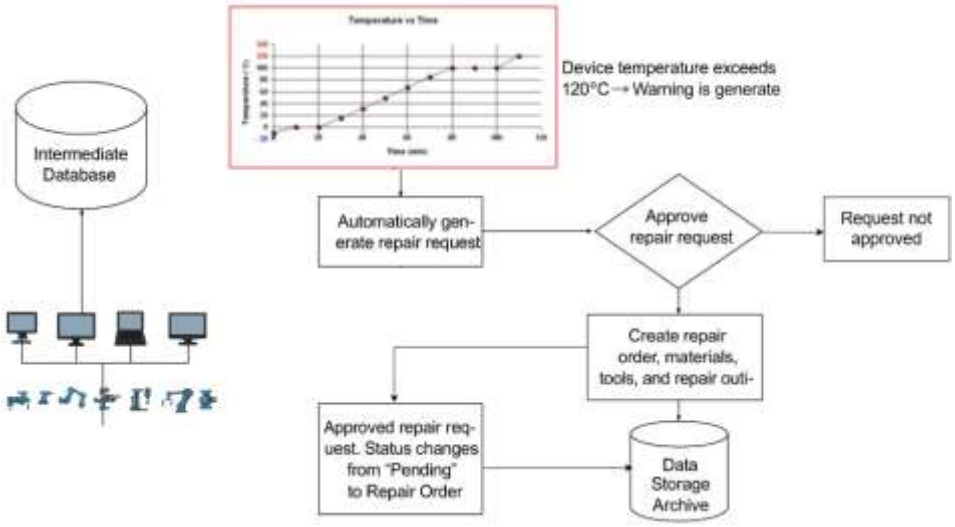


Figure 2.3: Predictive maintenance process

2.2.4 Distinguish between types of maintenance

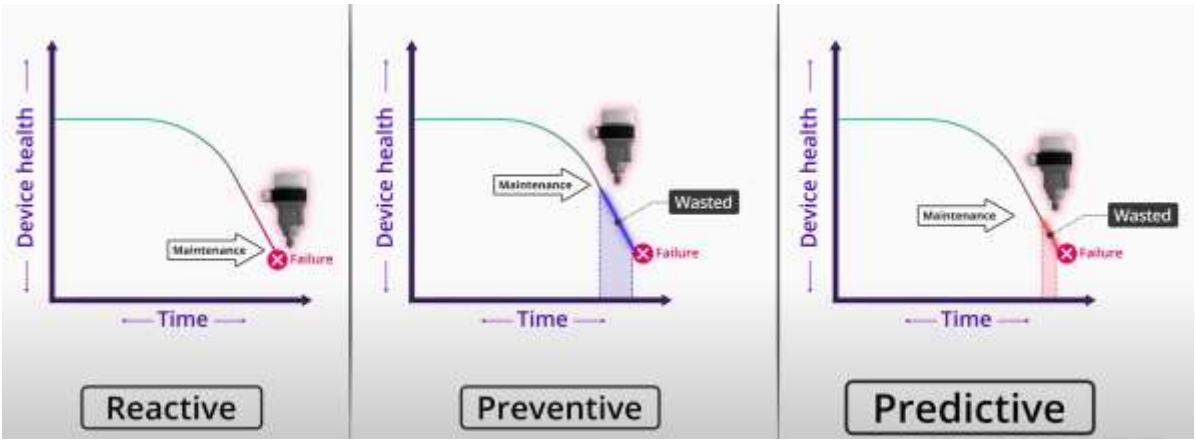


Figure 2.4: Distinguishing types of maintenance[4]

Table 2.1: Distinguishing types of maintenance[4]

	Reactive Maintenance	**Preventive Maintenance**	**Predictive Maintenance**
Definition	Only performed when the equipment fails or stops working.	Maintenance is scheduled periodically, regardless of equipment condition..	Real-time monitoring of sensor data to predict failures and perform timely maintenance.

Advantages	- No periodic maintenance cost - Simple and easy to apply	- Reduces risk of unexpected failures - Extends equipment lifespan	- Optimizes maintenance time and cost - Minimizes downtime - Timely maintenance
Disadvantages	- Causes production disruption - High sudden repair cost - Shortens equipment lifespan	- Periodic maintenance cost - Possible over-maintenance if not needed	- High initial investment (sensors, software, AI) - Requires high-quality data and expertise

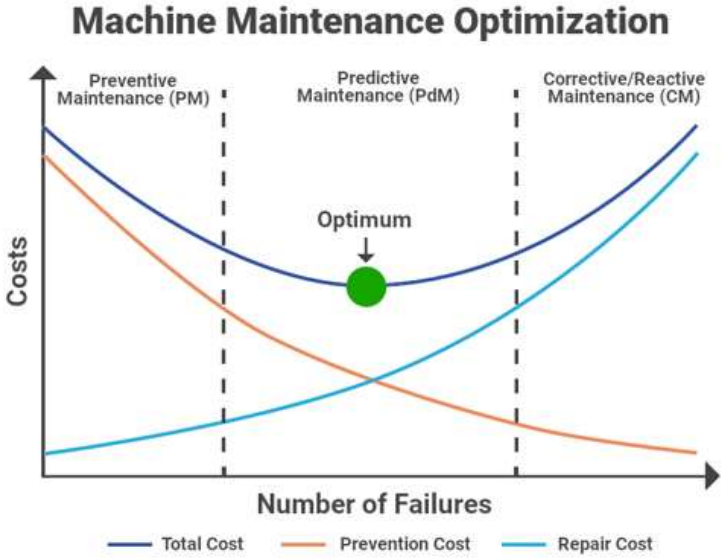


Figure 2.5: Optimizing maintenance using predictive maintenance [5]

⇒ **Conclude:**

Predictive maintenance overcomes the limitations of the above two methods by using real-time data and AI algorithms to accurately determine when maintenance is needed. This helps businesses:

+ Reduce repair and replacement costs.

+ Minimize unplanned downtime.

+ Improve system performance.

Figure 2.6: Improving operational efficiency [6]

Improve occupational safety in industrial environments.

Figure 2.7: Reducing the risk of accidents.[7]

The application of predictive maintenance not only makes economic sense but also plays an important role in ensuring the stability of the production system and improving product quality.

2.3 Get data

2.3.1 Using analog sensor

- Function

  + Reads analog signals (e.g., 0–5V, 0–10V, or 4–20mA)

+ Connects directly to PLCs, microcontrollers, or ADCs

- Advantages
  + Simple wiring and easy setup
  + Low cost and widely available
  + No complex configuration required
- Disadvantages
  + Limited transmission distance (typically < 50 meters)
  + Susceptible to electrical noise
  + Requires ADC if used with microcontrollers

2.3.2 Using sensor and communication protocol rs485

- Function
  + Communicates digitally via RS485 standard (typically Modbus RTU)
  + Transfers digital sensor data (e.g., temperature, vibration)
- Advantages
  + Long-distance communication (up to 1200 meters)
  + Strong noise immunity
  + Multi-drop support: multiple sensors on one bus
  + High data accuracy and reliability
- Disadvantages
  + Requires configuration (address, baud rate, protocol)
  + More complex to implement
  + Higher cost than typical analog sensors

2.3.3 Select data retrieval method

After evaluating both methods of sensor integration, the RS485 communication protocol is selected as the more suitable solution for this project.

Compared to analog sensors, RS485 offers long-distance transmission, better resistance to electrical noise, and the ability to connect multiple sensors on a shared

communication line (multi-drop), making it ideal for industrial environments. Additionally, digital data transmission over RS485 ensures higher accuracy and greater reliability.

Although analog sensors are simpler and less expensive, their limited range and vulnerability to noise make them less appropriate for scalable and robust predictive maintenance systems.

Therefore, sensors using the RS485 communication protocol are chosen to ensure system reliability, flexibility, and long-term performance in real-world industrial applications.

2.4 Processor

2.4.1 PLC s7-1200

- Function
  - + Industrial automation control
  - + Logic processing, real-time operation
  - + Interfaces with sensors, HMI, SCADA
  - + Supports Profinet, Modbus, analog/digital I/O
- Advantages
  - + High stability and noise resistance
  - + Designed for harsh industrial environments
  - + Integrated with TIA Portal (professional programming environment)
  - + Supports wide range of expansion modules
- Disadvantages
  - + Higher cost
  - + Requires licensed software (TIA Portal)
  - + Less flexible for low-level embedded control

2.4.2 Arduino mega 2560

_ Function

+ Embedded control for small projects

+ Handles basic sensor and actuator signals
+ Suitable for LED, motor, and sensor control
+ Supports UART, SPI, I2C communication

_ Advantages

+ Low cost and open-source
+ Easy to learn and program
+ Large online community and support
+ Simple development via Arduino IDE

_ Disadvantages

+ Not suitable for industrial environments (sensitive to noise)
+ Lacks electrical isolation and protection
+ May become unstable in long-term 24/7 operation

2.4.3 Select design option

After comparing functionality, stability, expandability, and suitability for industrial environments, the Siemens S7-1200 PLC is considered the more appropriate choice for control systems that require high reliability, continuous operation, and integration with industrial equipment.

While the Arduino Mega offers low cost and ease of use, its lack of industrial-grade protection, noise immunity, and robustness makes it unsuitable for real-world production systems.

Therefore, the S7-1200 PLC is selected as the main controller in this project.

CHAPTER 3 COMPONENTS OF THE SYSTEM

3.1 Overall architecture of the system

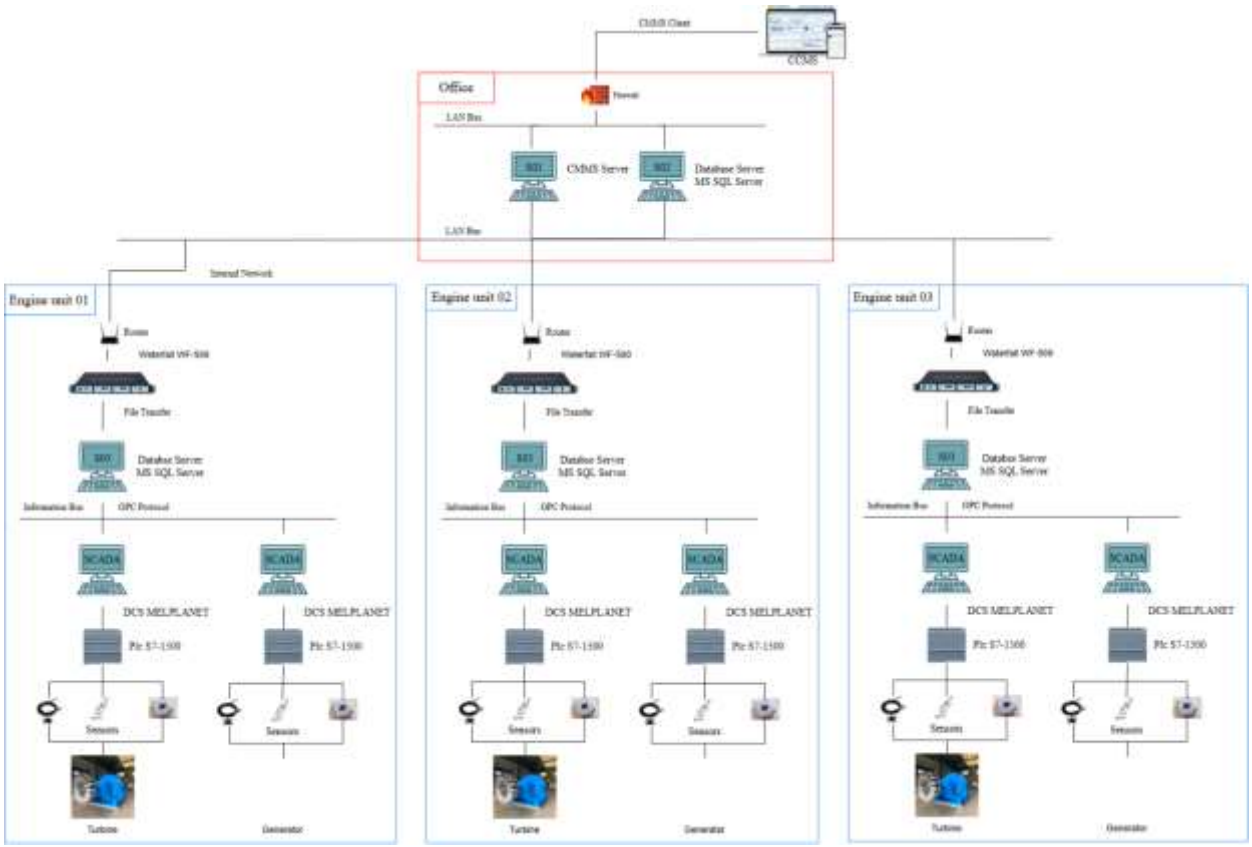


Figure 3.1: Overall architecture of the system

3.1.1 Office block

Main sections:

- S01 - CMMS Server:
- Dedicated server for maintenance, repair software (Maintenance, Repair, Overhaul) – storing incident information, maintenance schedules, reports.
- S02 - Database Server (MS SQL Server):
- Central database server – contains all device, operational, reporting data.
- Firewall:

Secure separation between office network and external enterprise network (Enterprise Network).

- CCMS (Client - Centralized Condition Monitoring System):

User interface (via web/app) to access maintenance data and monitor the entire system remotely (dashboard, reports...).

3.1.2 Engine Unit

- Network and data transmission:
- File Transfer:
- Processing equipment at the factory:
  + SCADA và RTUD1
  + DCS MELPLANET
- PLC and sensor:
  + PLC S7-1200 (Siemens)
  + Sensors:
  + OPC Protocol Communication

3.2 Sensors needed in the system

3.2.1 Temperature sensor

Figure 3.2: Temperature sensor GDSN10404403

- Function[8]: Monitors the temperature of motors, bearings, pumps, etc., to detect overheating caused by friction or overload.
- Operating Principle:
  + RTD (Resistance Temperature Detector): Measures the change in electrical resistance of a metal (e.g., Pt100, Pt1000) corresponding to temperature variations.
  + Thermocouple: Uses the Seebeck Effect (thermoelectric effect) to generate a voltage proportional to the temperature.

3.2.2 Vibration Sensor

Figure 3.3: Vibration Sensor

- Function[9]: Measures the vibration and oscillation of machinery to detect signs of imbalance, wear, or bearing failure.
- Operating Principle:
  + Utilizes the piezoelectric effect or MEMS (Micro-Electro-Mechanical Systems) to convert vibrations into electrical signals.
  + The collected data is sent to a PLC (Programmable Logic Controller) for processing and analysis.

3.2.3 Pressure Sensor

Figure 3.4: Pressure Sensor

- Function: Measures the pressure of lubricating oil, compressed air, and fluids in hydraulic systems.
- Operating Principle[10]:
  + The mechanical deformation of the sensor diaphragm generates an electrical signal.
  + Common types of sensors: Capacitive pressure sensors, piezoelectric sensors, or strain gauge sensors.

3.2.4 Shaft Displacement Sensor

Figure 3.5: Shaft Displacement Sensor

- Function: A shaft displacement sensor is used to measure the misalignment, vibration, or movement of rotating shafts in industrial machines. It helps detect imbalances, shaft bending, or bearing wear, supporting condition monitoring and predictive maintenance.
- Operating Principle[11]:
  + The sensor typically works based on eddy current or proximity detection principles.
  + As the shaft moves closer or farther from the sensor, the change in distance alters the electrical signal (voltage or current), which is used to determine shaft displacement.

3.2.5 Current Sensor

Figure 3.6: Current Sensor

- Function: Monitors the current consumption of motors to detect abnormal operating conditions.
- Operating Principle[12]:
  + Uses a Hall sensor or CT (Current Transformer) to measure the electrical current.
  + Detects overload conditions, phase loss, or motor wear.

3.3 Processor

3.3.1 PLC Siemens S7-1200 CPU 1215C DC/DC/DC

Introduction[13]

- PLC S7 1200 is a basic PLC programmable controller with a compact and flexible modular design. PLC S7-1200 fully supports control and communication functions, suitable for various automation applications, and for small to mediumscale applications.
- PLC S7 1200 is widely used in industrial and civil applications such as conveyor systems, lighting control, highpressure pump control, packaging machines, printing machines, textile machines, mixers, and distribution stations. electricity, heating/cooling control…

Figure 3.7: PLC Siemens S7-1200 CPU 1215C DC/DC/DC

Technical Specifications

Table 3.1: Technical Specifications

Specification	Value / Description
Order Number	6ES7 215-1AG40-0XB0
CPU Type	1215C

Power Supply	24 VDC (20.4 – 28.8 VDC)
Digital Inputs (DI)	14 channels (24V DC, Sink/Source)
Digital Outputs (DO)	10 channels (24V DC, Transistor)
Analog Inputs (AI)	2 channels (0 – 10V)
Analog Outputs (AO)	None (requires expansion module)
Program Memory	75 KB
Data Memory	2 MB (with SD card support)
Working Memory (RAM)	100 KB
Instruction Execution Time	0.08 µs / logic bit
Communication Ports	1x Ethernet (Profinet 10/100 Mbps)
Supported Protocols	Profinet, Modbus TCP (upgradeable)
Built-in Features	PID control, Web Server, RTC, HMIsupport, Data Logging
Dimensions (W x H x D)	110 x 100 x 75 mm
Operating Temperature	–20 °C to +60 °C
Expansion Capability	8 I/O modules + 3 communication modules + SD memory card

24 VDC Sensor Power Supply

- For increased noise immunity, connect "M" to chassis ground even when not using sensor power.
- For sink inputs, connect "-" to "M" (display).
- For power inputs, connect "+" to "M".

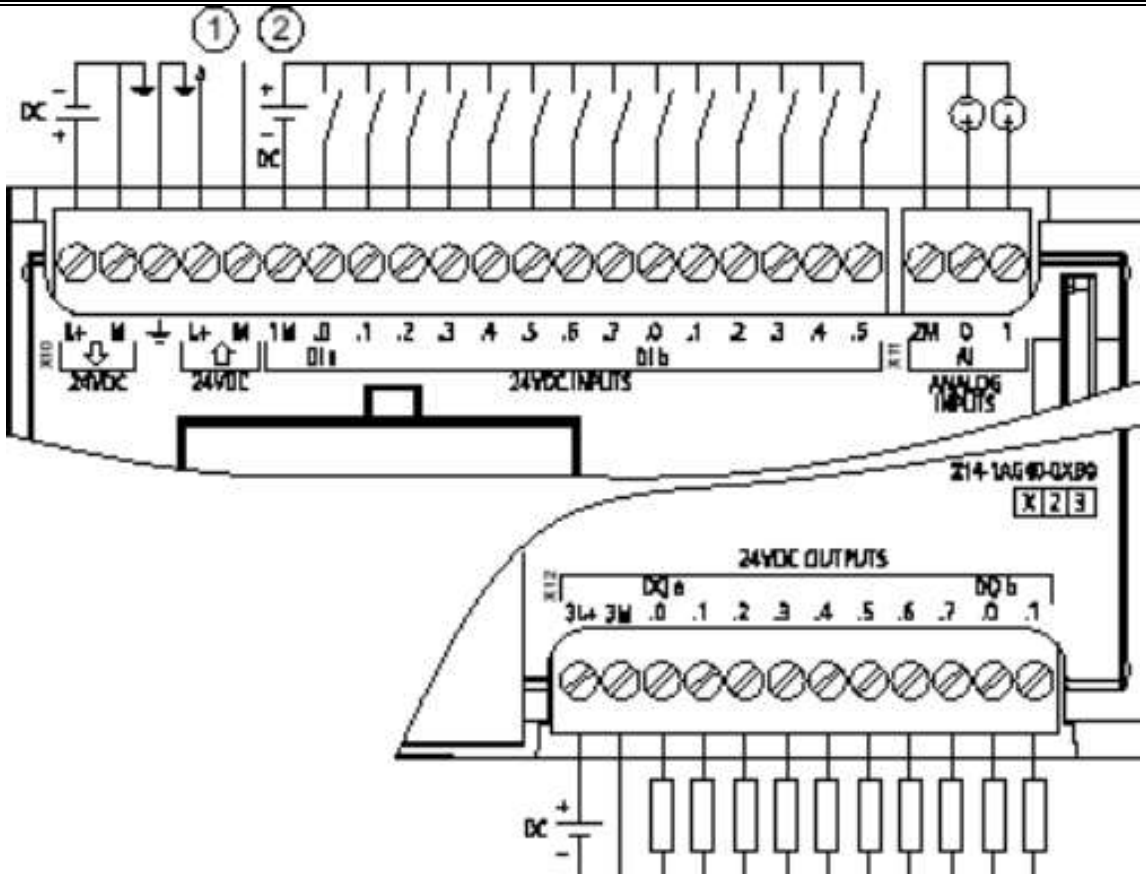


Figure 3.8: Communication pin diagram PLC 1215C DC/DC/DC [14]

3.3.2 Siemens SM 1231 RTD AI Module

Funtion[15] :

- The SM 1231 RTD is an analog input expansion module for the Siemens S7-1200 PLC.
- It is used to measure temperature via RTD sensors such as PT100, PT1000, etc.
- It connects directly to the S7-1200 CPU via the left-side expansion port.

Figure 3.9: SM 1231 AI

Table 3.2: SM 1231 RTD AI Model Specifications

Parameter	Value / Description
Analog Input Channels	4 or 8 channels (depending on model)
Supported Sensors	RTDs: PT100, PT1000, Ni100, Ni1000, Cu10, etc.
Resolution	16-bit
Measuring Range	Depends on RTD type (typically –200 to +850°C)
Input Signal Type	Resistance (Ω) via Wheatstone Bridge principle
Connection Types	2-wire, 3-wire, or 4-wire
Power Supply	24 V DC (from PLC or external source)

- Advantages:
  + High-accuracy temperature measurement
  + Sensor diagnostics built-in
  + Seamlessly integrates into TIA Portal (STEP 7)
  + Compact DIN rail mountable, directly next to S7-1200 CPU
- Typical Applications:
  + Monitoring industrial equipment temperature (motors, gearboxes, etc.)
  + HVAC and furnace temperature control
  + Predictive maintenance (PdM) systems using RTD temperature input

CHAPTER 4 CMMS MAINTENANCE SOFTWARE

4.1 CMMS software interface[16]

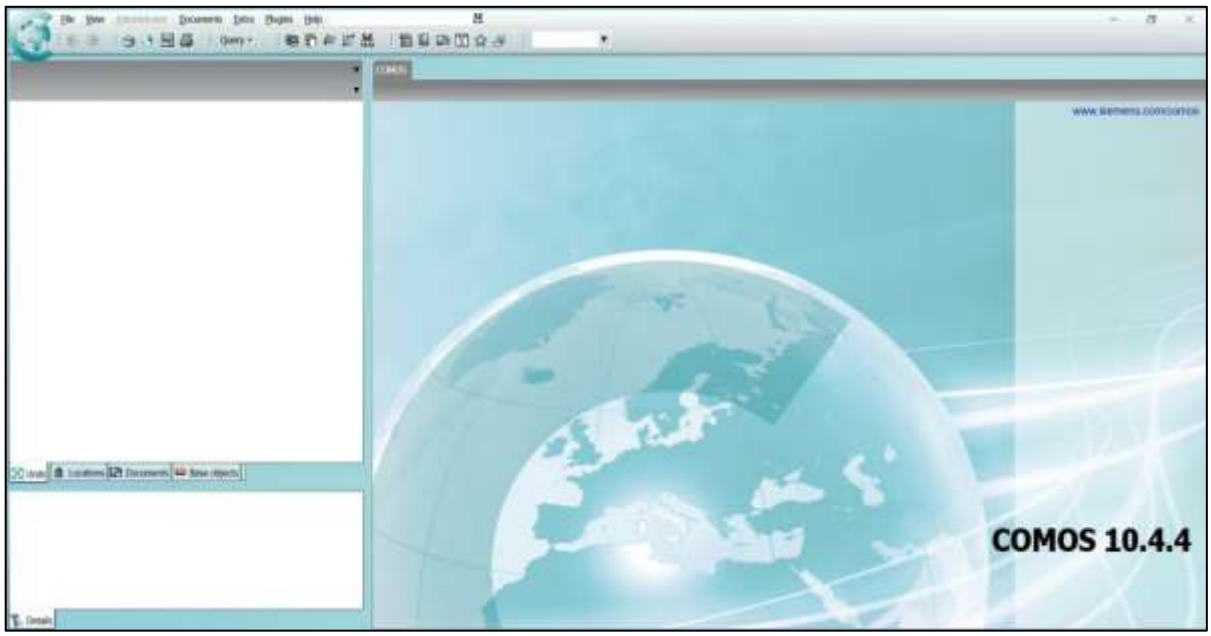


Figure 4.1: CMMS software interface

As an asset management software, machine asset information is in the form of a directory tree according to the equipment level in the factory, from large systems to small systems to equipment that needs maintenance, and further down are the components in the equipment.

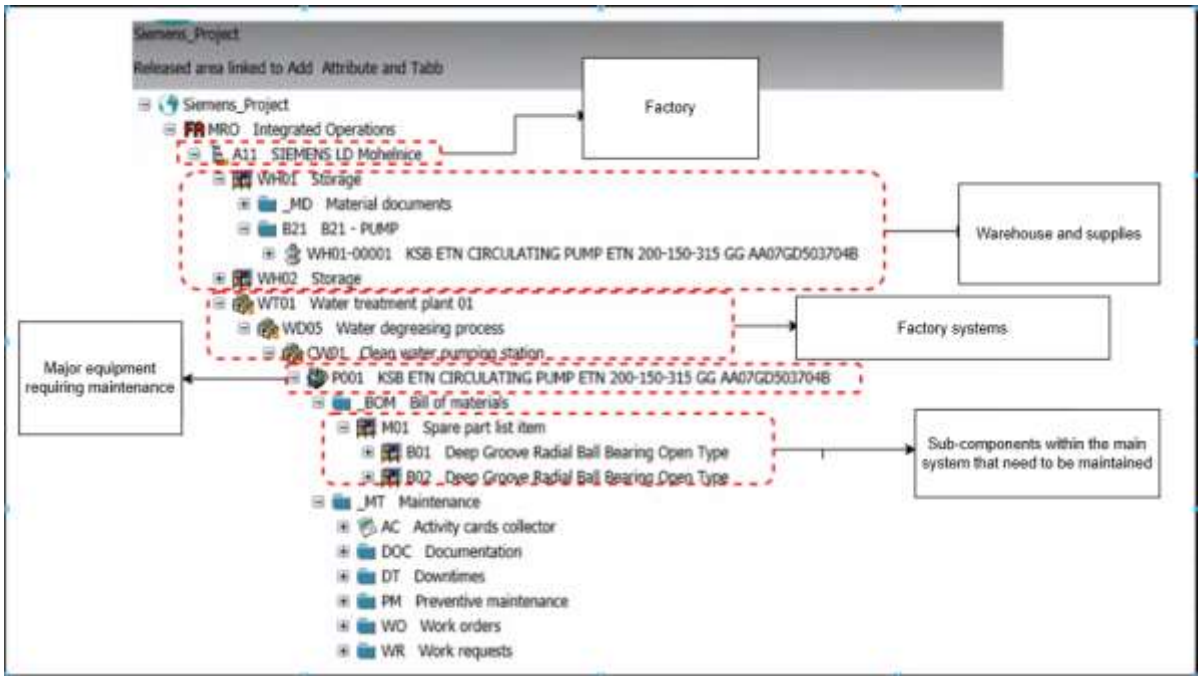


Figure 4.2: Directory tree in CMMS software

Personnel management by company department level

Data asset information is linked together through a common master data

COMOS MRO software has maintenance strategies:

- Preventive maintenance
- Reactive maintenance
- Condition Based maintenace
- Risk-based maintenance,…

4.2 Create a new Maintenance Project

4.3.1 Create a Working Layer

Click on the icon  **(Open Database)**

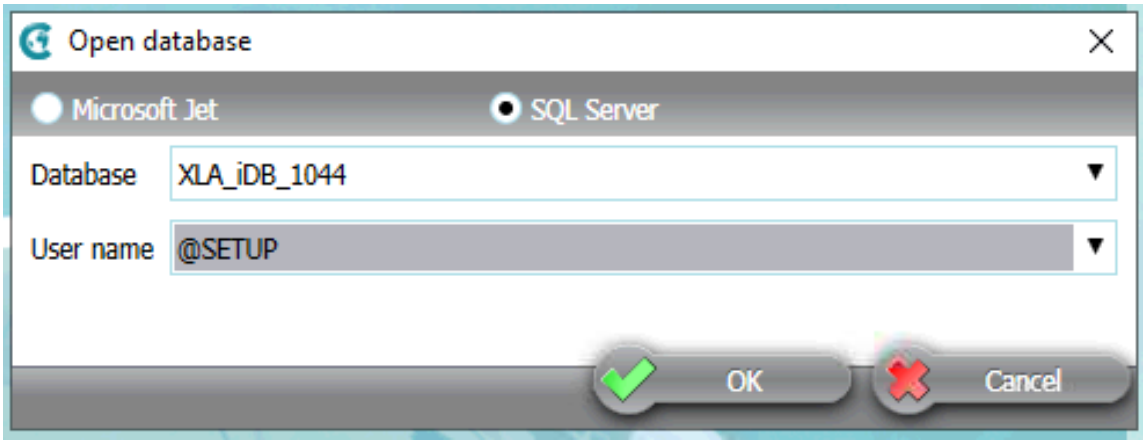


Figure 4.3: Open database window

Select information as shown and press OK

Click on the icon  (**Open Project)**

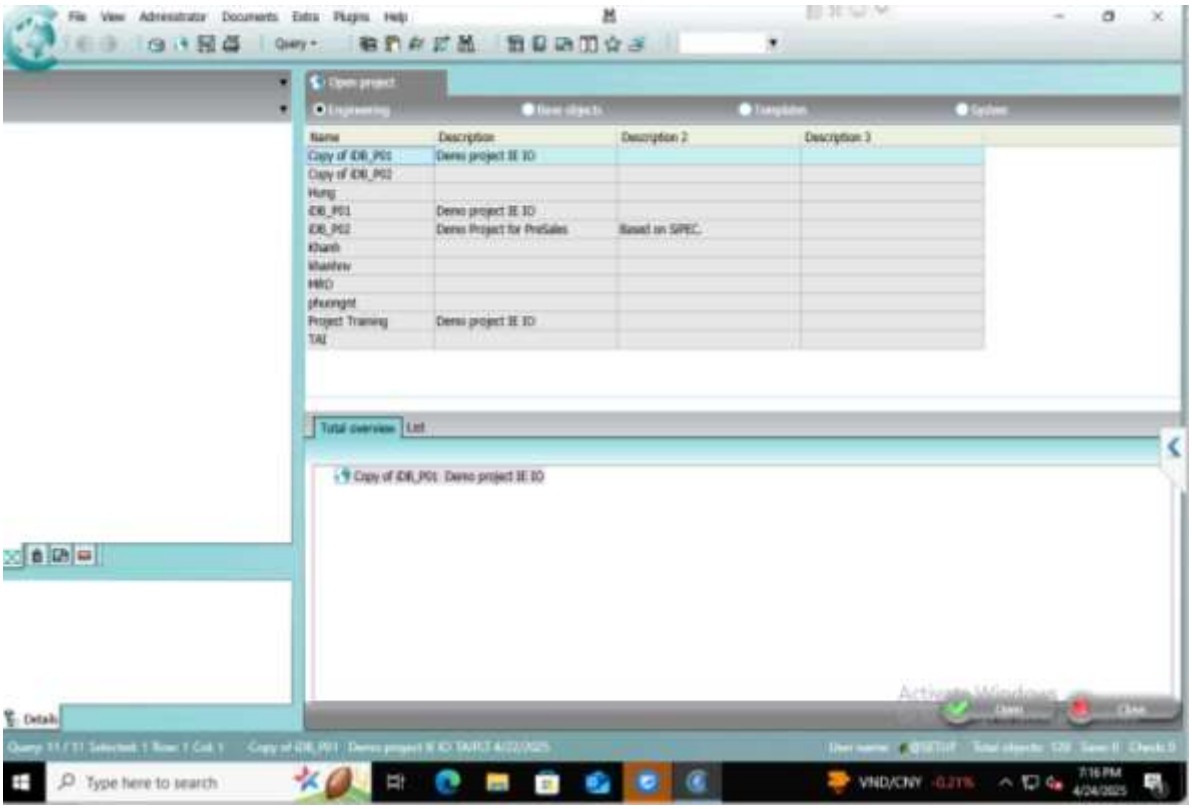


Figure 4.4: Open Project interface

Initialize a new Working Layer in Base objects

- Purpose: To create new Working Layer versions that inherit the old Working Layer above and when adjusting the new Working Layer, it will not cause errors in the old Working Layer. After the Working layer is adjusted without causing errors, it can be released and merged to the original Working Layer or Project.
- Scope: create Working Layer in Engineering project and Base project

Open the Base objects window and click on the required project in Working Layer the Total overview window appears.

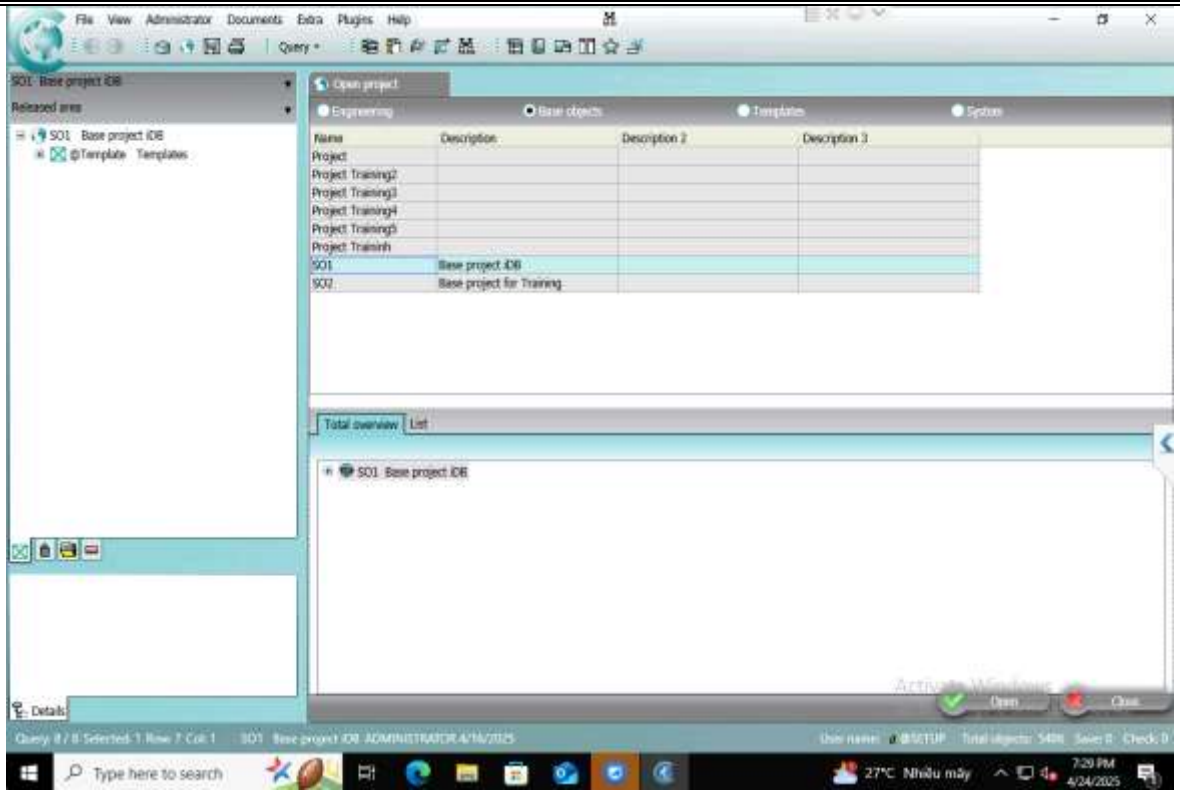


Figure 4.5: Base object

Create a new Working Layer from any existing Working Layer:

Right click any existing Working Layer ➔ select New ➔ select Working Layer

The new Working Layer is created and the Working Layer information dialog opens

Adjust the Working Layer information and press Apply

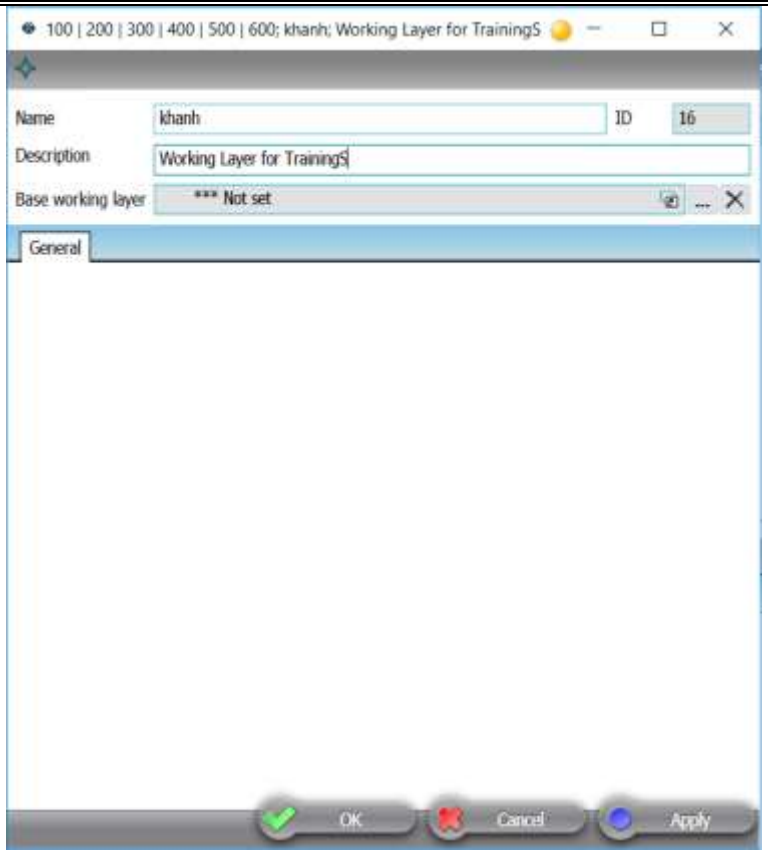


Figure 4.6: Working Layer information dialog box

4.3.2 Create a Project

Open the Engineering window

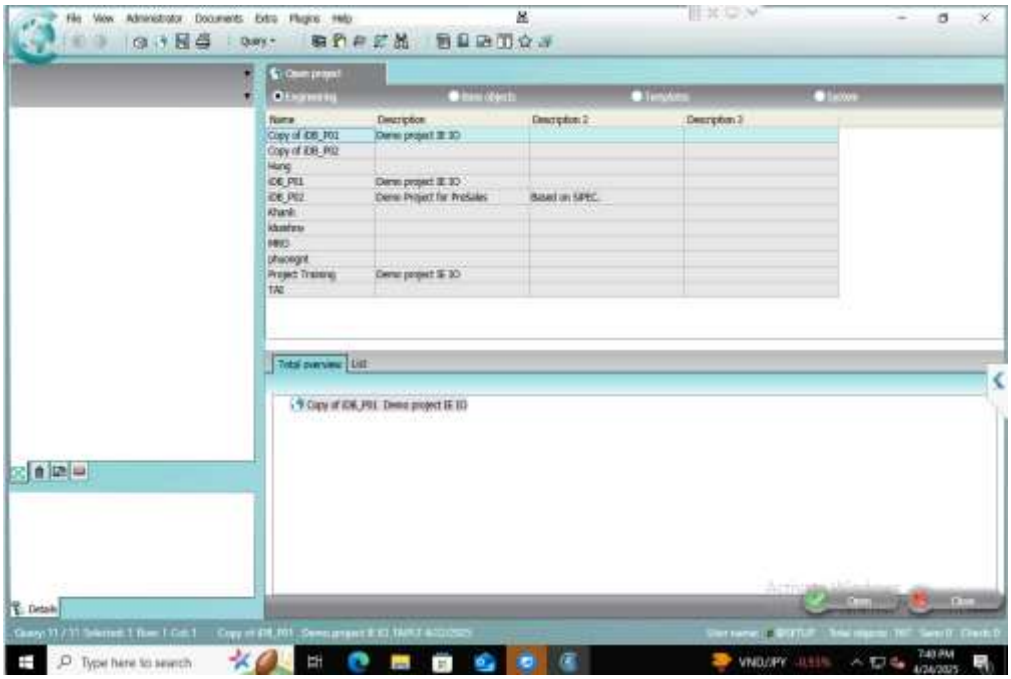


Figure 4.7: Engineering window

Click ➔ new ➔ Project

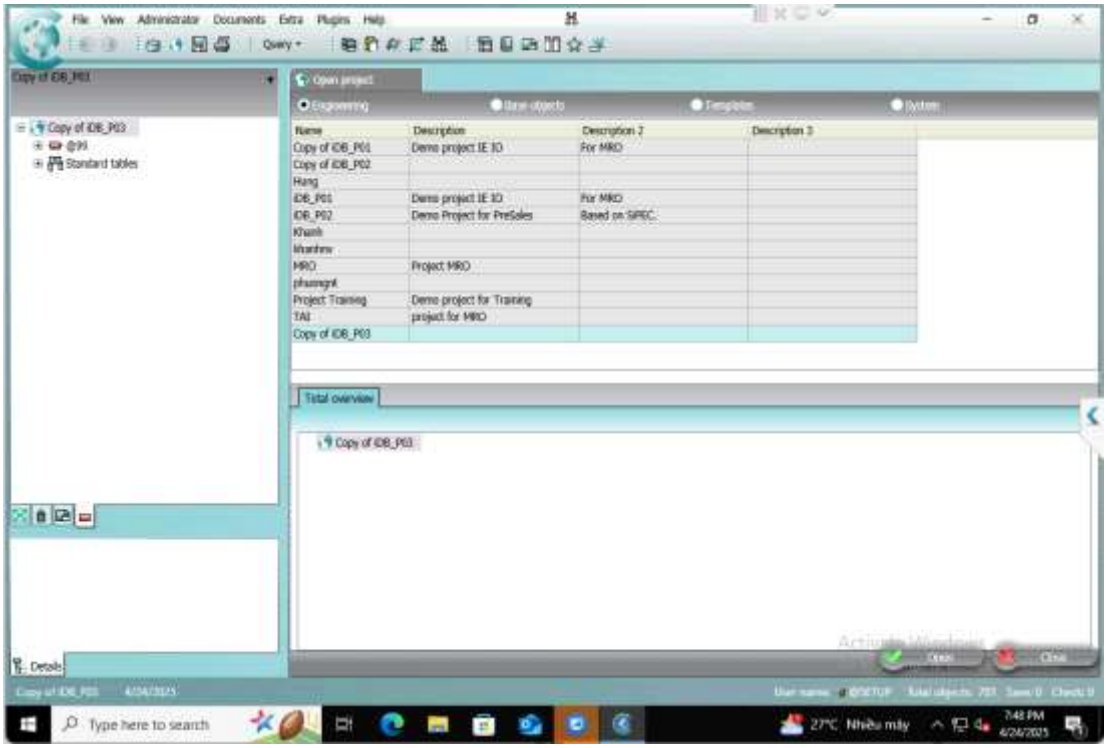


Figure 4.8: Newly created project

Name the project and edit the project configuration with the properties section.

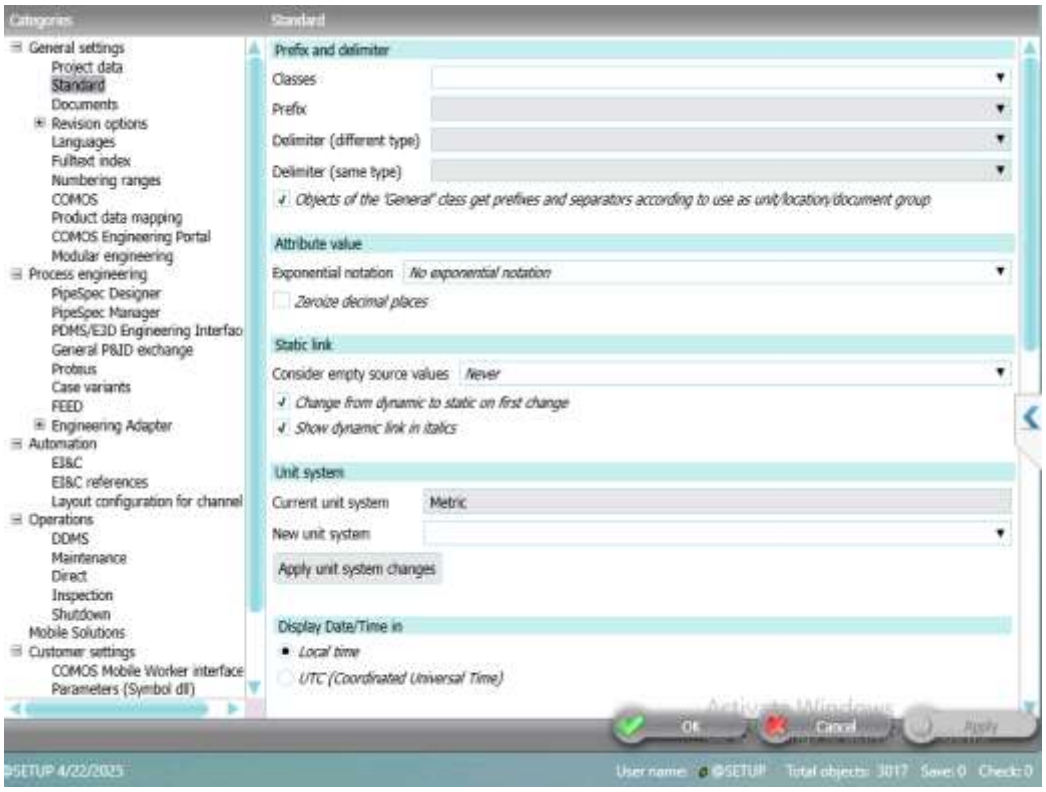


Figure 4.9: Standard window

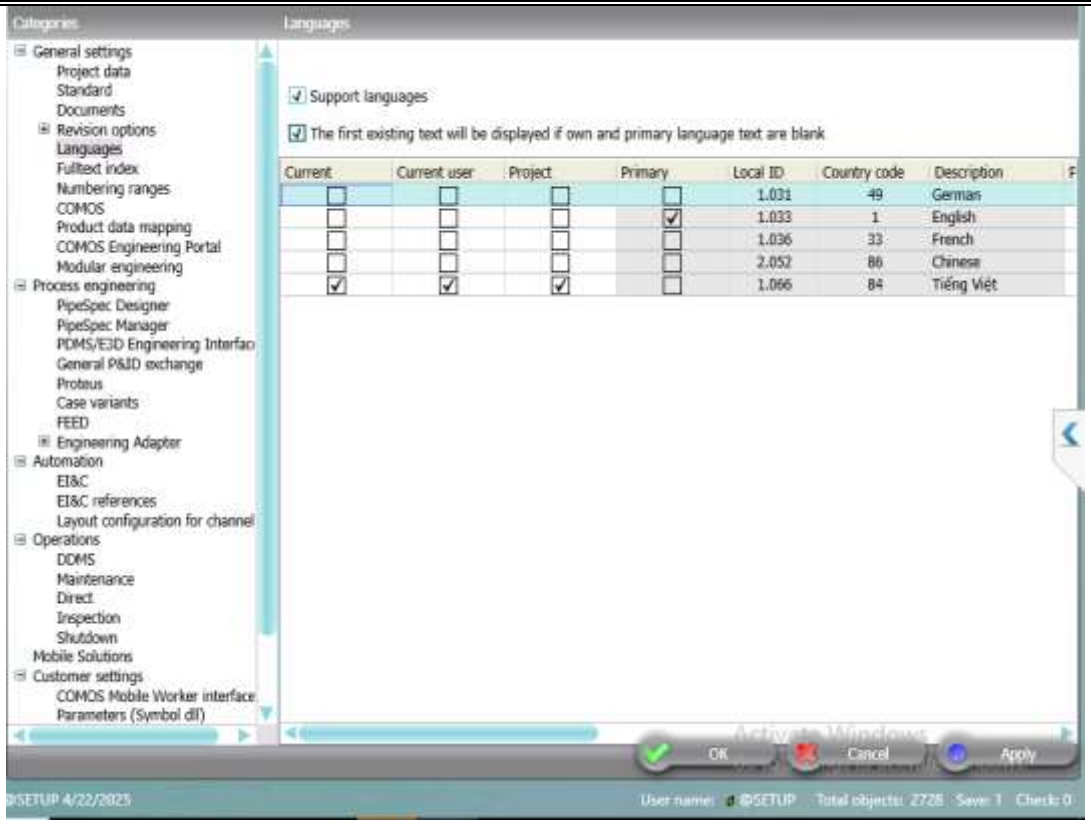


Figure 4.10: Language window

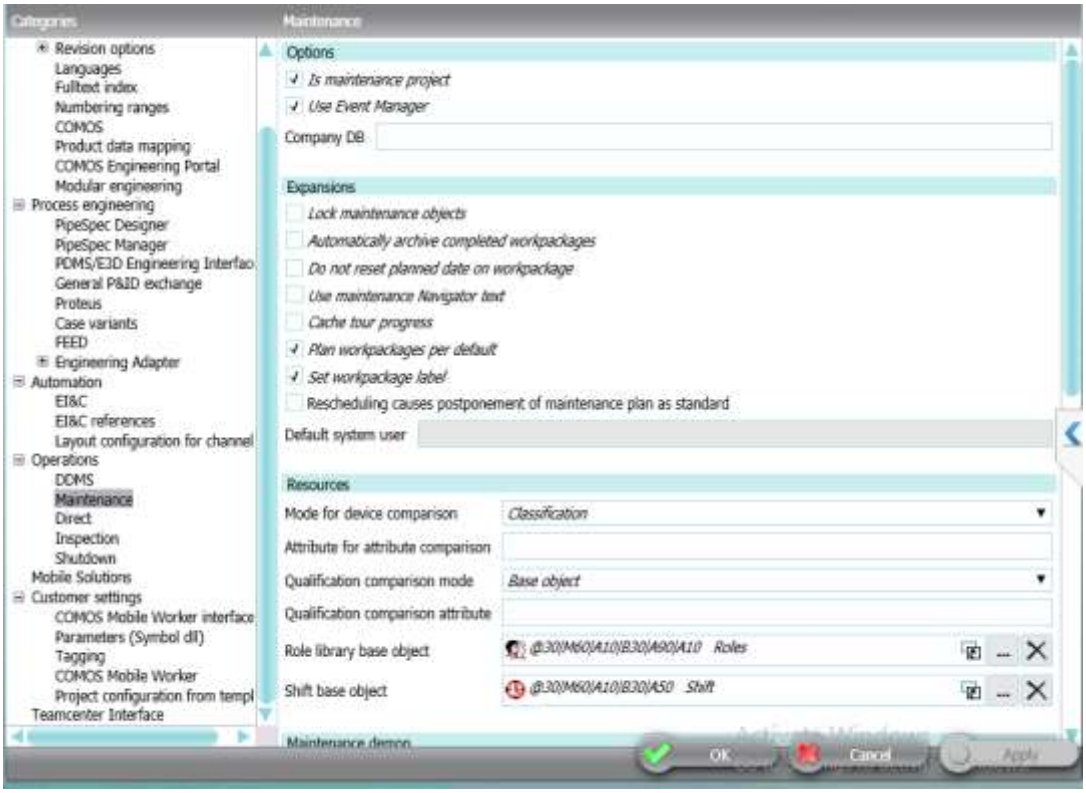


Figure 4.11: Maintenance window

4.3 Create device directory tree

Purpose: Create a directory tree to manage equipment according to tree structure level, maintenance management, material warehouse management. Convenient for searching, adding and removing data.

Continue to create an MRO module in Project MRO.

Right click on the current MRO Project➔ select New➔ select A10 Integrated Operations.

Adjust MRO module information

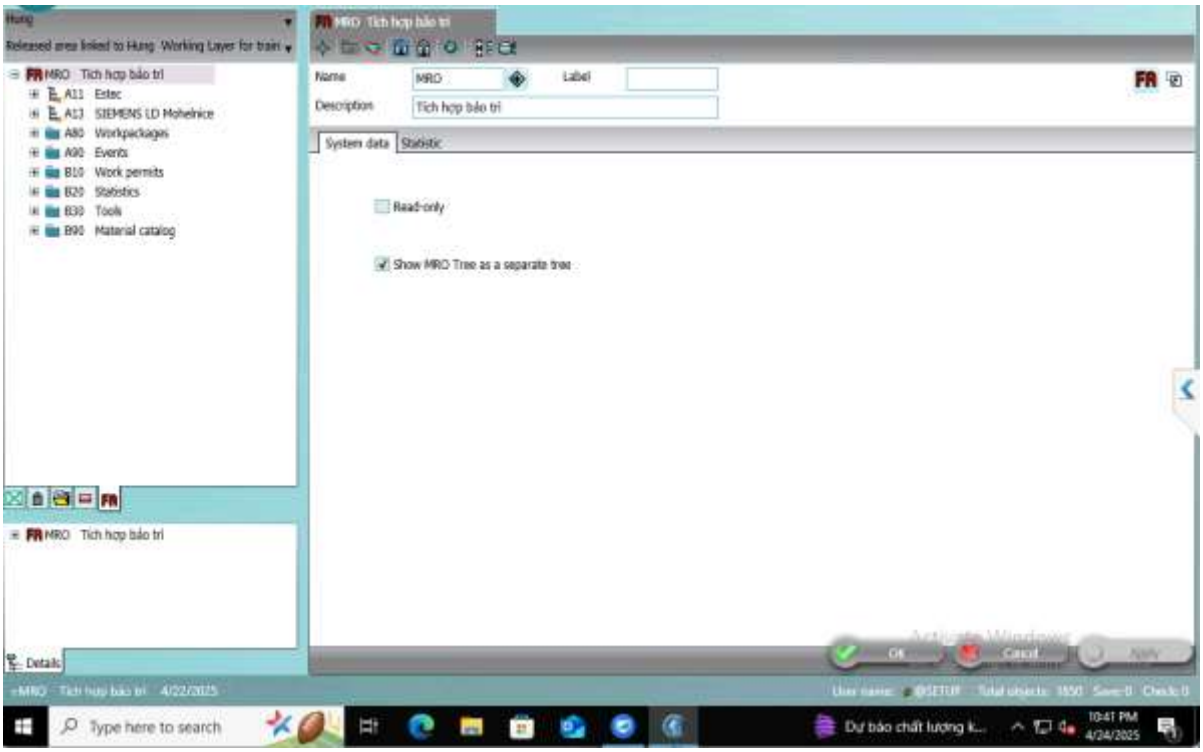


Figure 4.12: MRO information

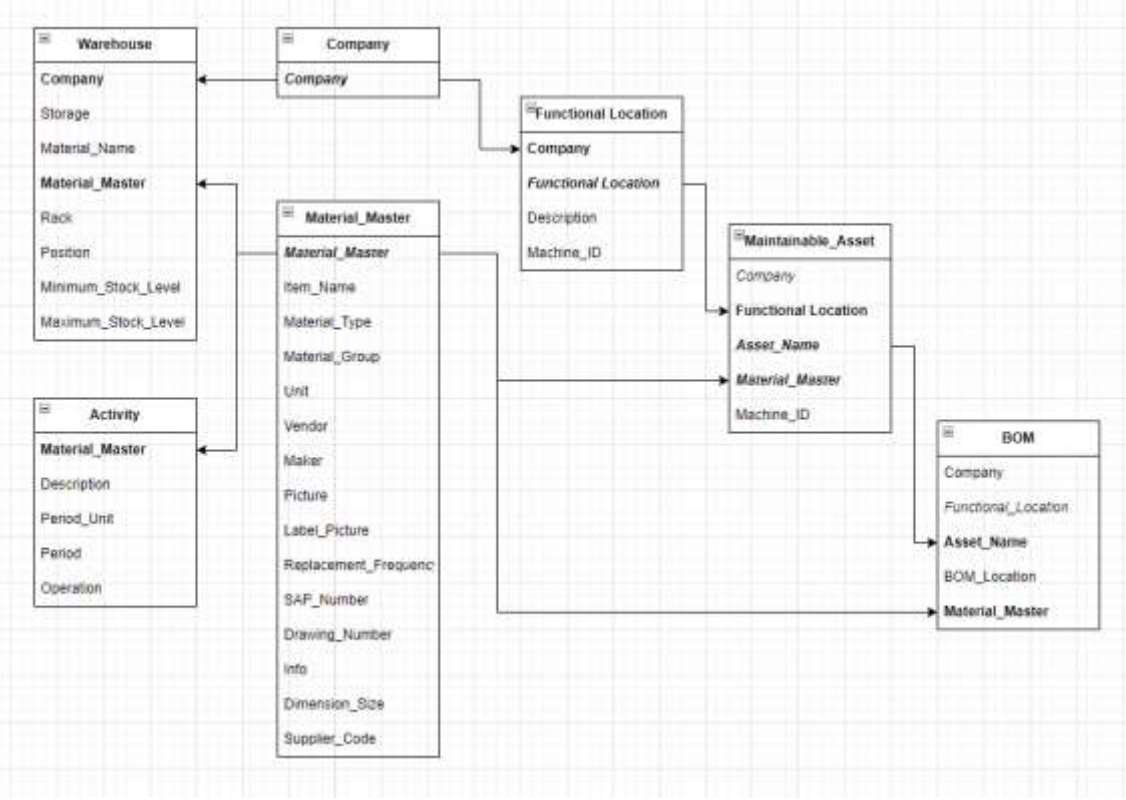


Figure 4.13: Diagram of exporting excel files into folders

- Customers/Makers/Suppliers: Sheet containing equipment suppliers
- Material Master: Master data sheet to link all sheet information together
- Activities: The sub-tasks of the sample task package after import will be in the Material_Master folder.
- Companies: Maintenance company
- Function Locations: Large systems of maintenance company
- Maintainable Assets: Major equipment requiring maintenance
- Asset BOM: Components contained in the equipment requiring maintenance
- Warehouse Items: Inventory items to replace Maintenance Assets or Asset BOM

4.3.1 Create device directory tree manually

Select the module "MRO Integrated Maintenance" ➔ New ➔ Company reference.

Then the new company is created ➔ continue to adjust the name and description of the equipment.

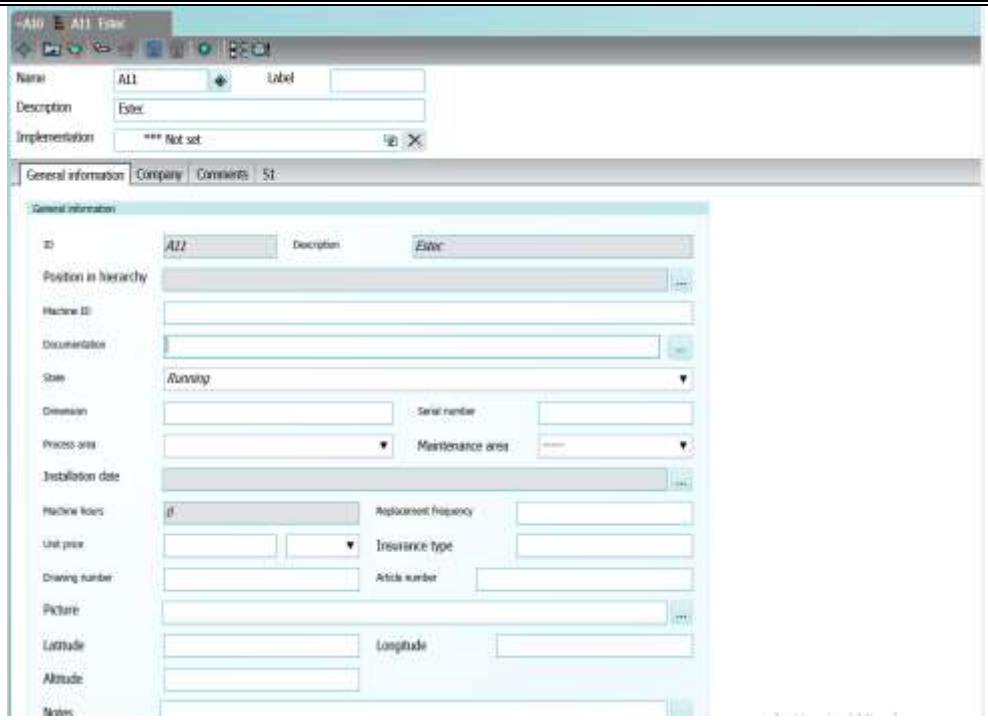


Figure 4.14: Creating a factory

From the company continue to create systems by selecting company ➔ New ➔ System

Continue to change the name, description and add system details

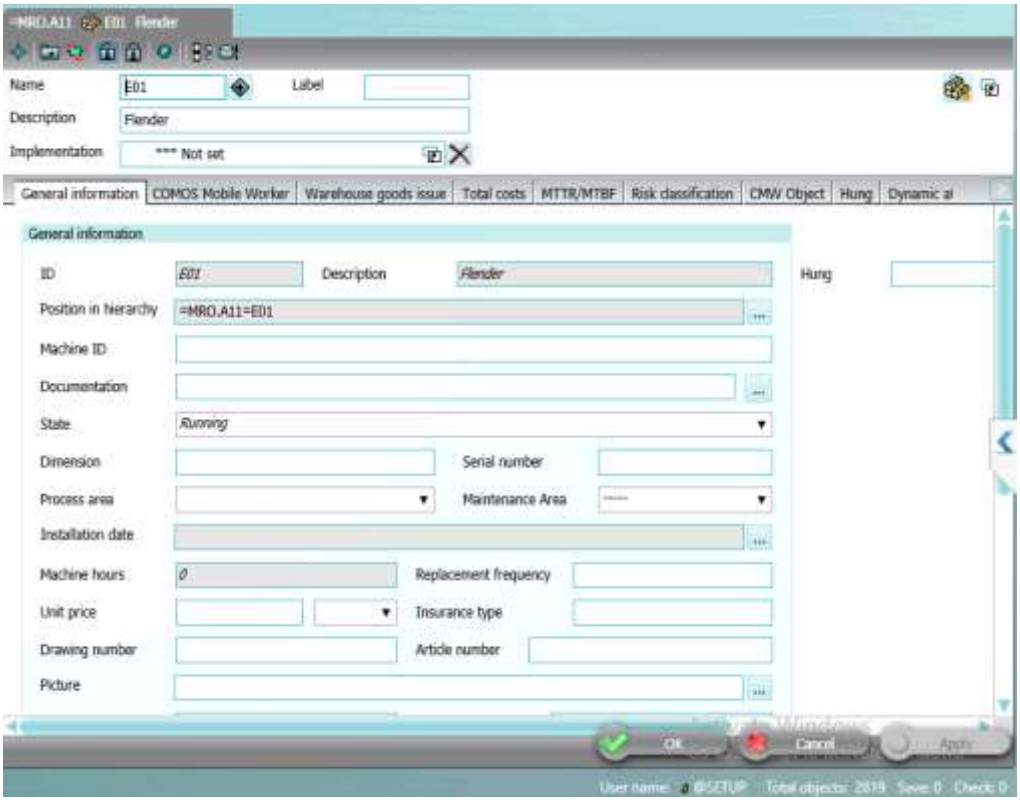


Figure 4.15: Creating a system

From created system ➔ New ➔ General maintenance equipment

Continue to change the name and information for the equipment

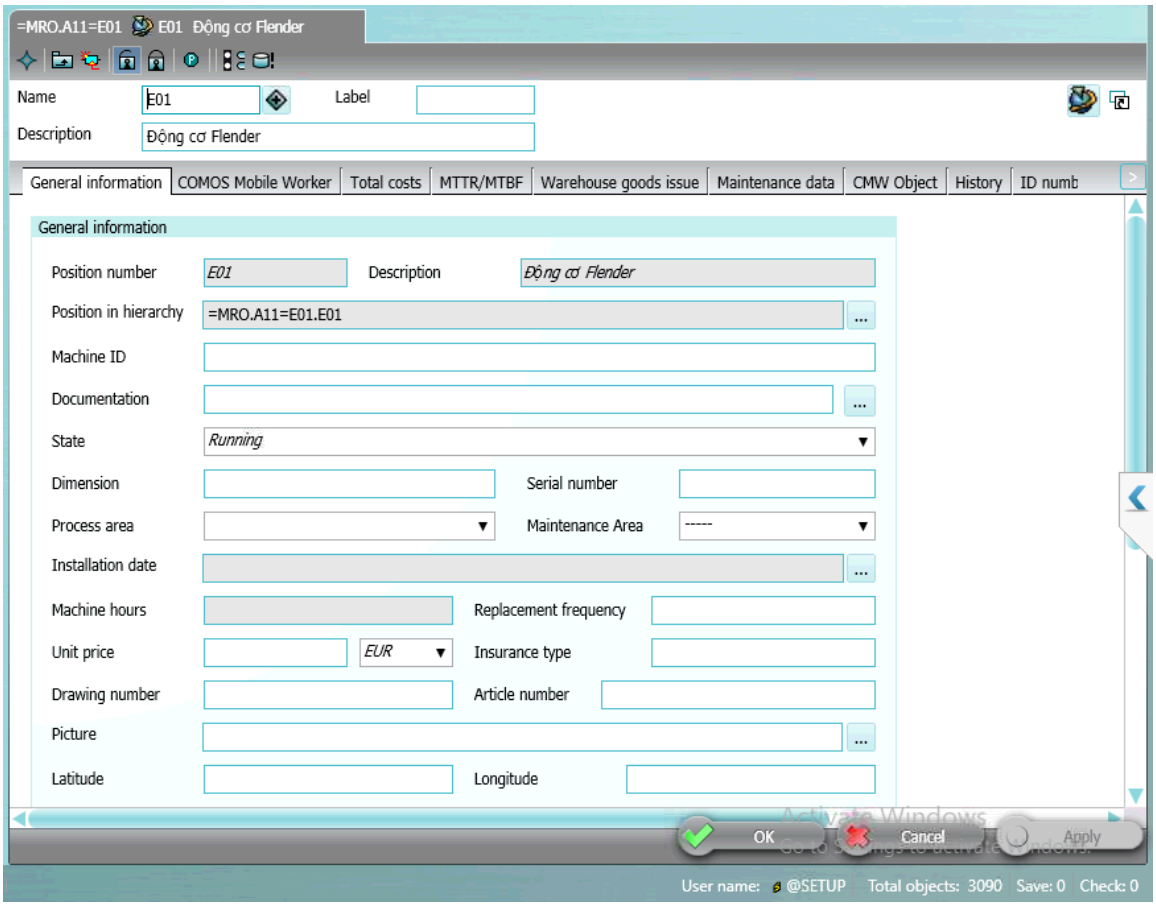


Figure 4.16: Create device

Create a BOM list of the device including:

+ Gearbox
+ Rotor
+ Stator

At BOM folder ➔ New ➔ List item.

Change the information of the BOM sub-device.

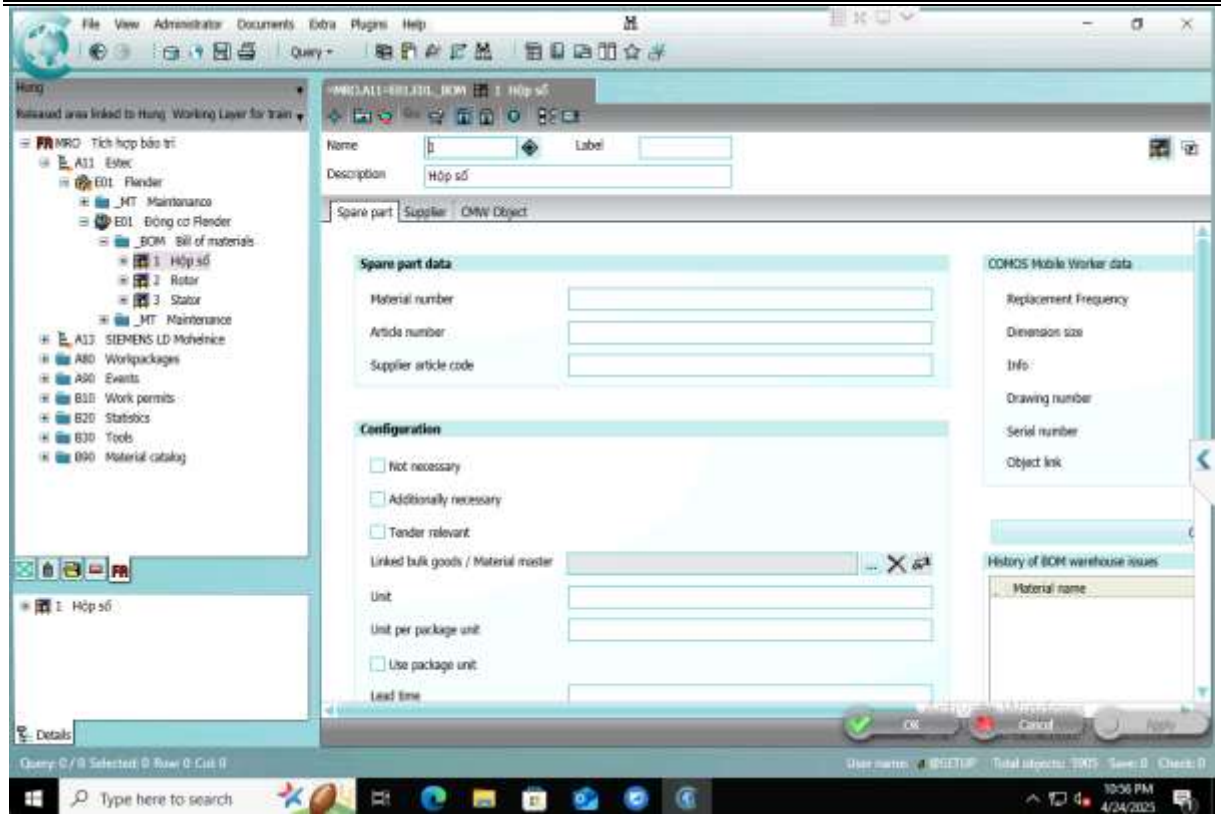


Figure 4.17: Creating a device BOM list

4.3.2 Create device directory tree by importing Excel file

Link structure and import order of Excel file: when importing, it must follow the order as shown in Figure 4.17 so that the data is fully loaded and tightly linked.

The structure of the MRO excel import file must match the structure of the import tables of the B30 Tools section in COMOS_MRO, including 8 sheets:

Figure 4.18: 8 sheets in B30 Tools section

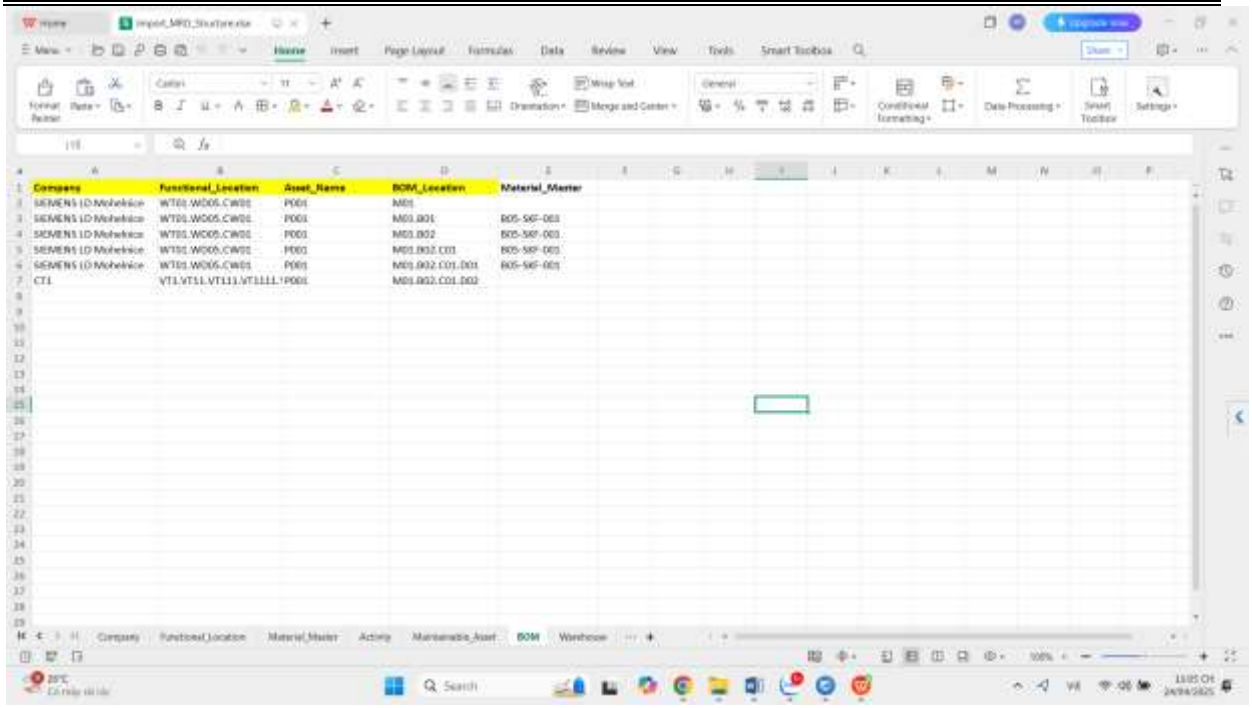


Figure 4.19: Available Excel file

Select the project to import ➔ select the data column to import ➔select the Excel file to import

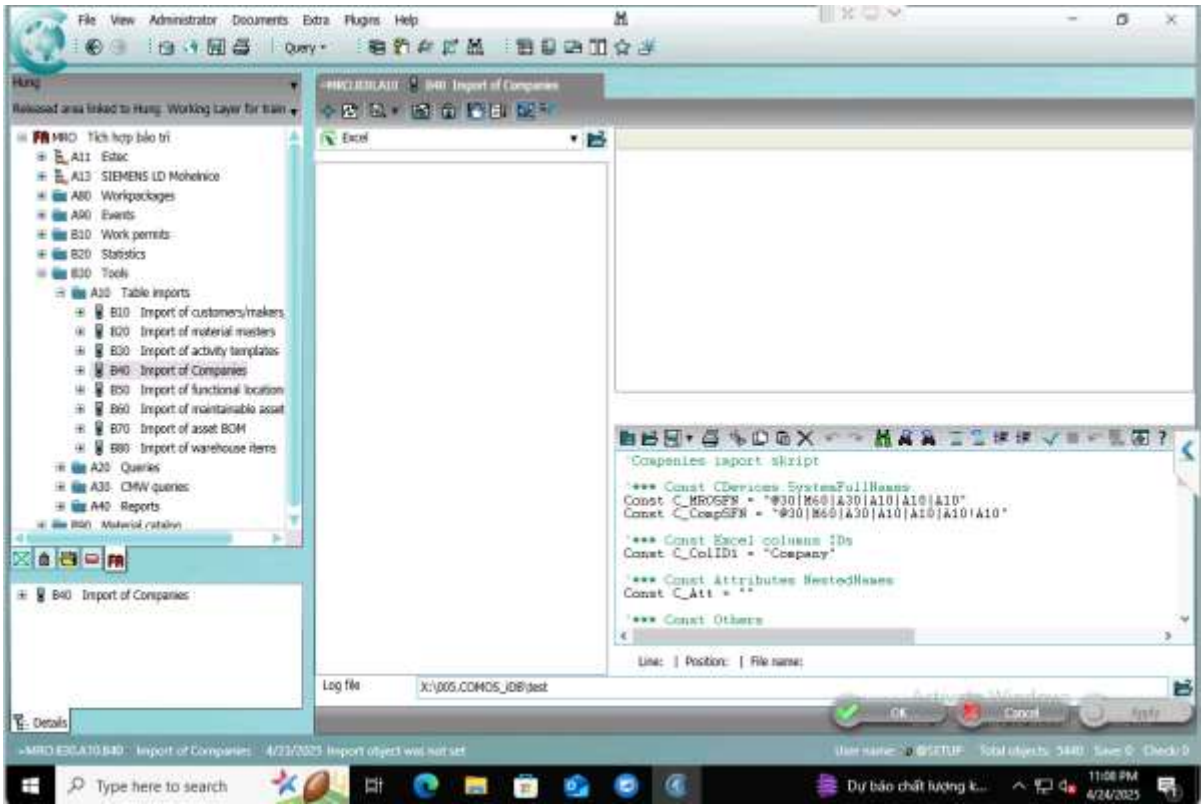


Figure 4.20: Import file Excel

After the imported excel file is available, select the import sheet of the excel file corresponding to the tool in the A10 Table import section (if you select the excel file column that does not match the import tool, the imported data will cause an error)

After clicking on the imported excel sheet, the data of the sheet will appear ➔ you can click the tick button to import the excel data

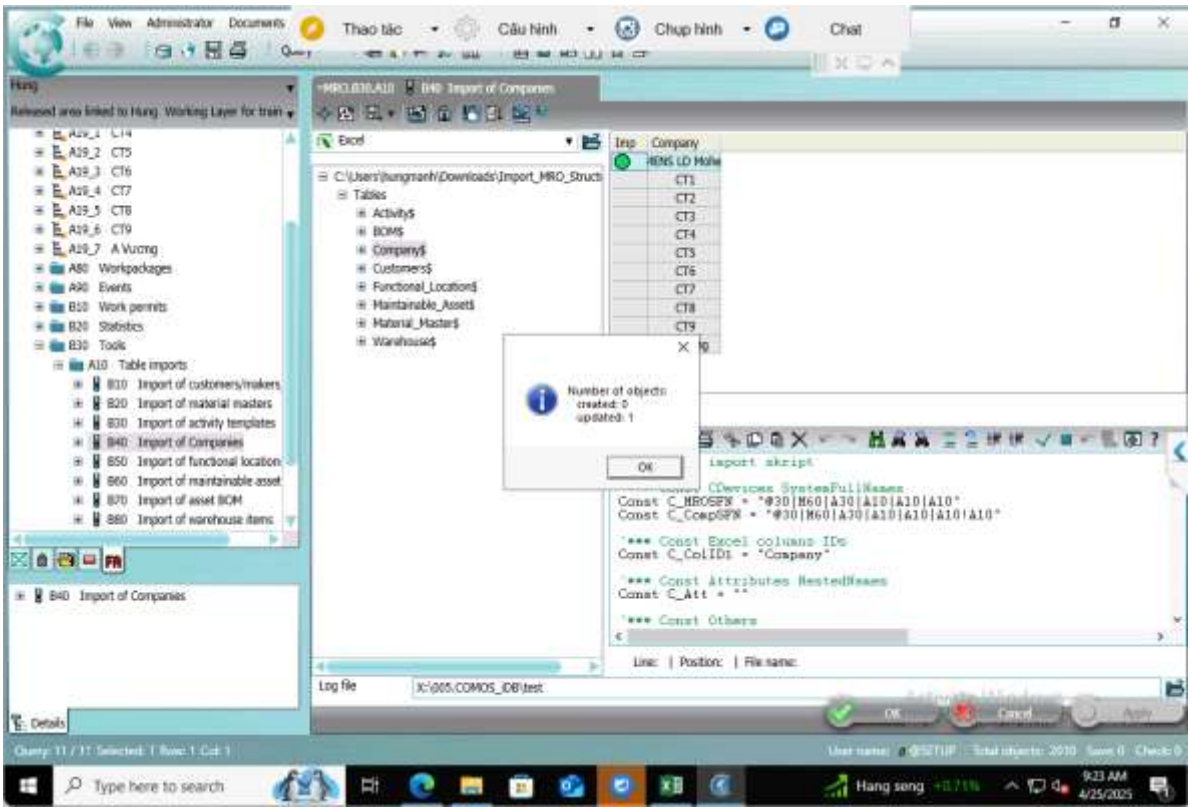


Figure 4.21: Data corresponding to the created import sheet

After repeating the import of each sheet of the corresponding excel file with COMOS's import tool, we have a directory tree.

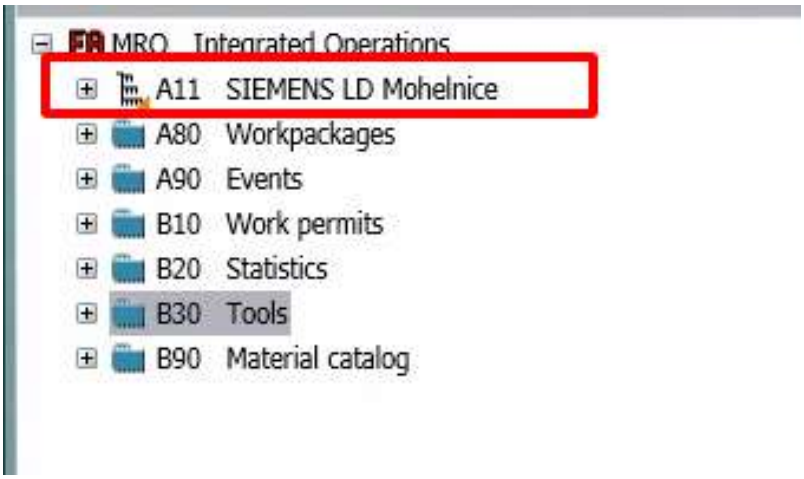


Figure 4.22: MRO directory tree object

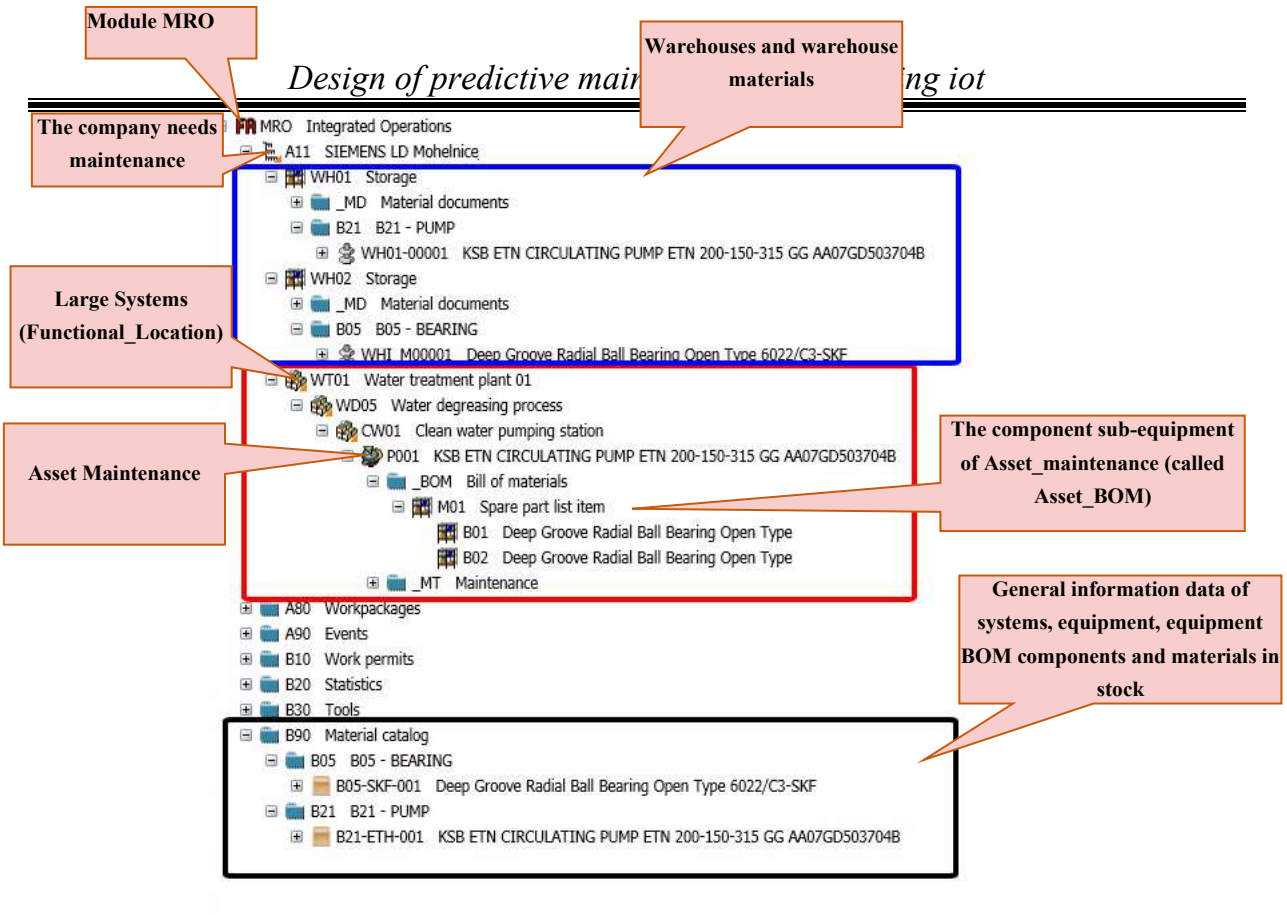


Figure 4.23: MRO directory tree object

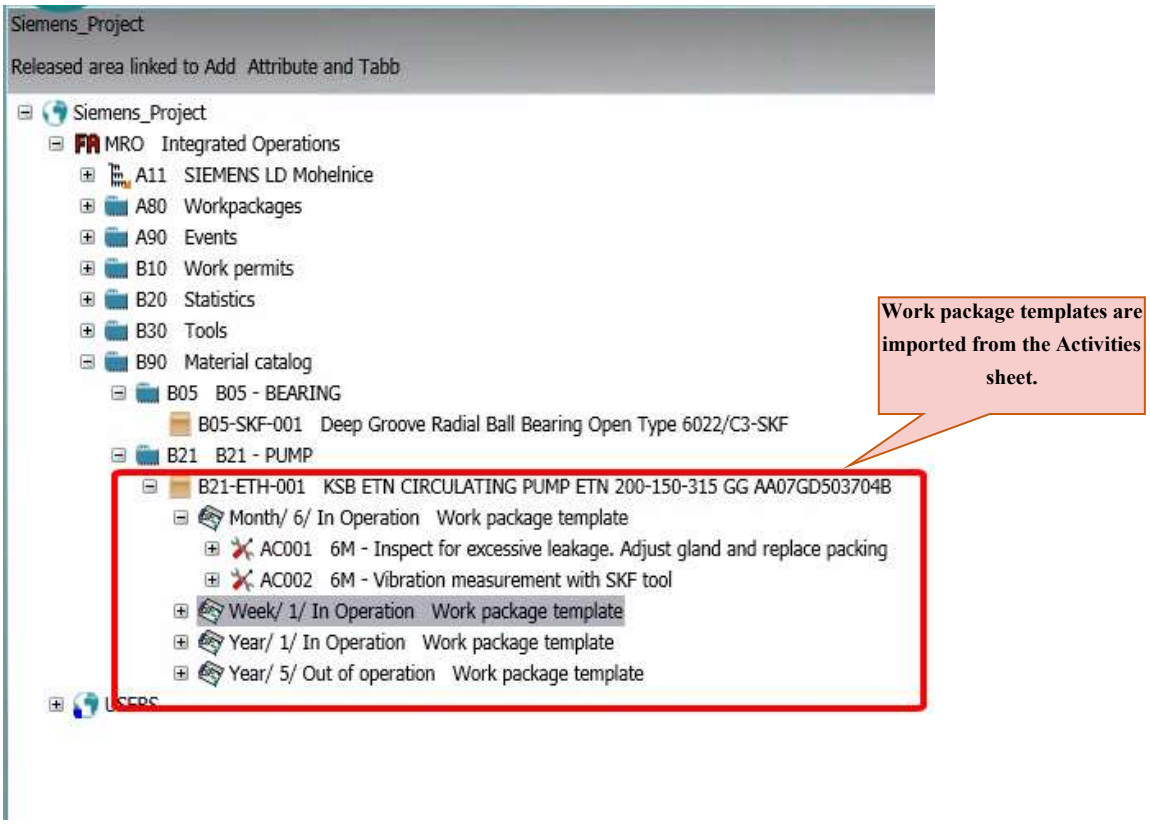


Figure 4.24: MRO directory tree object

4.4 Using Object Debugger[17]

Feature Description: used to determine the ID of an object or from the ID of an existing object find its position in the device tree.

4.4.1 Use the Object Debugger to search for the unique identifier (UID)

Feature Description: Each object (such as systems, devices, BOM components of devices, material warehouse, etc.) has a unique identifier (UID) that is categorized into a specific object group. This UID allows the object's attributes to be updated via the COMOS API and enables data synchronization through CMW.

Each object in COMOS is uniquely identified by a UID in the format U:systemtype:systemuid.

First, open the Extra menu ➔ Object Debugger window.

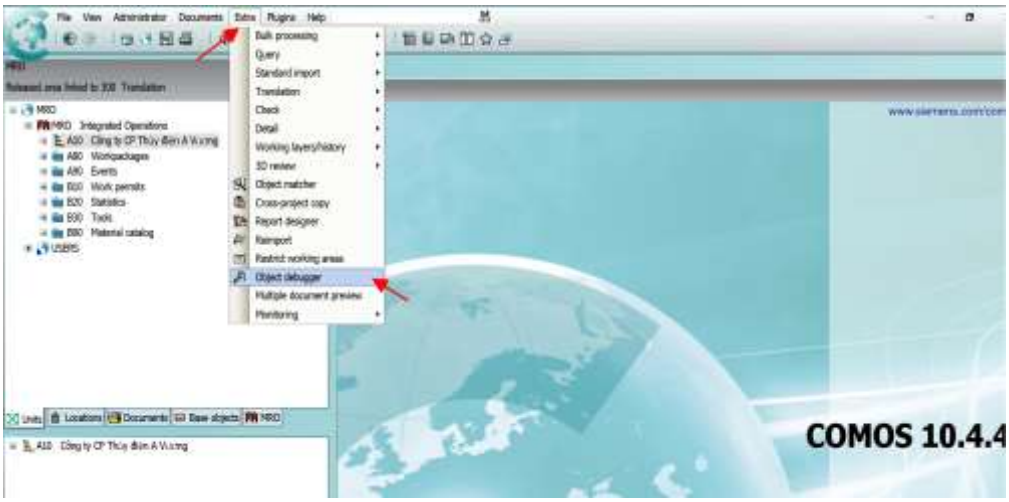


Figure 4.25: Open the Extra menu

Then, the Object Debugger window will appear.

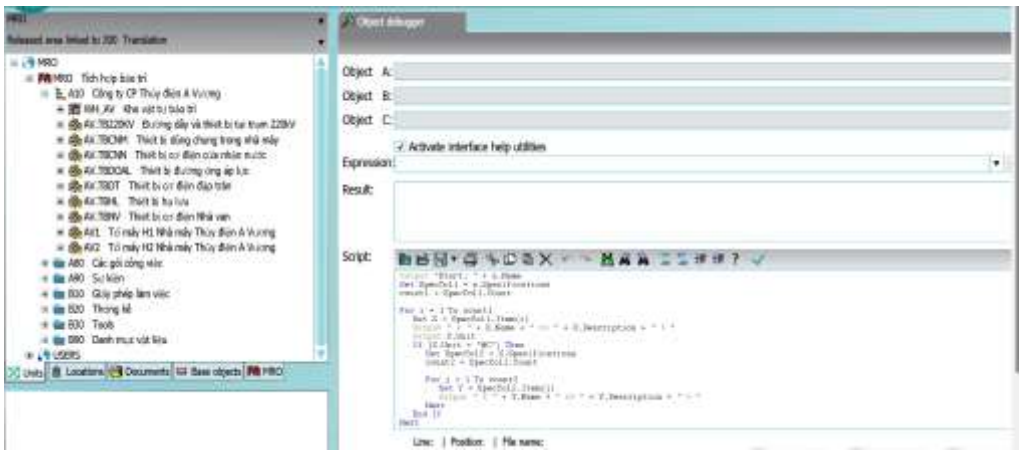


Figure 4.26: Object Debugger window

To obtain the systemuid of any object in COMOS, drag and drop the object into Object A and execute the following command.

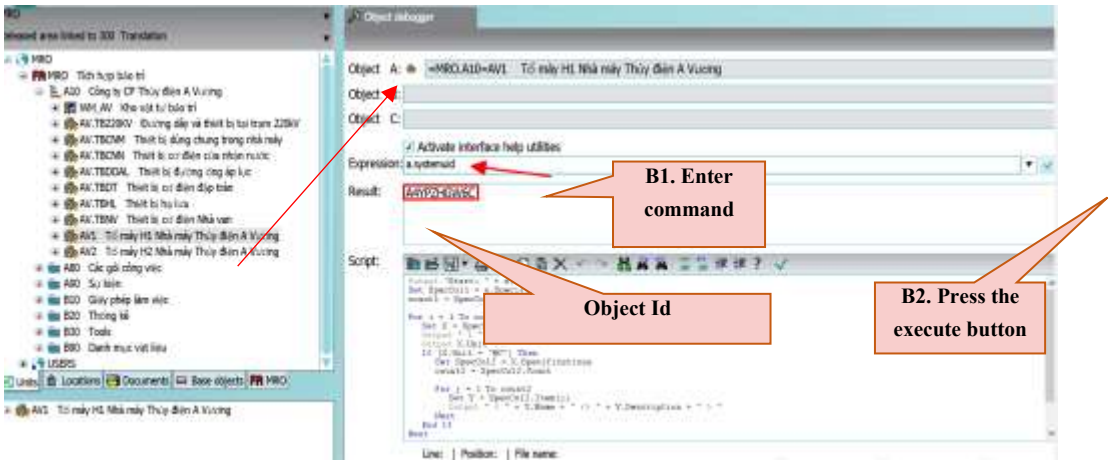


Figure 4.27: UID of the Object

To obtain the systemtype of any object in COMOS, drag and drop the object into Object A and execute the following command.

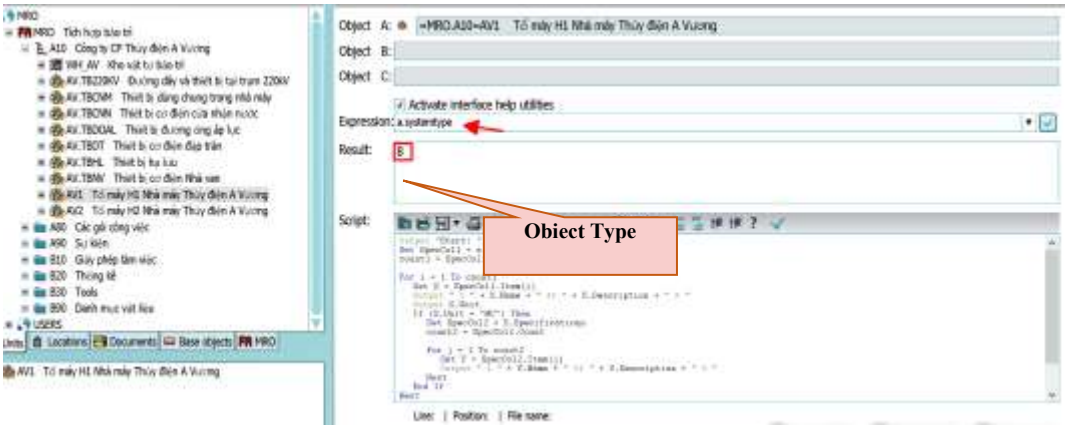


Figure 4.28: Type of the Object

The resulting example UID of the object is: **U:8:A4YPZHGW6C**.

4.4.2 Find Object in directory tree when given ID code

Feature Description: Using a given UID, you can search for the corresponding object information in the project tree by executing a script in the Object Debugger.

Clear the entire script as shown in the image.

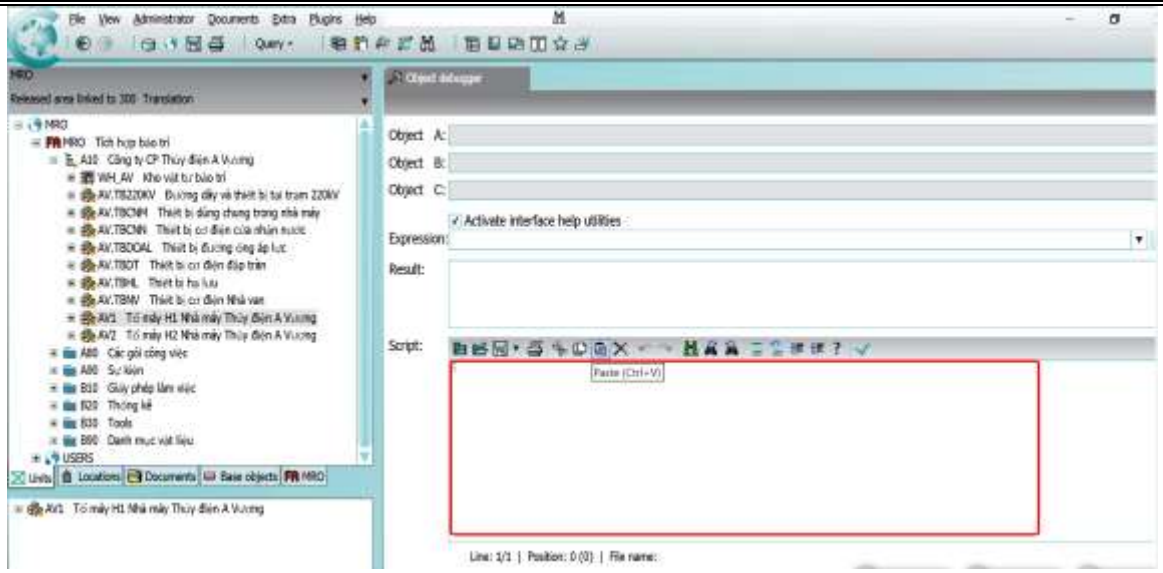


Figure 4.29: Delete the script as shown

Copy the following code into the blank space, using a sample UID as an example.

U:8:A4YPZHGW6C

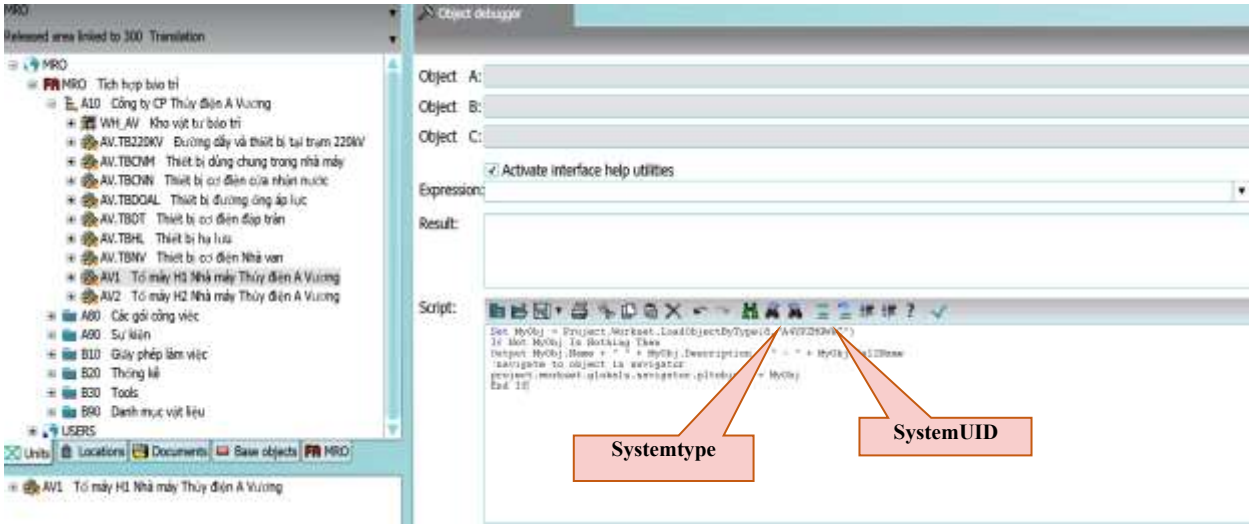


Figure 4.30: Copy the code into the following space

Execute the command, and the system will automatically navigate to the object with that UID.

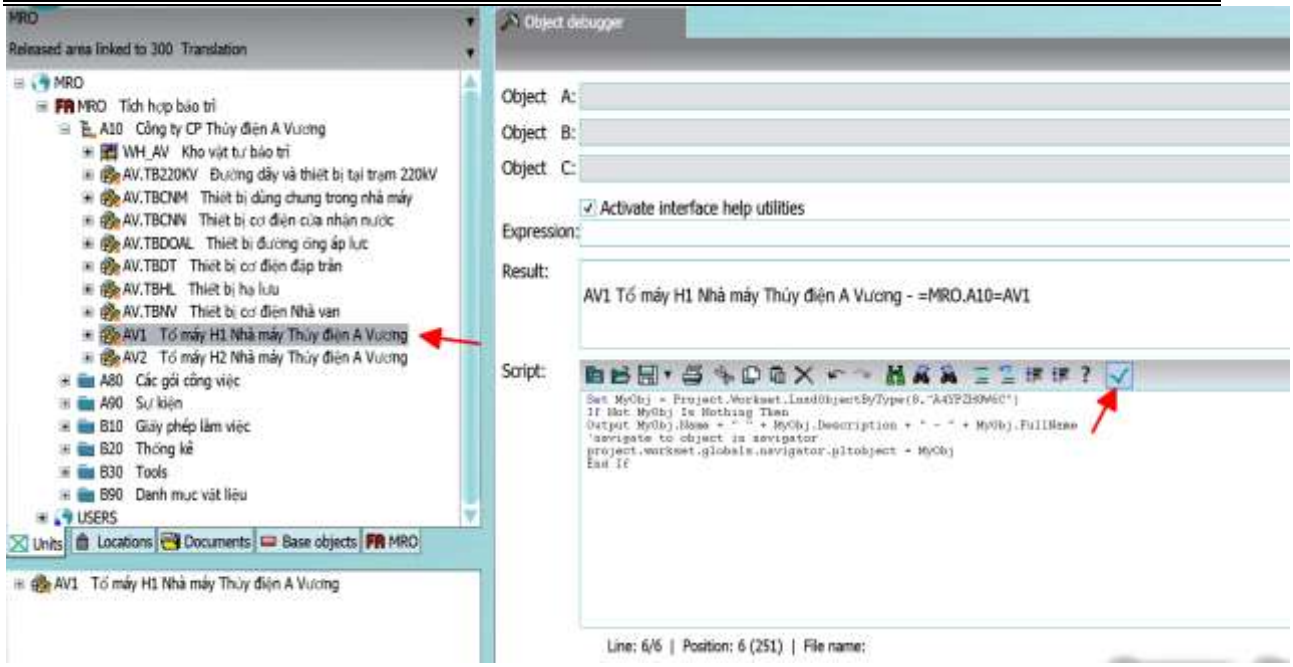


Figure 4.31: Execute command that object

4.4.3 To retrieve the NestedName of any attribute

Description: Each object has multiple attributes, and each attribute has its own unique identifier called the NestedName of the attribute.

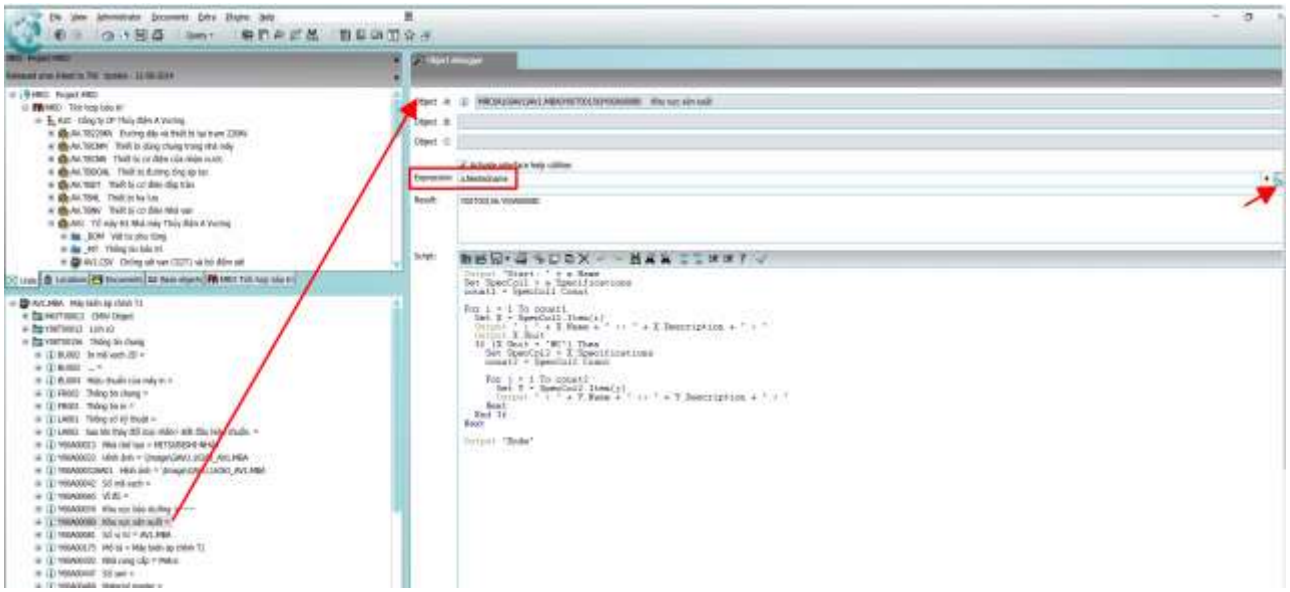


Figure 4.32: Return NestedName

⇨ The **r**eturned NestedName of the attribute is:

Y00T00156.Y00A00080

4.5 Custom tabs and properties in Base Project

4.5.1 Navigate Object to Base object in base Project

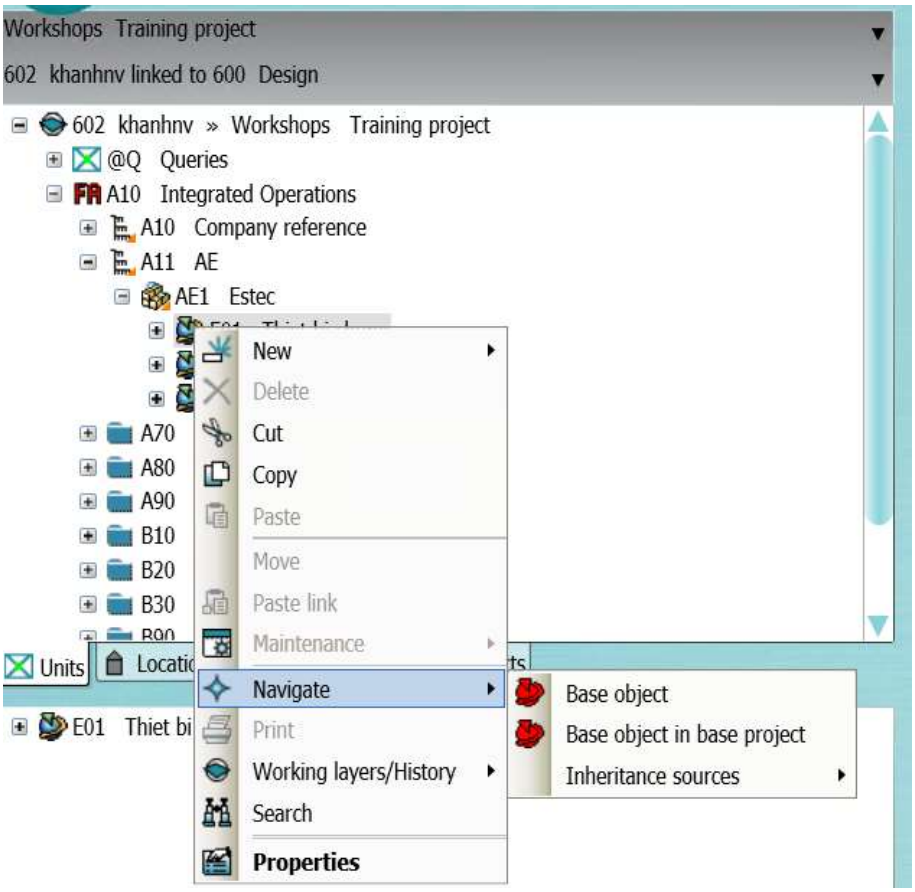


Figure 4.33: Click navigate then select base object in base Project

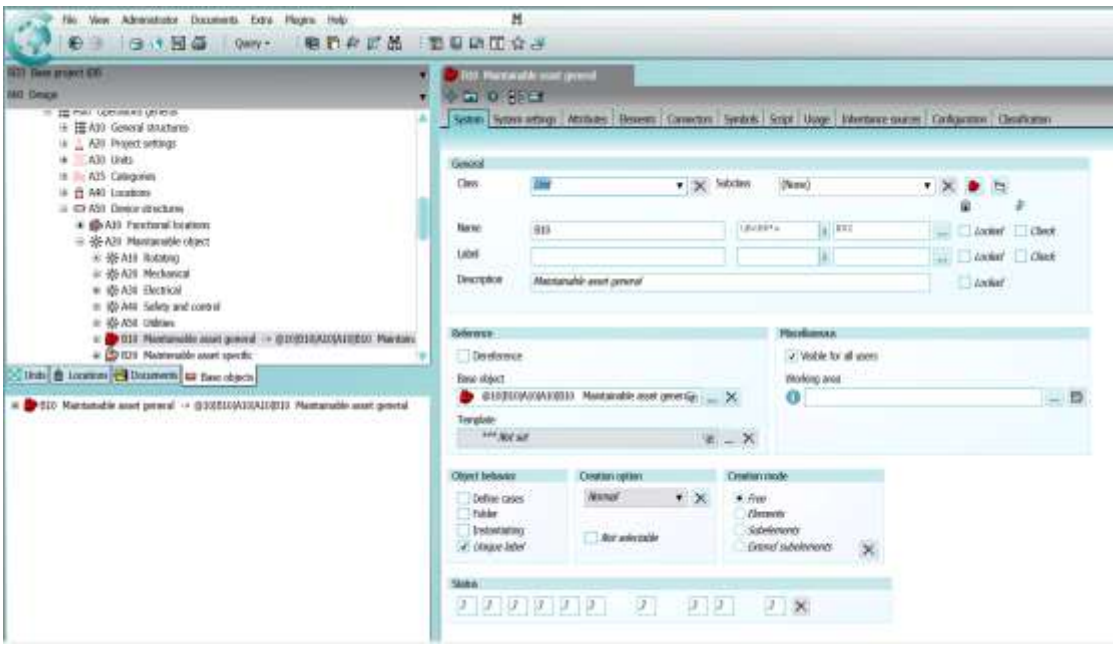


Figure 4.34: Interface in base Project

4.5.2 Customize tabs and properties to display

Purpose: To customize the interface by adding attribute tabs and new attributes ➔ thereby enhancing the information of equipment or adding extended functionalities.

Add a new WL (Working Layer) in the SO1 project (Base project located in Base objects to avoid errors or conflicts with other WLs).

Select the Engineering project you want to modify.

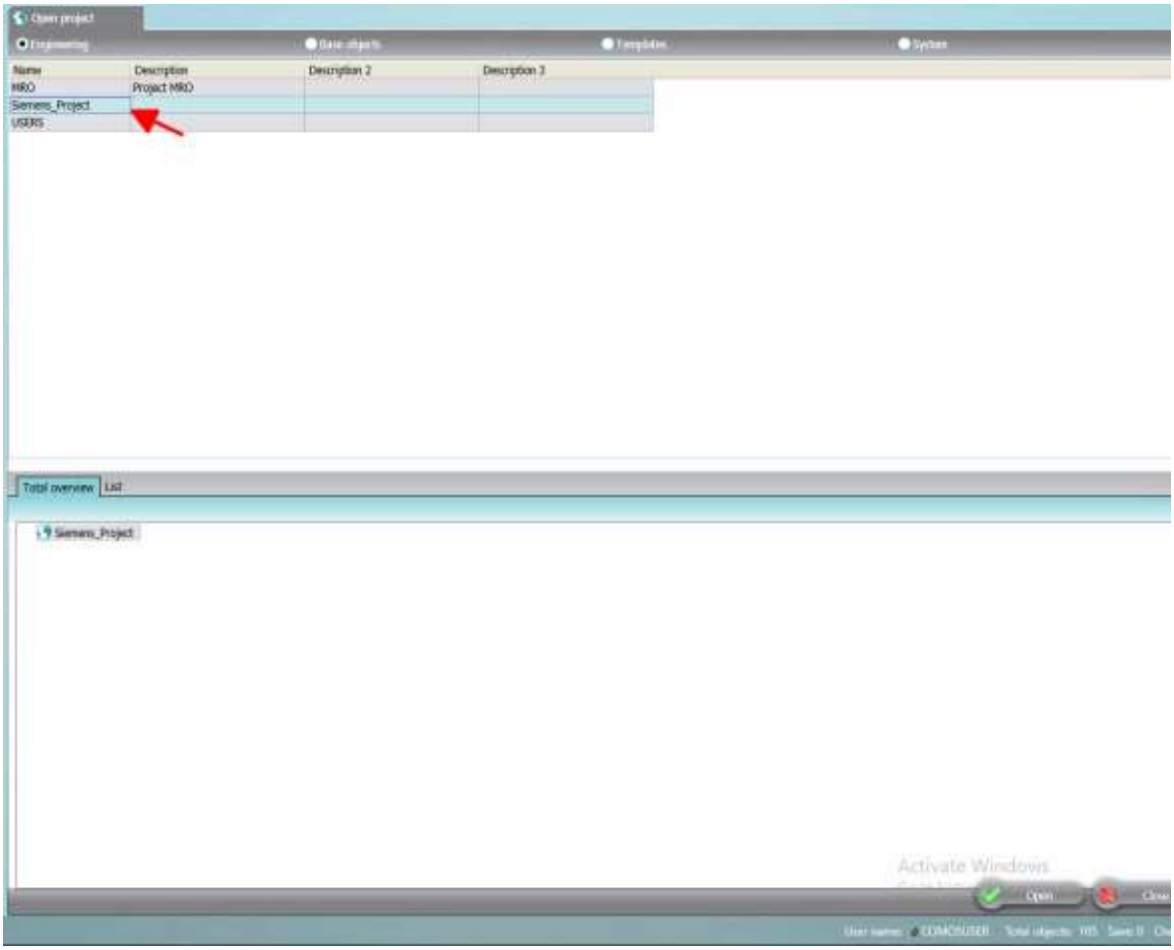


Figure 4.35: Select Project Data.

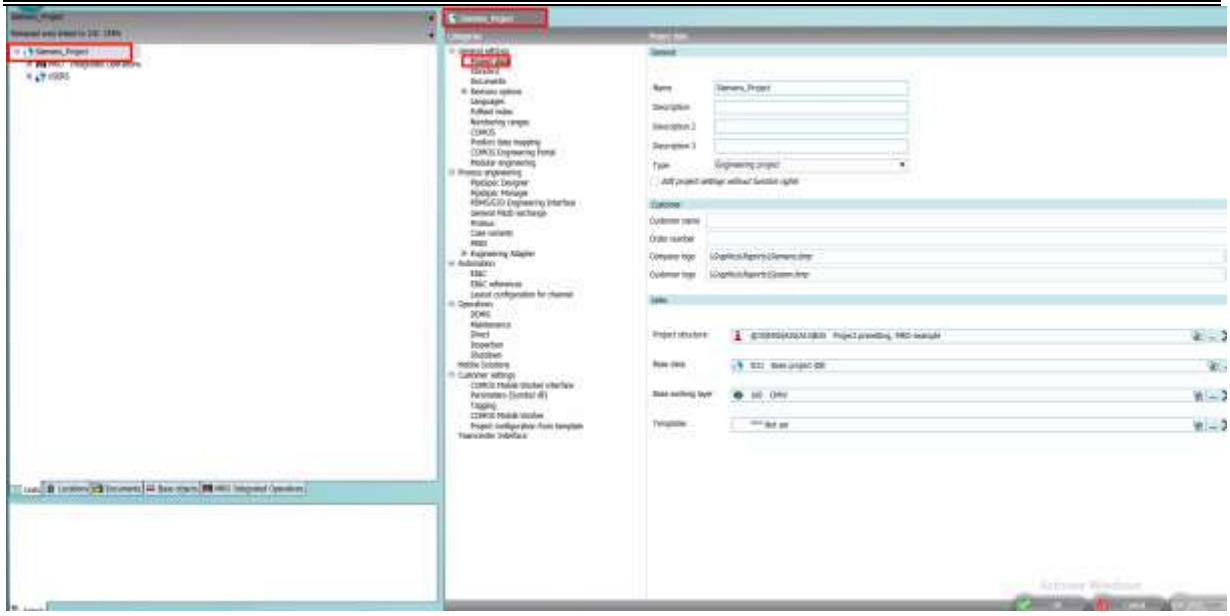


Figure 4.36: Point to the WL in the Base project that was just created.

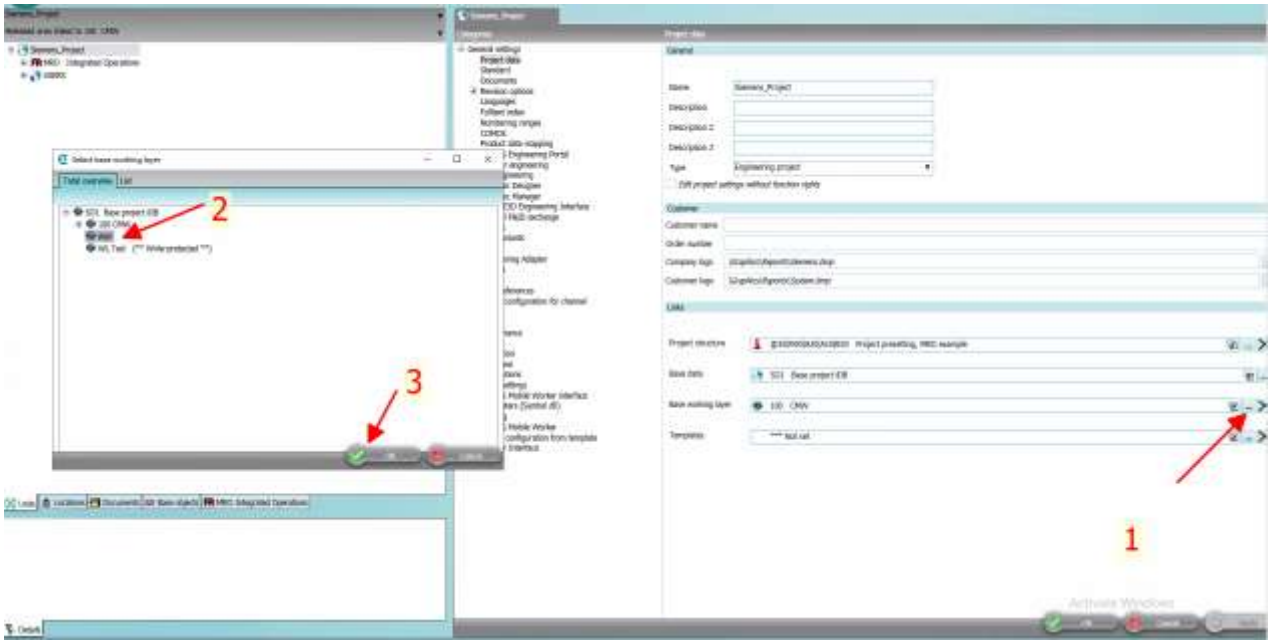


Figure 4.37: After configuration

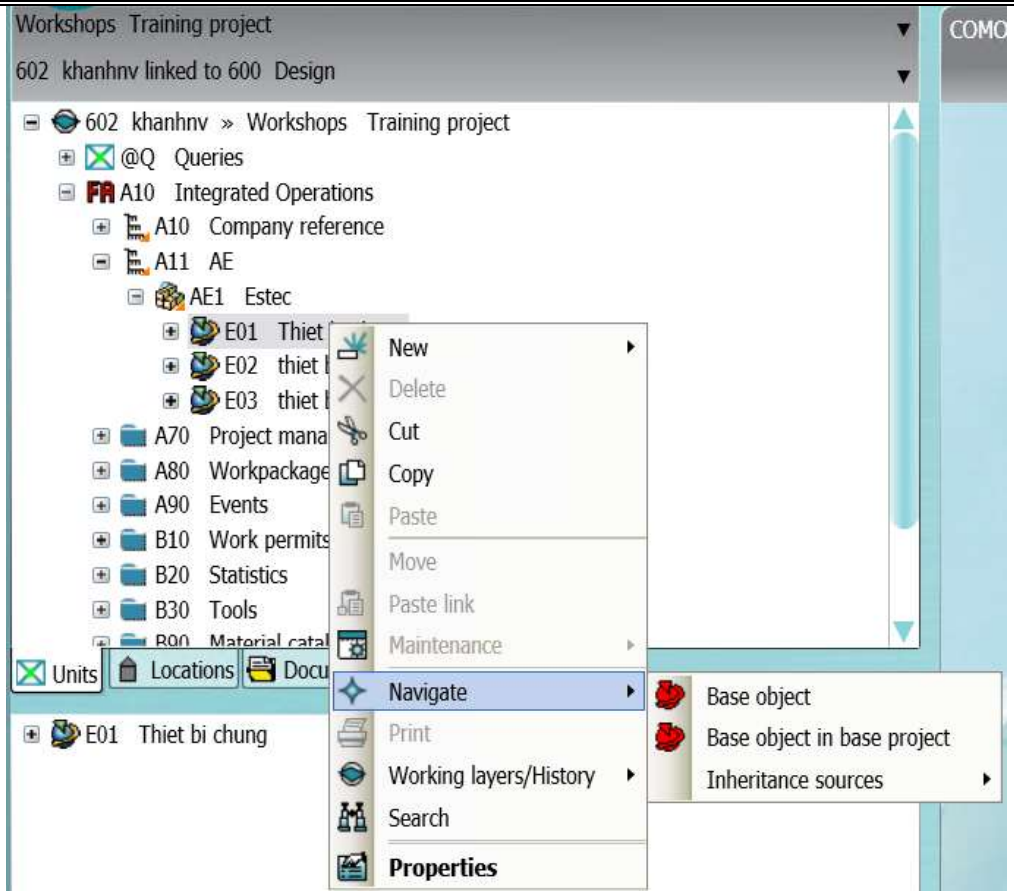


Figure 4.38: Navigate it to the Base Objects section.

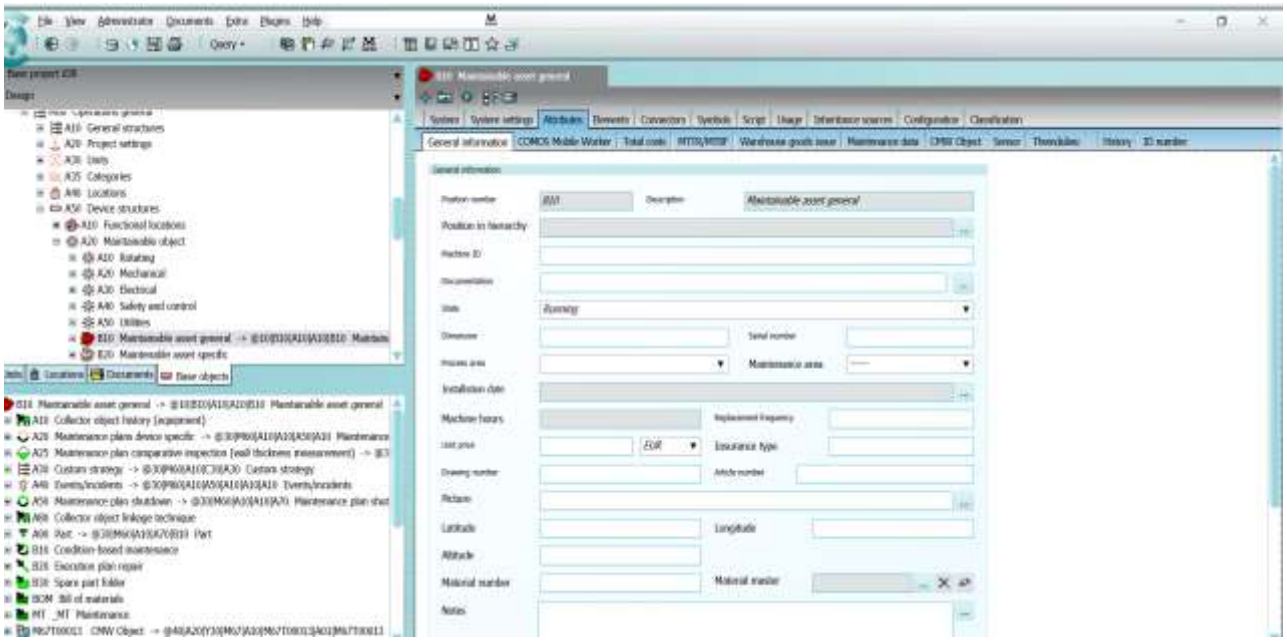


Figure 4.39: Select the Attributes Tab

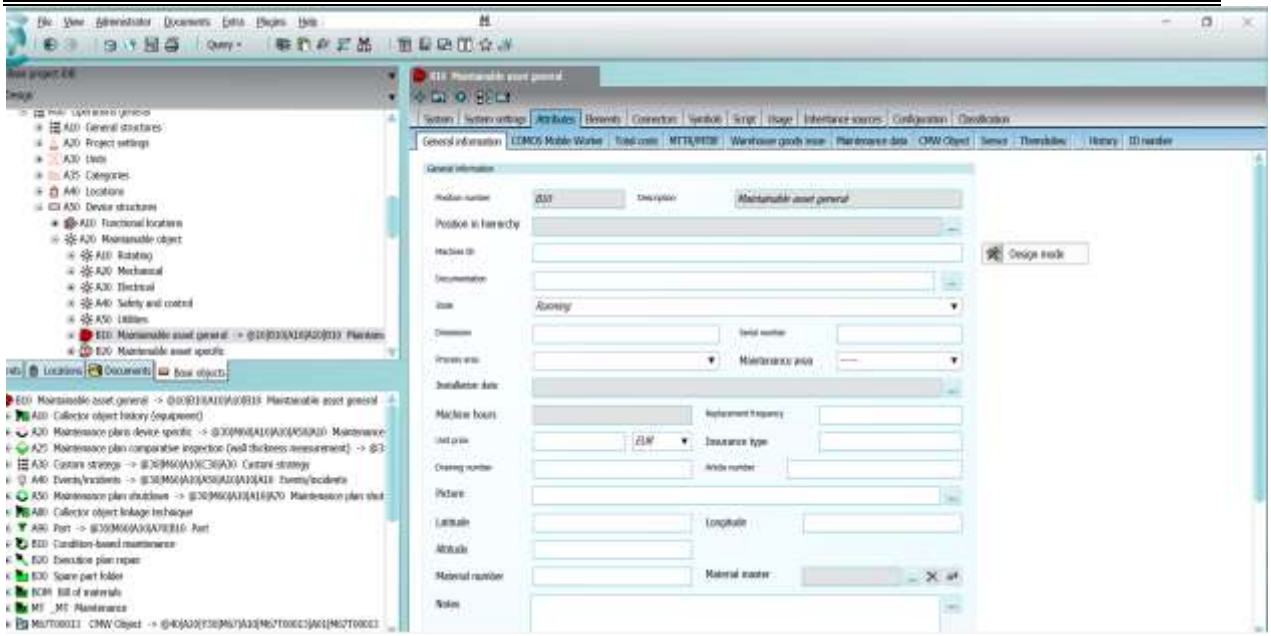


Figure 4.40: Enable Design mode

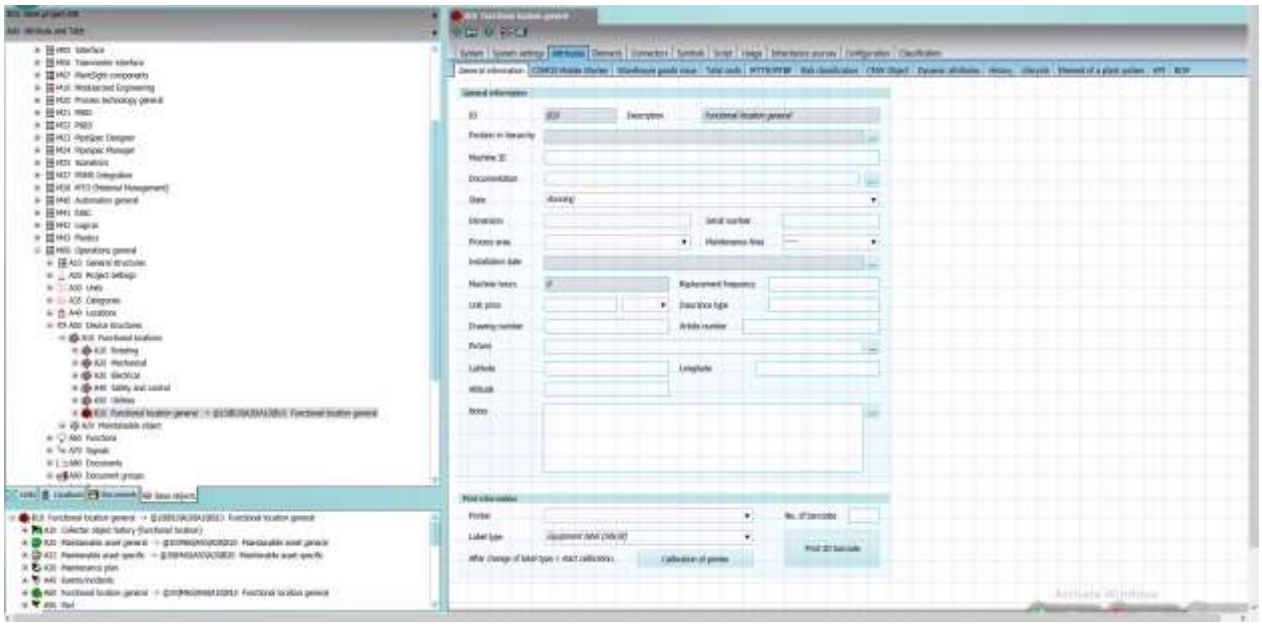


Figure 4.41: Design mode

Add new tab

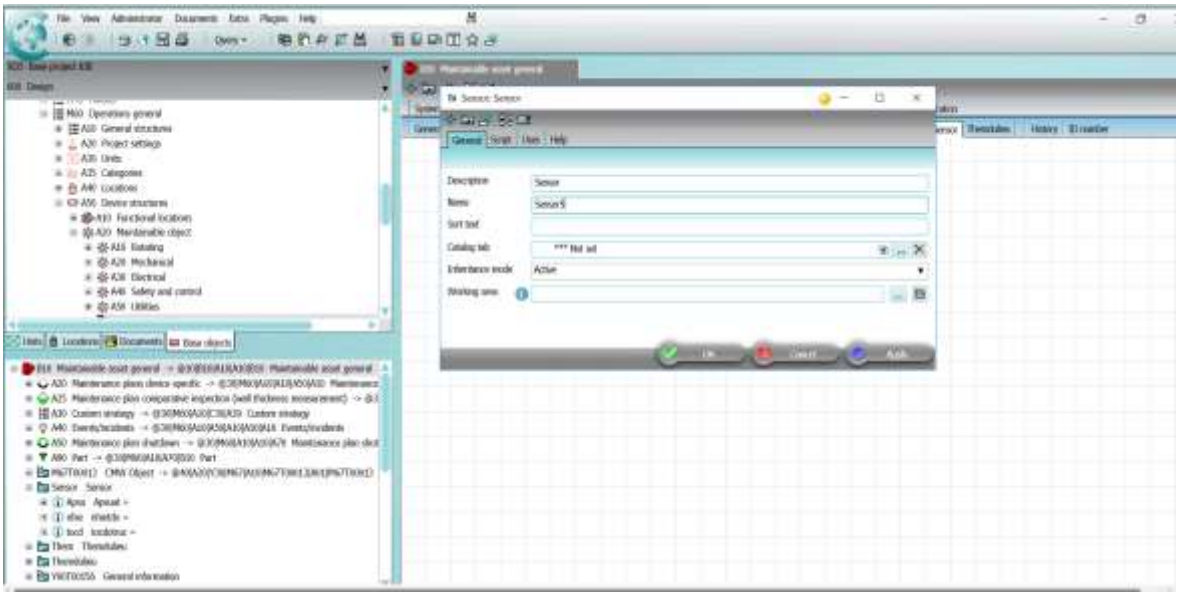


Figure 4.42: New Tab Parameters

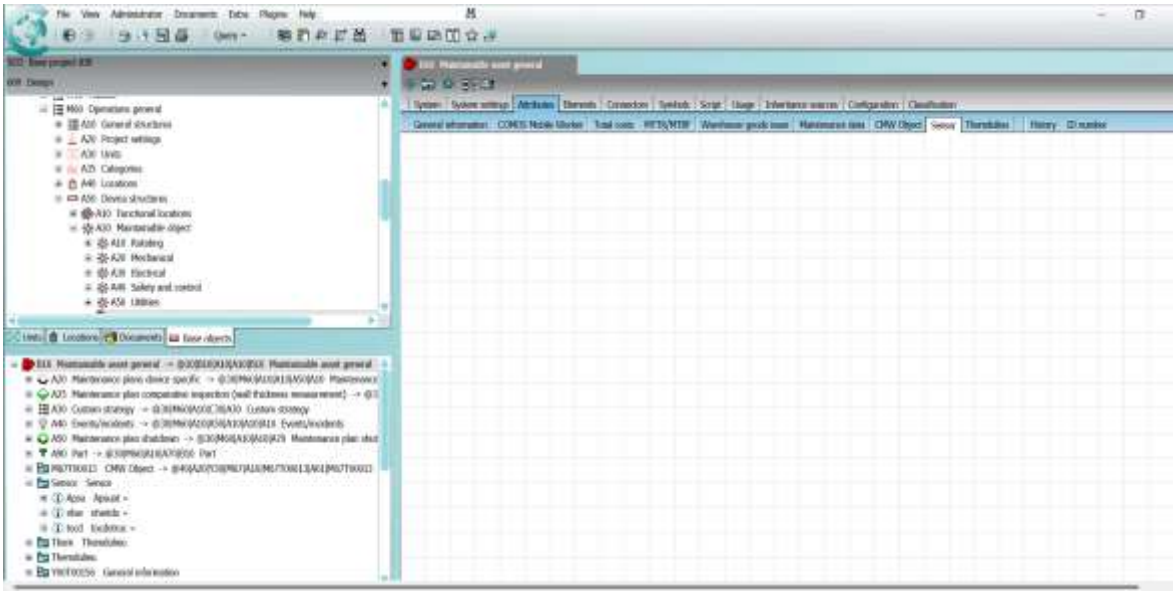


Figure 4.43: New Tab

Add new atrribute

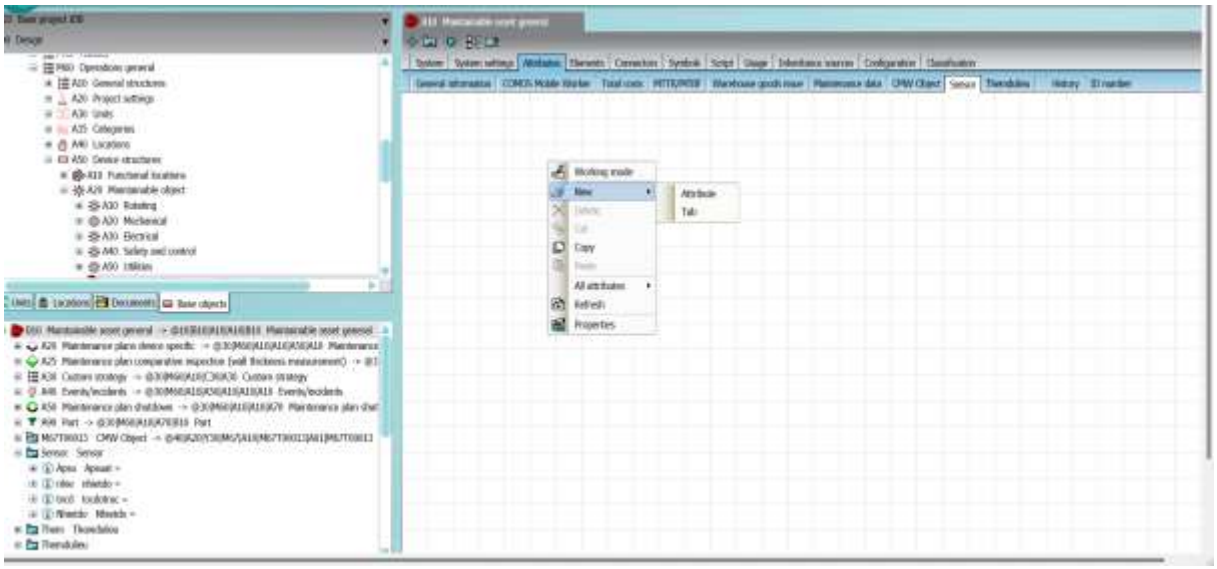


Figure 4.44: Select New then select Attribute

The property information window appears, you need to adjust the description name (Description) and identification name (Name), then click Apply to save the information.

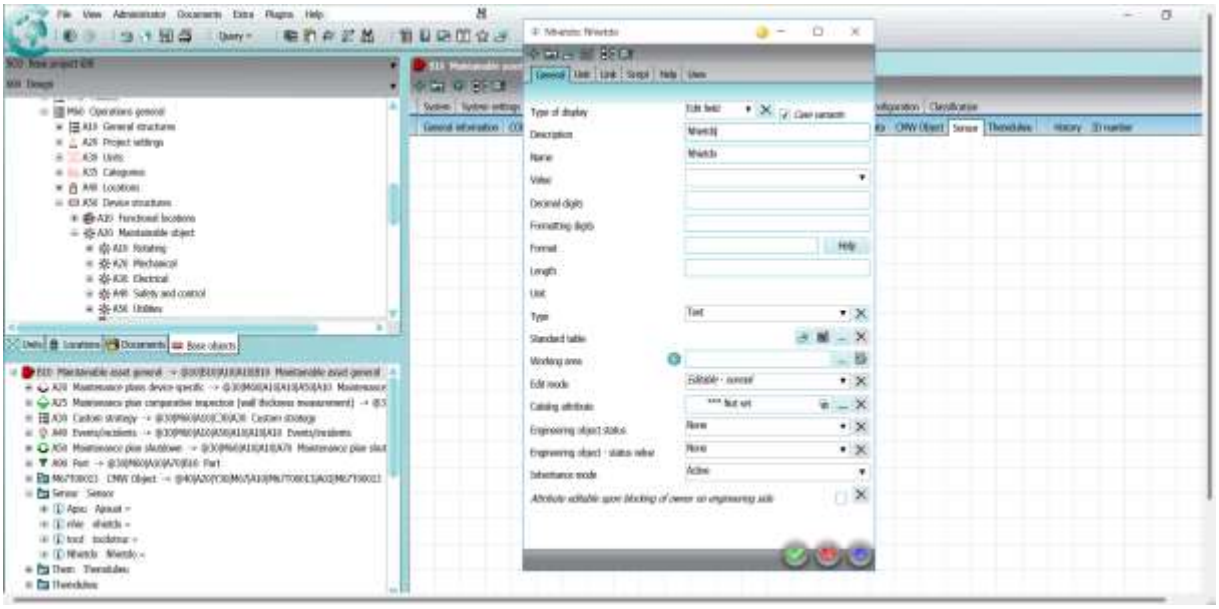


Figure 4.45: Attribute information window

So the new attribute has been created, click Apply to save and finish adding the attribute to that tab.

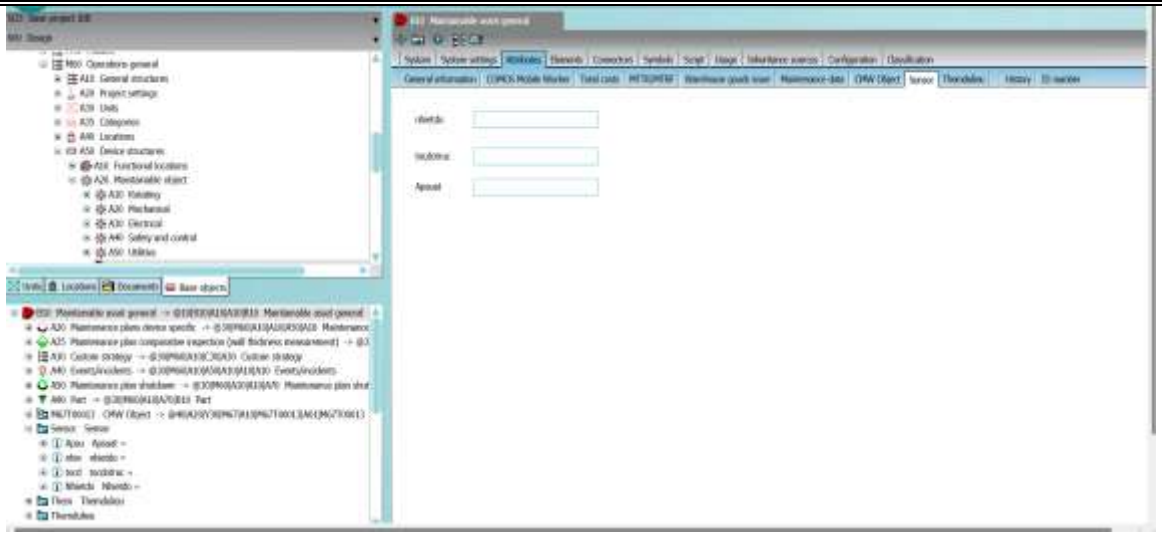


Figure 4.46: Switch to Working mode

4.6 Use API to get data

Introducing COMOS API

Main COMOS APIs:

- API Documents
- API Projects
- API Sessions

Figure 4.47: Main APTs

4.6.1 API Sessions

API Sessions: Access sessions of the COMOS REST API initiate and terminate a data synchronization session using a SessionId returned as a string. It includes two main APIs: login and logout.

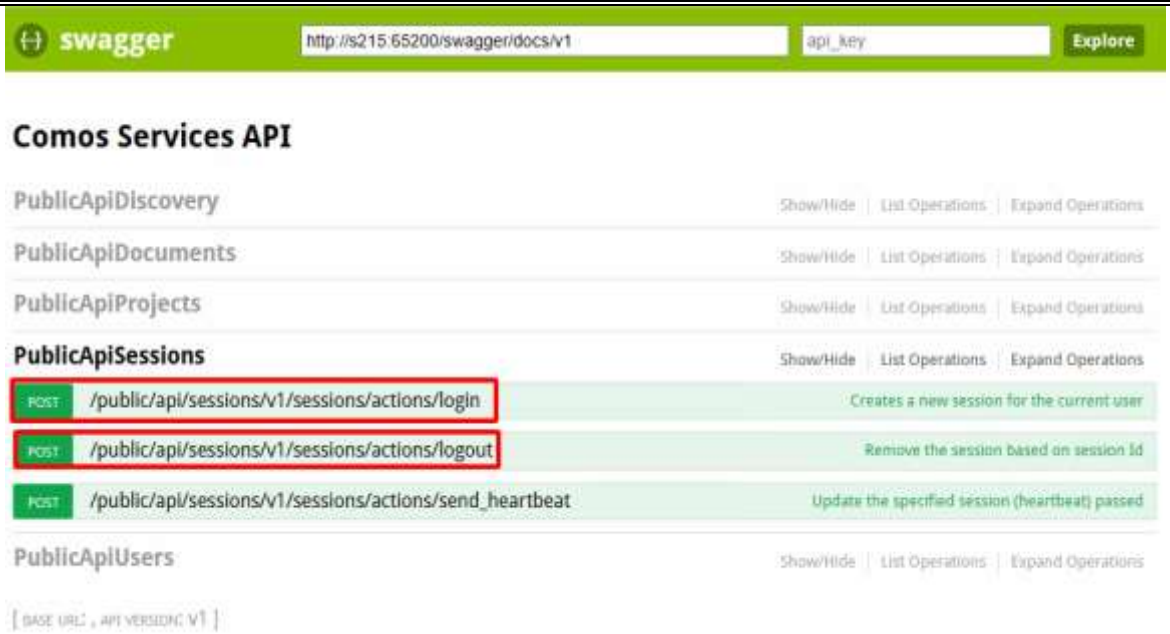


Figure 4.48: API Sessions

a. API login

Purpose: To obtain a session ID to start a working session with the API

How to do it: Go to API Sessions ➔ select the login API. Then click on "Try it out" to get the session ID.

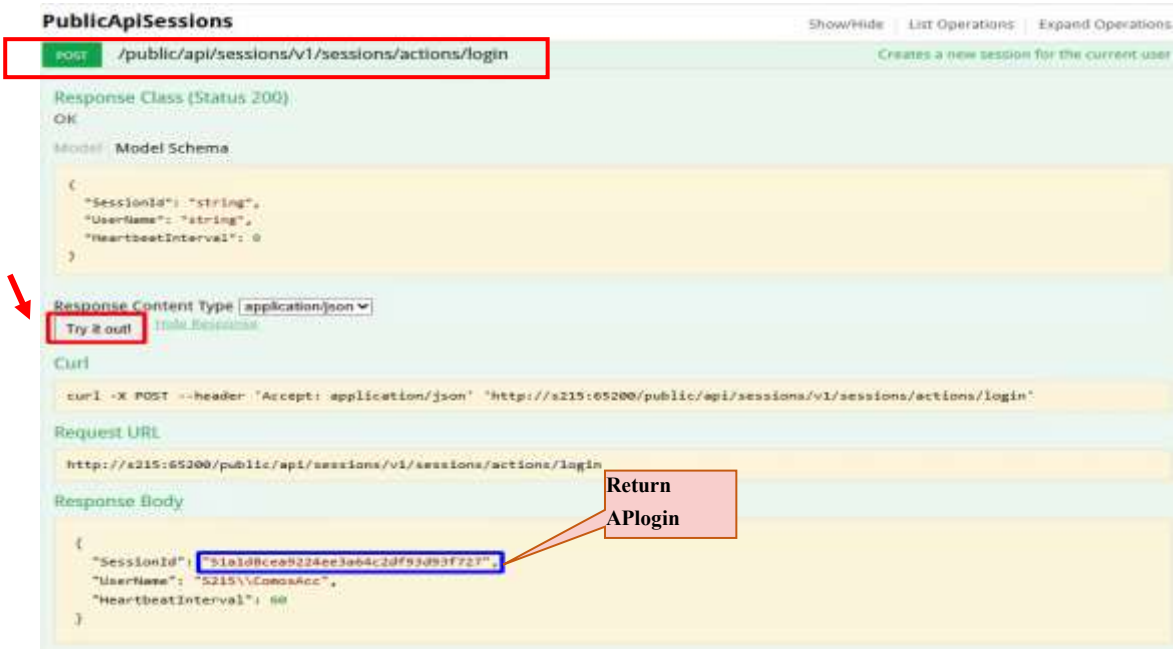


Figure 4.49: How to get Session ID

b. API logout

Purpose: To exit the session session ID login

How to do: go to API Sessions ➔ select API logout

Enter Session ID logout

Then select try it out to end the session

4.6.2 API Projects

API Projects: through the UIDs (taken from COMOS to form a list) of Projects and each Object of a Project can return data in the form of Json or text strings:

Get full and read Object data from COMOS

Update Object data to COMOS

Create or delete Objects according to the tree structure

Figure 4.50: API Project

a. API to get Object information

Purpose: To get all the attribute information of an Object in COMOS

How to do it:

go to API Project ➔ select API Get the objects

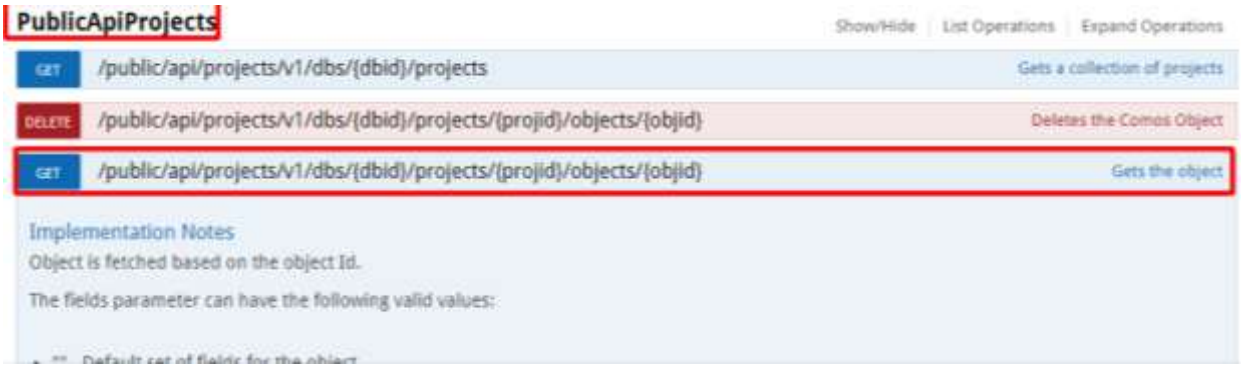


Figure 4.51: Get information Project

At API Get the objects: enter the required parameters and click try it out

In which:

- dbid: is the database key settup when installing COMOS Web
- projid: is the UID of the MRO project being executed
- objid: is the UID of the Object in the corresponding MRO project

Comos-API-Session: Session ID login returned

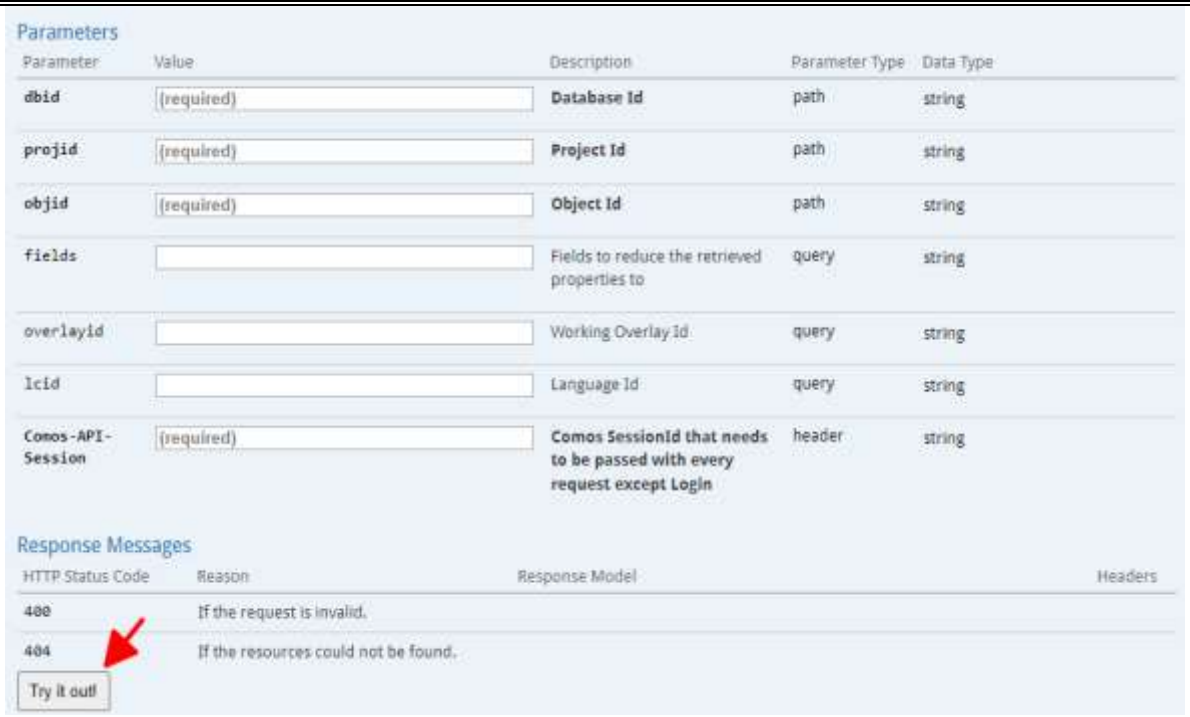


Figure 4.52: Session ID login returned

b. API updates Object information via nestedName

Purpose: To update information of an Object in COMOS

How to do:

go to API Project ➔ select API Update the given object

Figure 4.53: API Update the given object

At API Get the objects: enter the required parameters and click try it out

In which

- dbid: is the database key settup when installing COMOS Web
- projid: is the UID of the MRO project being executed

- objid: is the UID of the Object in the corresponding MRO project
- value: enter the identifier value of the attribute

```
[
{
"NestedName":"string",
"Value":"string",
"LinkObject":{
"UID":"string",
"ProjectUID":"string"
}
]
```

Comos-API-Session: Session ID login returned

4.6.3 API Documents

API Documents: through this API you can download/upload documents from COMOS

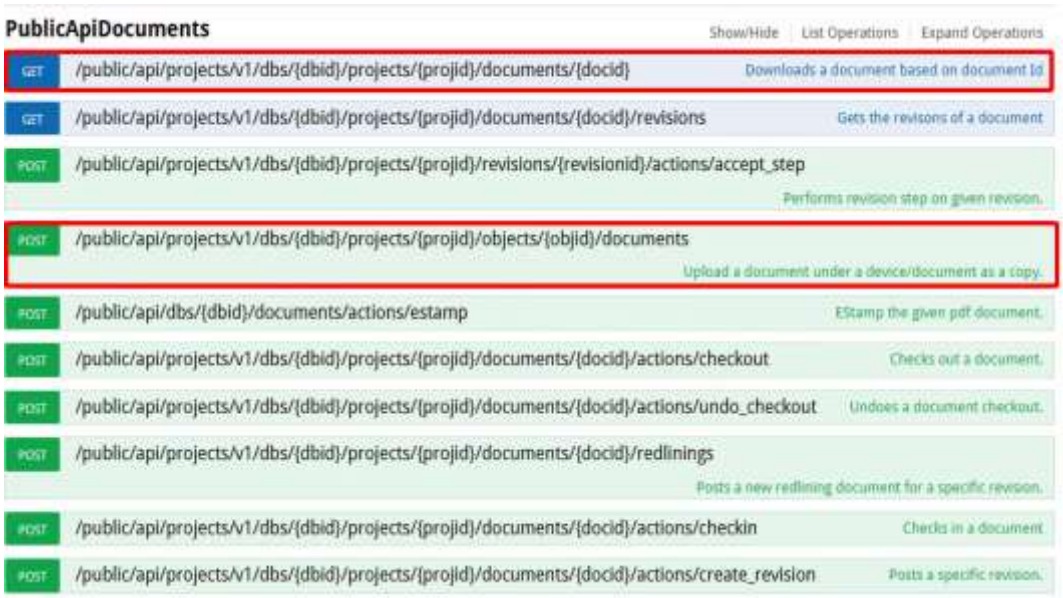


Figure 4.54: API Documents

How to do: enter the required data fields, select the file you want to upload the document to, then click Try it out!

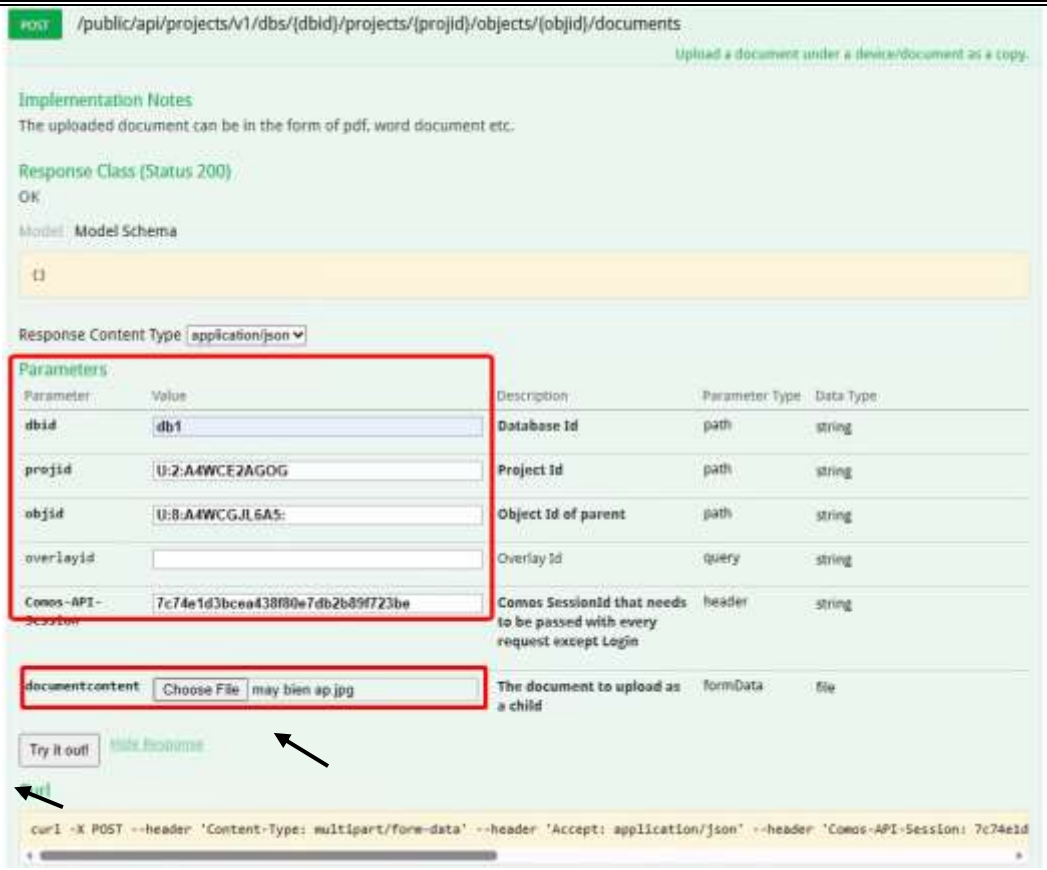


Figure 4.55: Upload documents

CHAPTER 5 MAINTENANCE SOFTWARE DESIGN

The team is developing a software system that integrates real-time monitoring and predictive maintenance functionalities. The interface allows users to input a list of sensors (NodeIds), monitor sensor data in real time, and configure warning thresholds. When a sensor value exceeds the defined threshold, the system automatically sends an alert email to technicians or responsible personnel to promptly handle the issue and prevent serious equipment failures.

The software supports:

- Real-time visualization of sensor data through charts.
- Threshold configuration and warning detection.
- Email alerting when values exceed the defined limits.
- Data communication with OPC UA servers (e.g., Prosys).
- Data uploading, AI model training, and equipment condition prediction.

This system aims to enhance equipment reliability, reduce unexpected downtime, and support proactive maintenance operations, making it suitable for industrial applications in smart manufacturing environments.

5.1 Algorithm flowchart for building CMMS software features

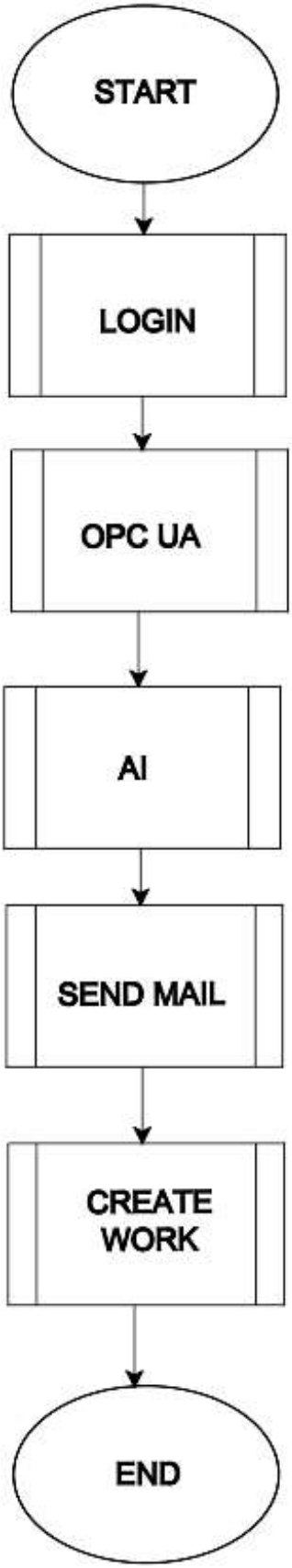


Figure 5.1: Main algorithm flowchart

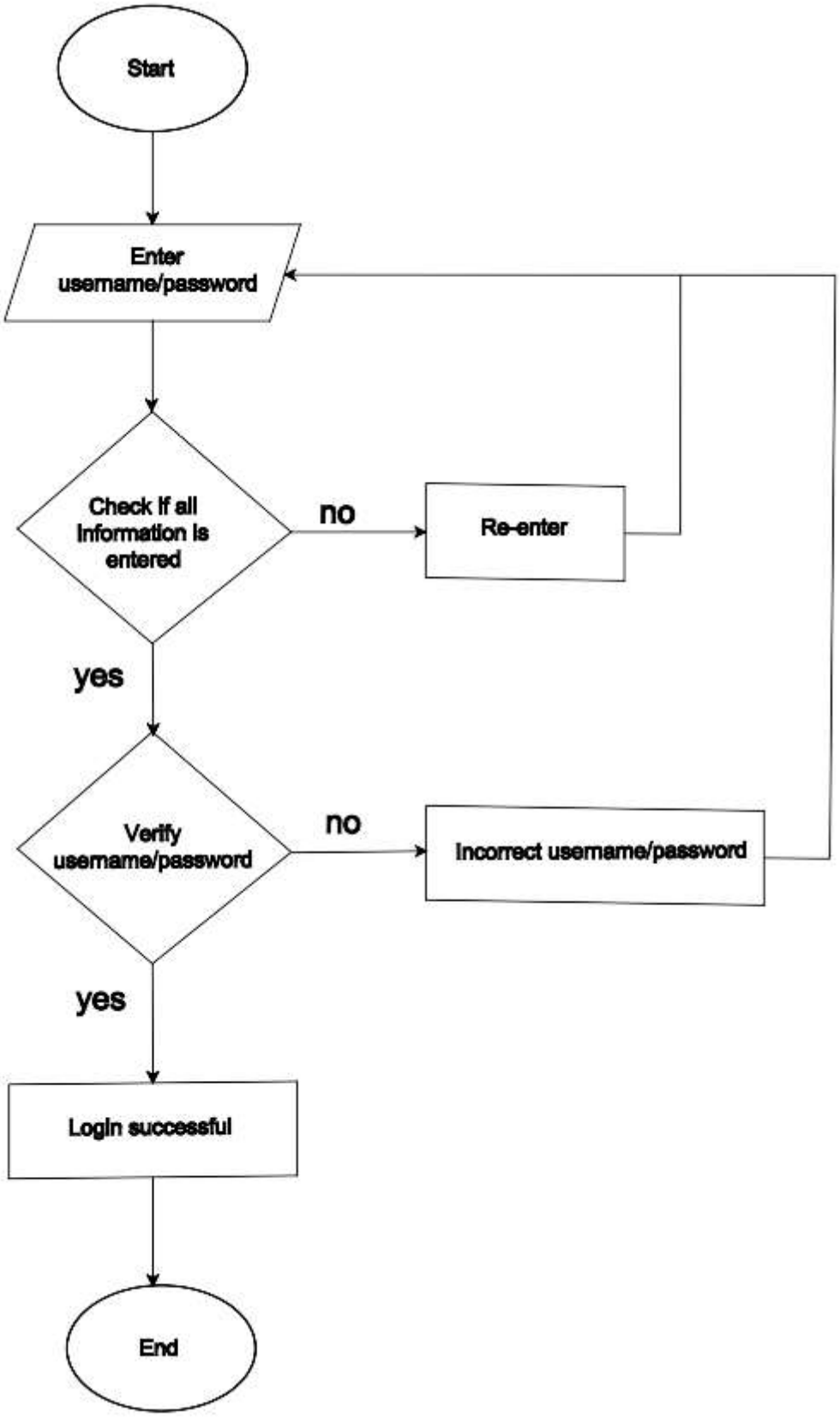


Figure 5.2: Login in the software

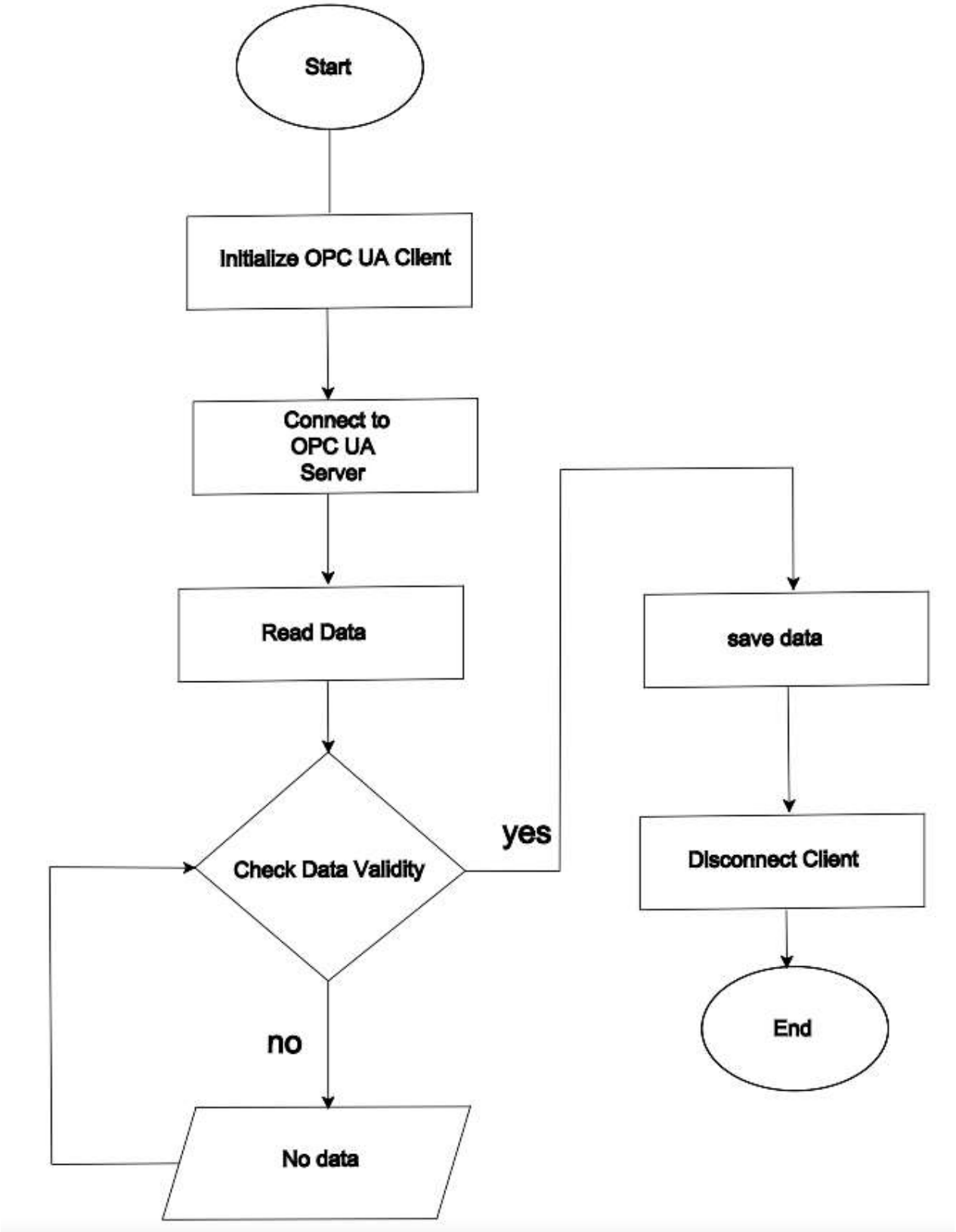


Figure 5.3: Algorithm flowchart for real-time data retrieval

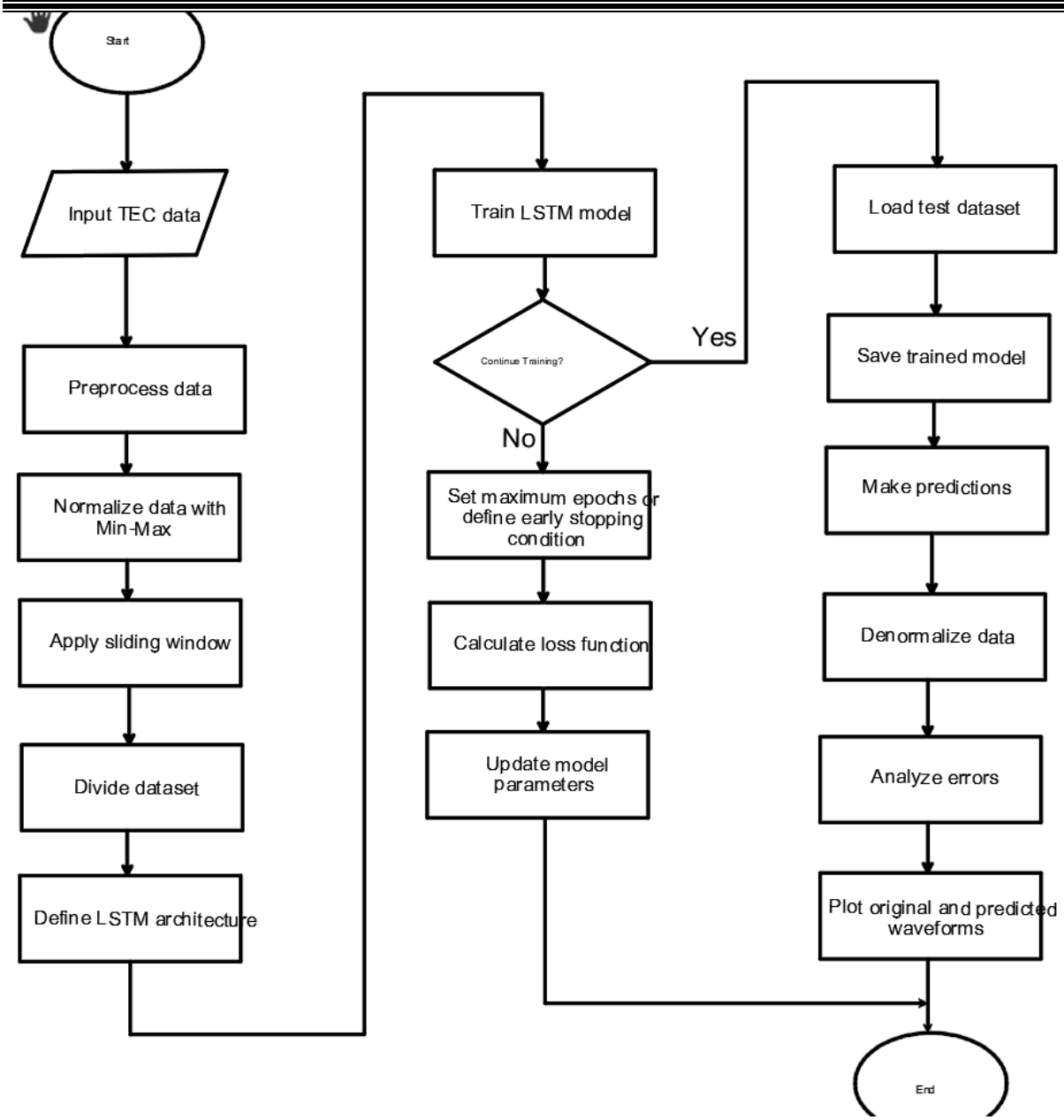


Figure 5.4: Training AI Model

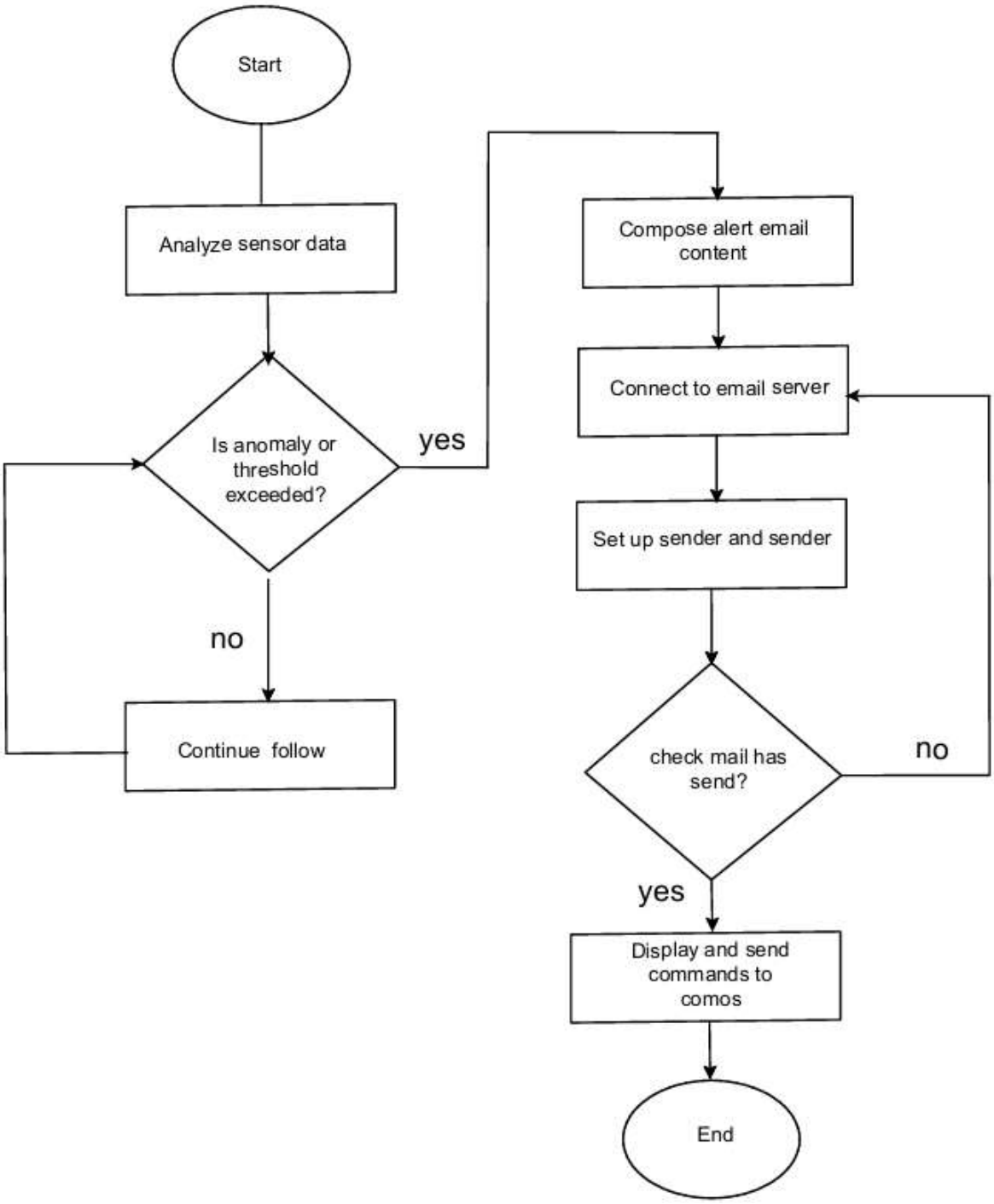


Figure 5.5: Algorithm flowchart for sending alert emails

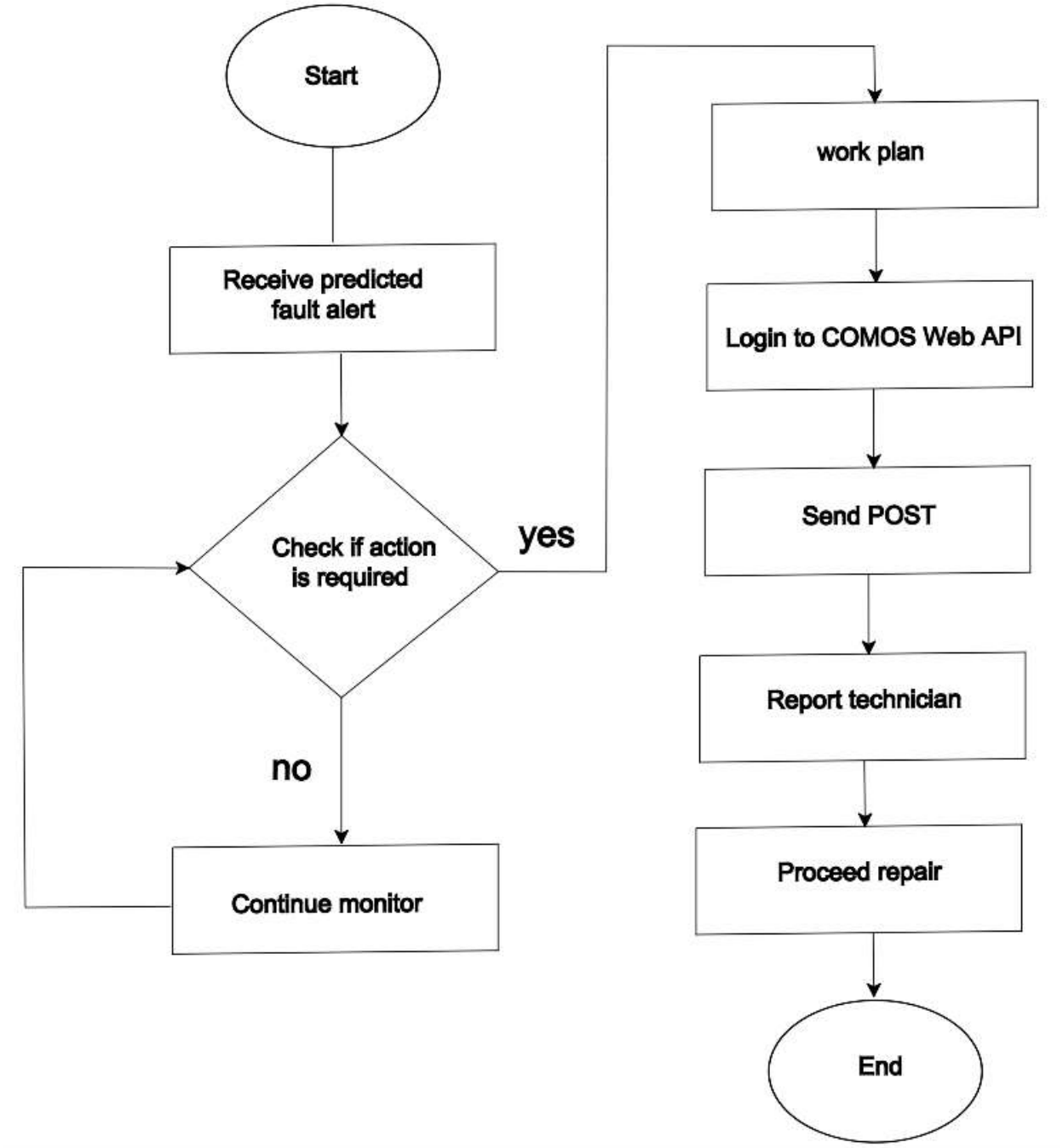


Figure 5.6: Algorithm flowchart for self-generating maintenance commands

5.2 Software interface design

The CMMS software interface includes the following modules:

- Login
- Dashboard
- Device Management

- Maintenance History
- Sensor Upload & Visualize
- Sensor Data
- Maintenance Plan
- Predict Fault

5.5.1 Login

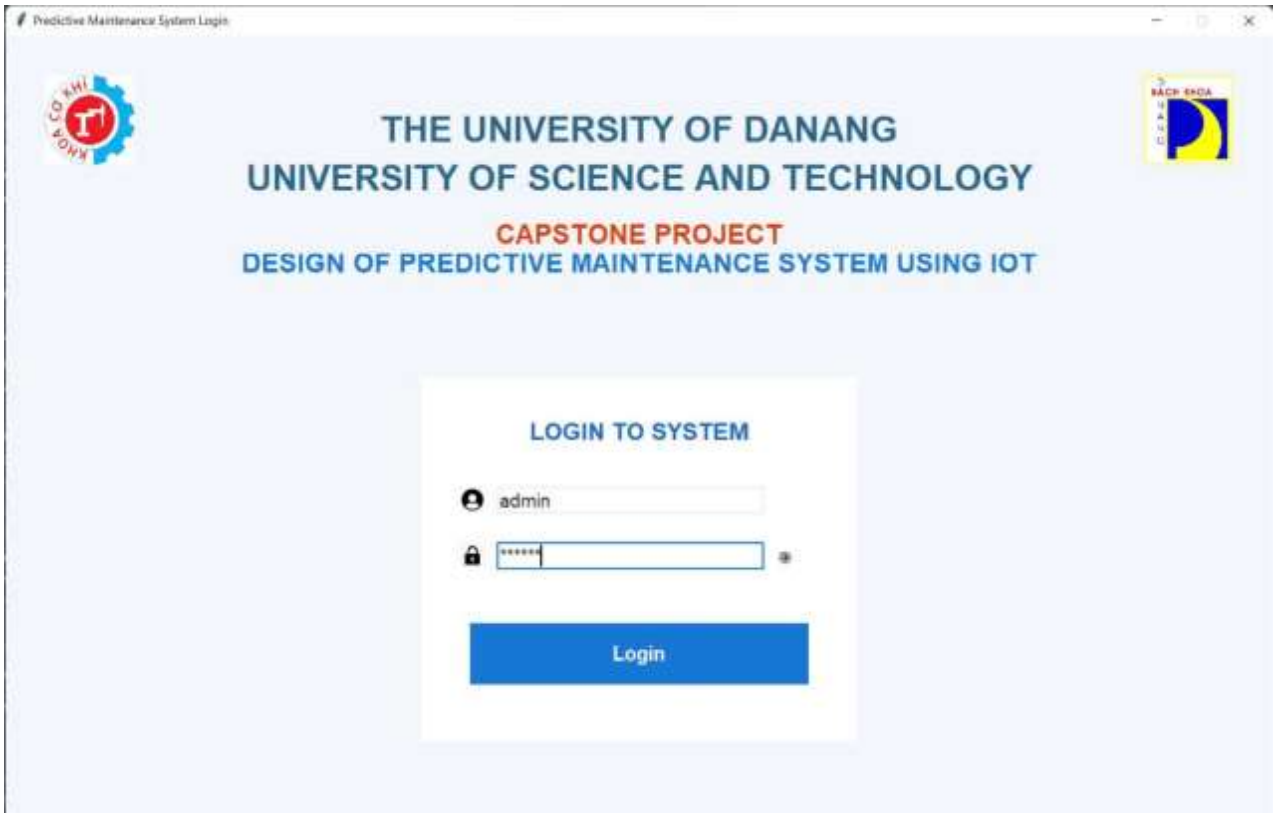


Figure 5.7: Login

The login section will display a login window for users to log into the CMMS software

Users need to enter the correct account and password provided to continue

5.2.1 Dashboard

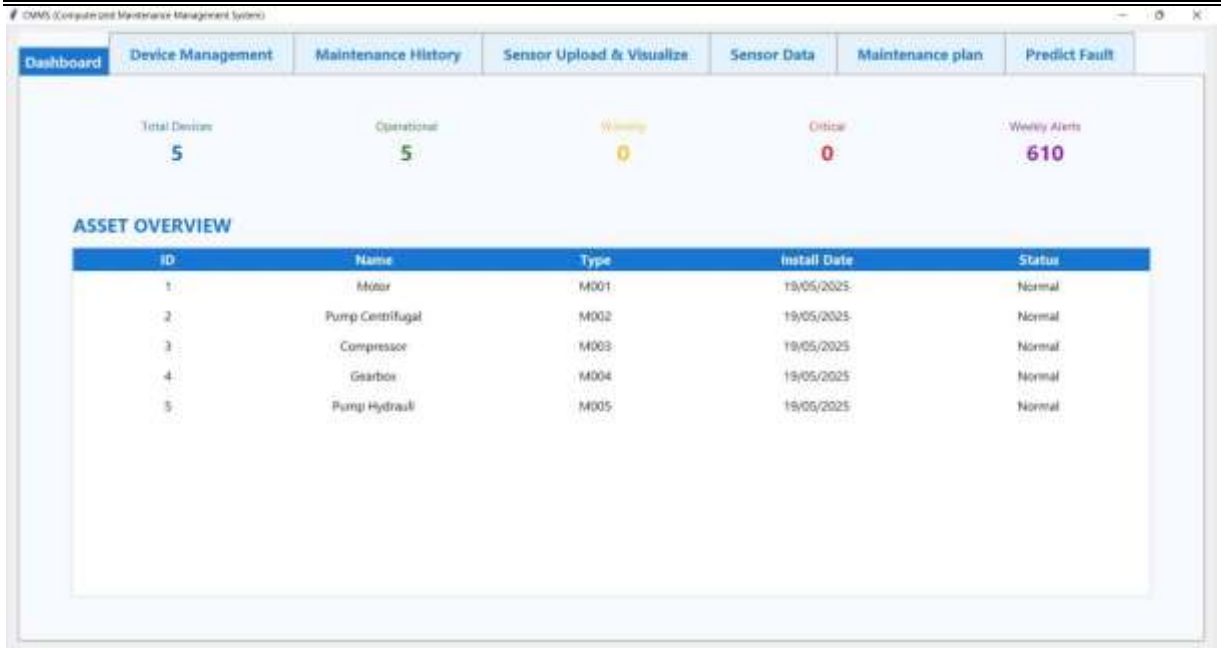


Figure 5.8: Dashboard

Within the devices are directory trees containing device properties.

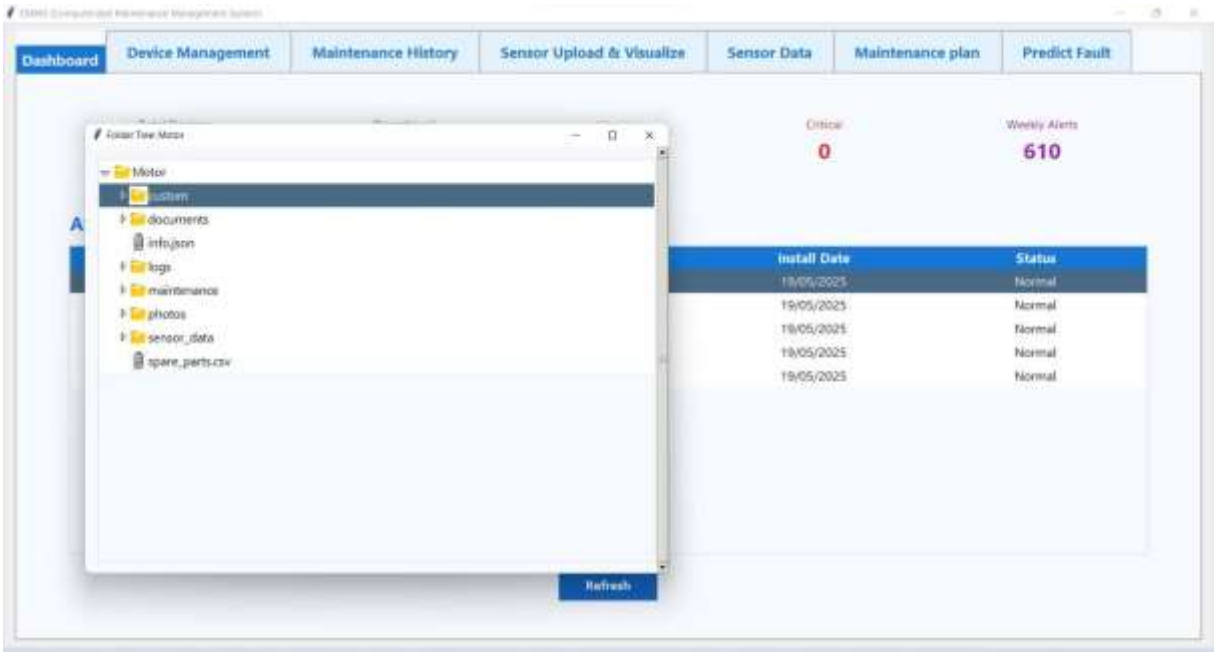


Figure 5.9: Directory trees

The Dashboard tab provides an overview interface of the predictive maintenance system, offering users a quick snapshot of the current status of all equipment within the facility. It displays key indicators such as the total number of devices, number of devices operating normally, devices with Warning status, devices with Critical errors, and the total number of alerts detected during the week. Below the summary, the Asset

Overview table presents detailed information for each device, including ID, name, type, installation date, and current status.

5.2.2 Device Management

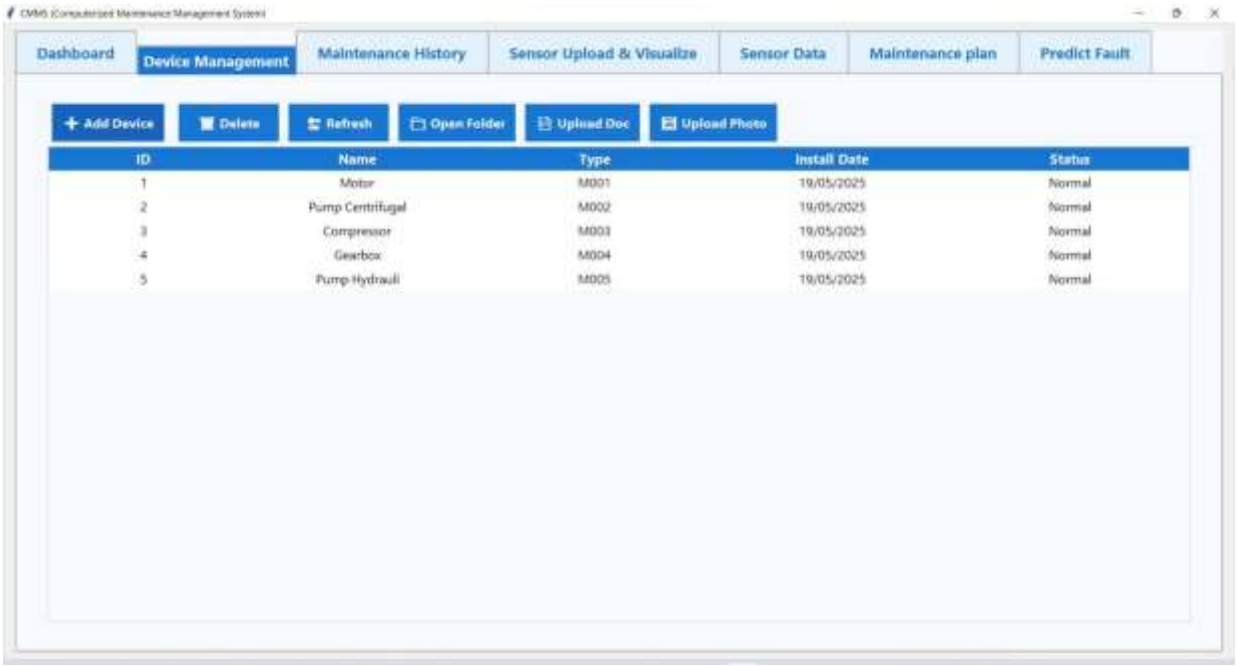


Figure 5.10: Device Management

Allows users to add new devices to the system:

- Input fields:
  + Device Name
  + Type
  + Install Date
  + Status
  + Description
- Click Add Device to save the device to the database.

This section is for adding equipment in the factory or machinery system.

5.2.3 Maintenance History

Maintenance history includes

- Preventive maintenance
- Periodic maintenance

- Predictive maintenance AI

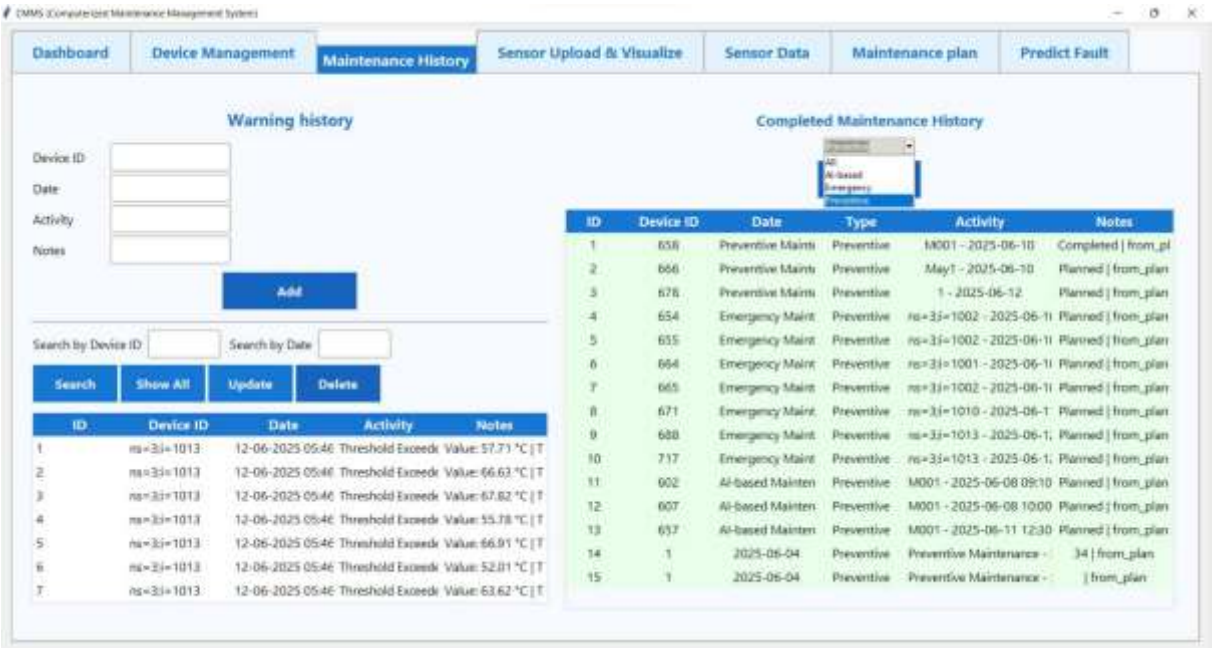


Figure 5.11: Maintenance History

Manage maintenance records for each device:

- Input maintenance information:
  + Device ID
  + Date of maintenance
  + Activity performed (e.g., lubrication, wear check)
  + Notes
- Search maintenance records by Device ID or Date.
- Display of saved maintenance history in a table below.
- Options to Update or Delete existing records.

5.2.4 Sensor Upload & Visualize

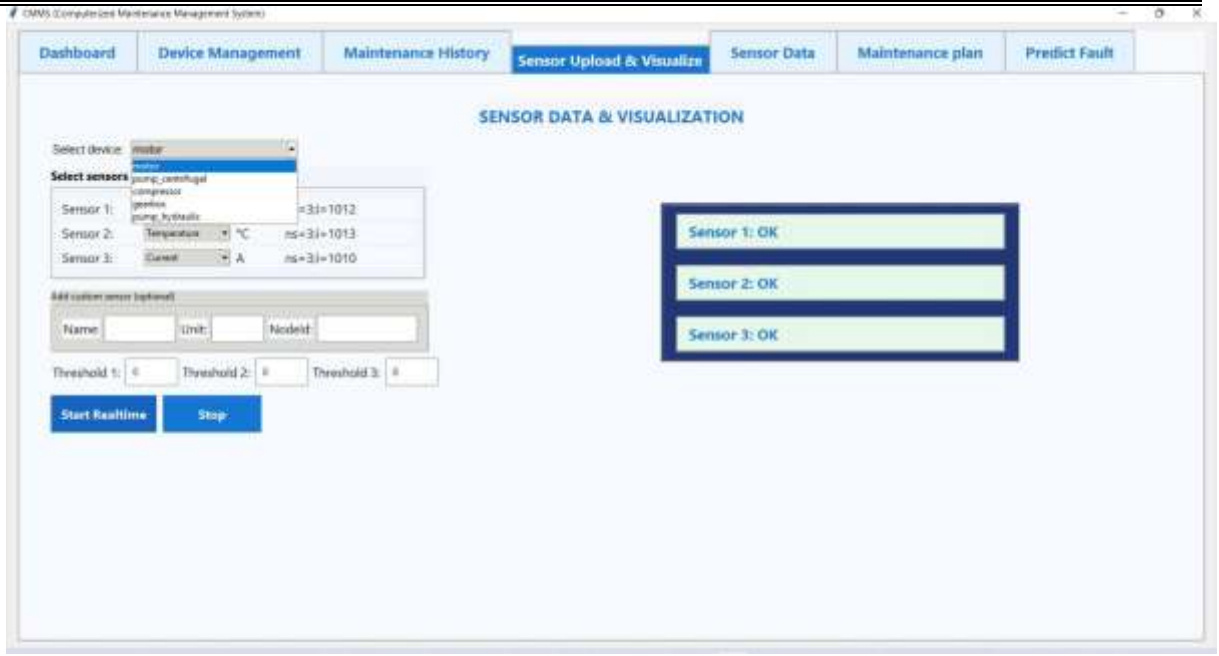


Figure 5.12: Real-time graph of Sensor

Device can be selected to display real-time chart

This section is responsible for importing and visualizing sensor data in real time or from historical files. It includes critical functionality for setting operational thresholds and triggering alerts:

_ Set Warning Threshold:
Users can define a warning threshold value (e.g., vibration, temperature, etc.). This value serves as the limit for acceptable operating conditions.

- Email Alert Notification:

  If the sensor value exceeds the defined threshold, the system will automatically send an email alert to the maintenance technician's email address. This helps in taking immediate action to prevent equipment failure.

- Sensor Selection and Display:

  + Enter a list of sensor NodeIds

  + Choose which sensor to display from the dropdown list.

  + The sensor data will be plotted in a real-time graph for visualization.

5.2.5 Sensor data

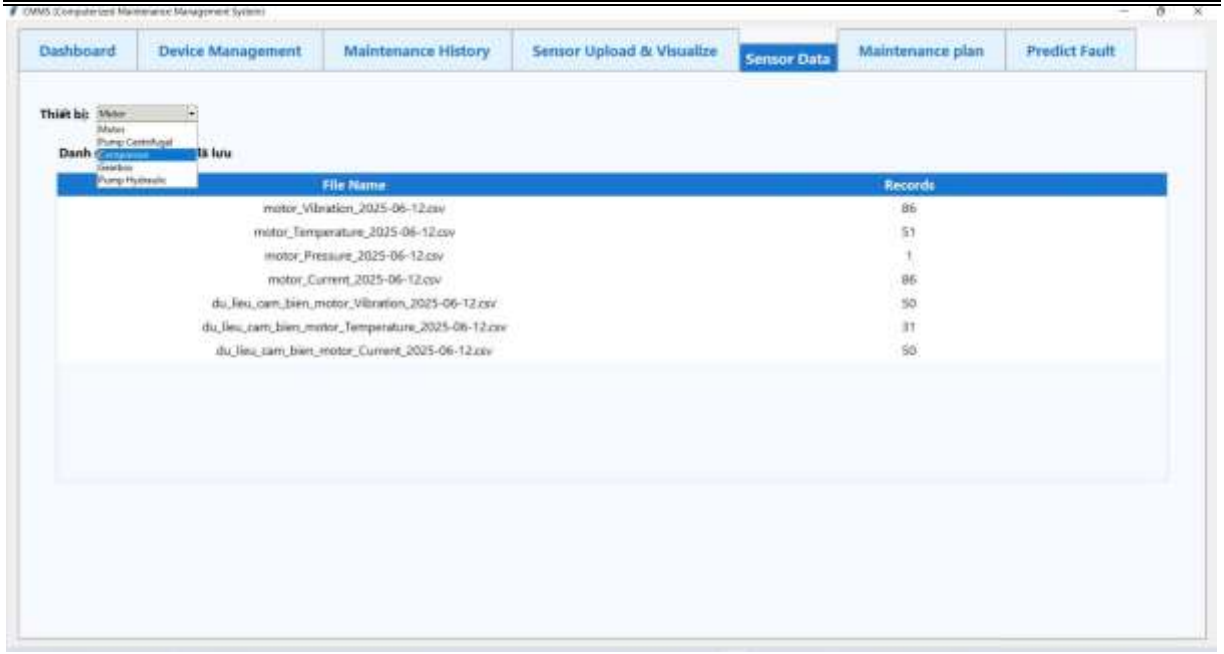


Figure 5.13: Sensor data

The Sensor Data tab displays a list of all sensor data files that have been previously saved into the system. Each entry includes the file name and the number of recorded data rows, allowing users to easily track and manage historical sensor datasets. This tab serves as a centralized location for reviewing collected data from various dates, which is essential for training prediction models, performing diagnostics, or analyzing trends over time.

5.2.6 Maintenance Plan

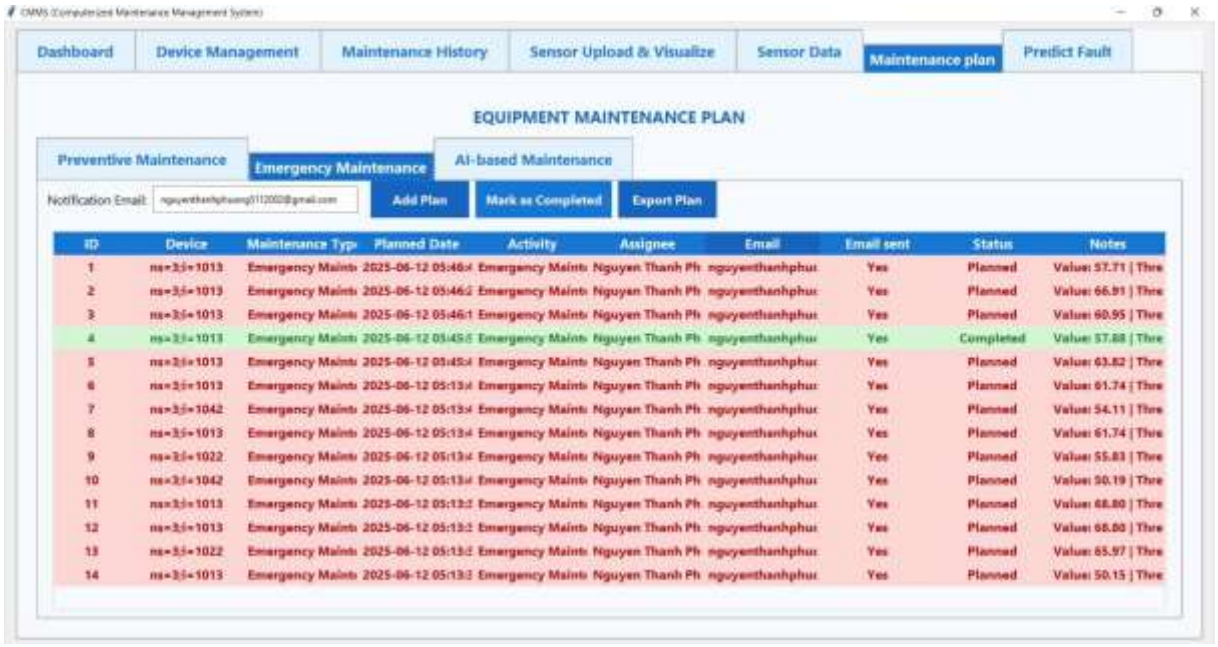


Figure 5.14: Emergency Maintenance

Main Functions of the Maintenance Plan Tab:

- Displays and manages all equipment maintenance plans.
- Supports three types of maintenance:
  + Preventive Maintenance
  + Emergency Maintenance
  + AI-based Maintenance
- Provides control buttons:
  + Add Plan: Add a new maintenance task.
  + Mark as Completed: Mark the task as completed.
  + Export Plan: Export the maintenance plan to a file.
- Sends automated email notifications to technicians when a new task is created.
- Information Displayed in the Plan Table:
  + ID: Unique identifier of the maintenance task.
  + Device: The equipment requiring maintenance.
  + Maintenance Type: Type of maintenance to be performed.
  + Planned Date: Scheduled date and time for maintenance.
  + Activity: Description of the maintenance activity.
  + Assignee: Person assigned to carry out the task.
  + Email: Recipient of the maintenance notification.
  + Email Sent: Indicates whether the notification email was successfully sent.
  + Status: Current status of the task (Planned / Completed).
  + Notes: Additional information such as sensor values or cause of issue.

5.2.7 Predict & Fault

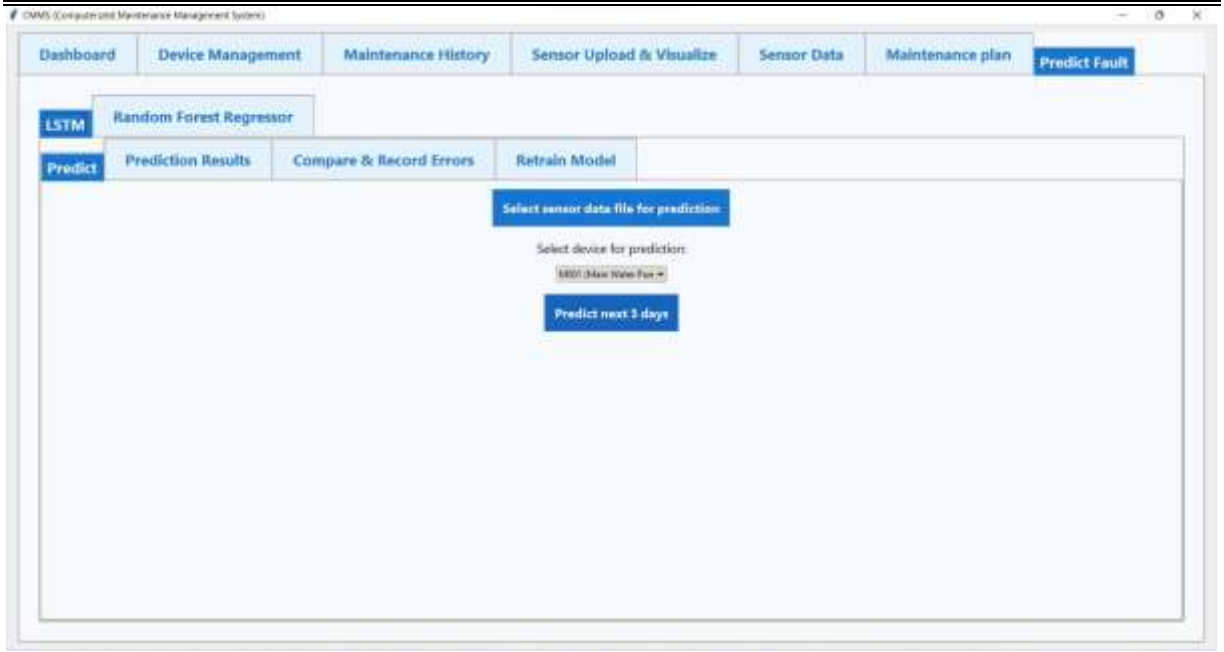


Figure 5.15: Predict fault

The Predict Fault tab allows users to forecast future equipment conditions based on collected sensor data. This interface provides two main algorithm options: LSTM and Random Forest Regressor.

- LSTM (Long Short-Term Memory) is a deep learning model designed for time-series data. The system automatically retrieves the most recent 30 hours of sensor readings to predict the likelihood of faults over the next 3 days.
- Random Forest Regressor is a machine learning algorithm based on an ensemble of decision trees. It analyzes statistical features such as mean, max, standard deviation, and alert ratios from the current day's data to estimate the number of potential faults for the next day.

This tab helps users detect early warning signs, optimize maintenance planning, and improve overall equipment reliability and uptime.

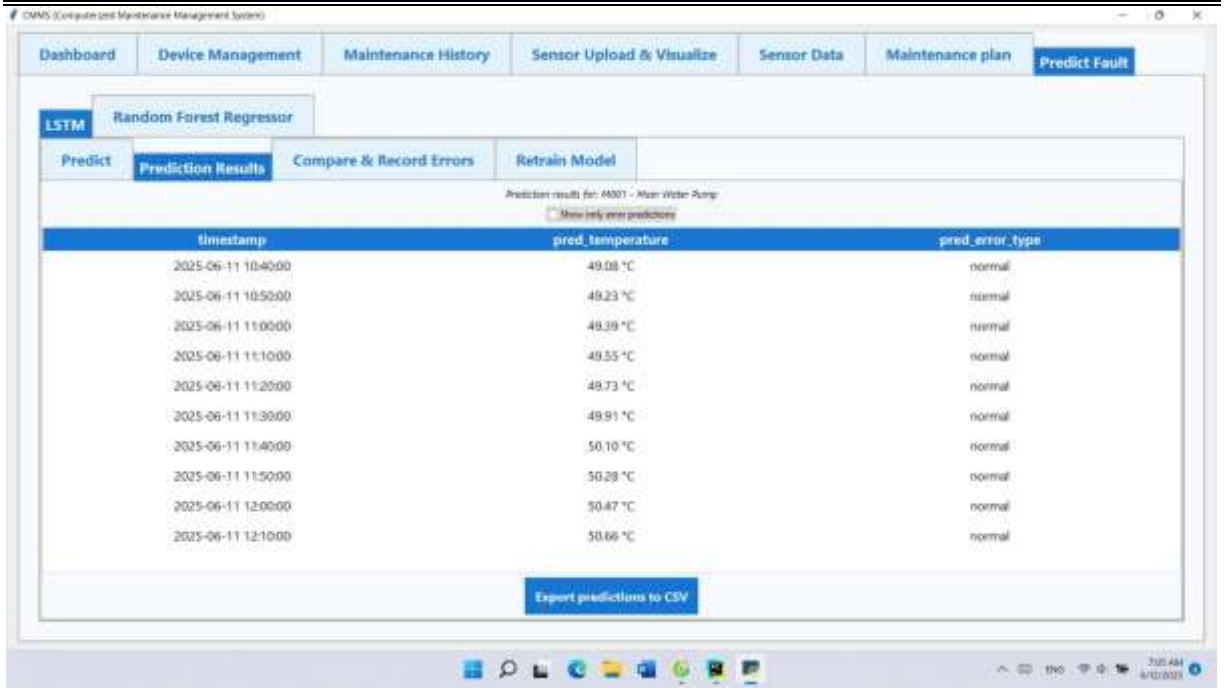


Figure 5.16: Results after prediction

5.3 Send email alert when threshold is exceeded

5.3.1 Objective:

This feature allows the CMMS system to promptly notify technicians when a device records a sensor value that exceeds the predefined safety threshold. This ensures timely maintenance action to prevent serious equipment damage or failure.

5.3.2 How It Works:

- The user sets a warning threshold for each sensor signal (e.g., temperature, vibration, pressure, etc.) in the "Sensor Upload & Visualize" interface.
- The system continuously monitors sensor data in real time.
- When a recorded value exceeds the threshold:
  + The system automatically generates an email alert.
  + The email contains the following details:
    - Alert Time
    - Assigned To
    - Device / Node ID
    - Sensor Name
    - Measured Value

- Threshold
- Activity Description
- Additional Note

- The email is sent to a predefined technician email address.

5.3.3 Example (Actual Email Screenshot)

The alert email will appear as follows:

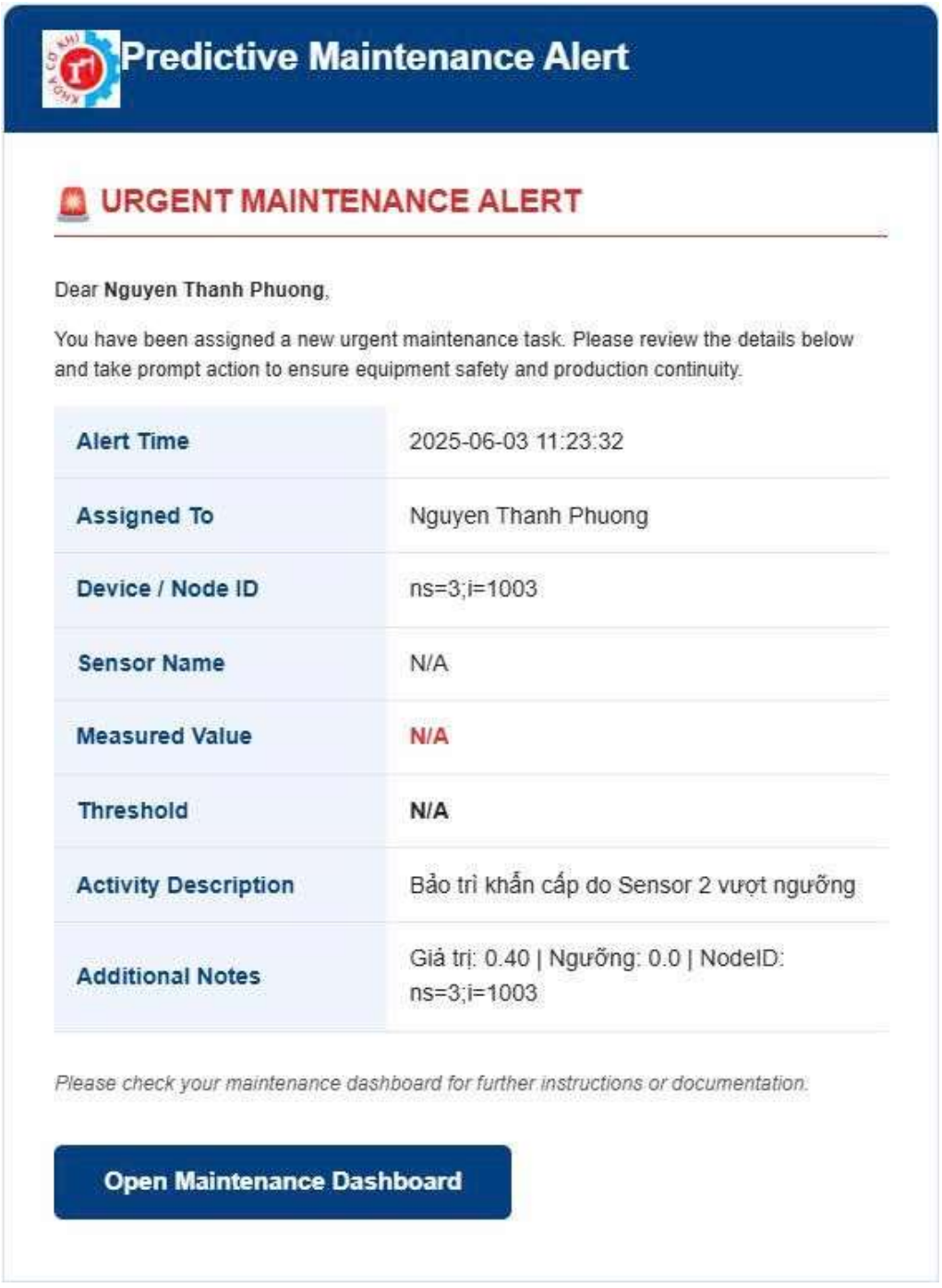


Figure 5.17: Email Alert

5.3.4 Benefits:

- Early detection helps prevent system or equipment failure.
- Increases maintenance efficiency and reduces machine downtime.
- Enables technicians to react quickly and accurately.

5.4 Isolation Forest Algorithm

5.4.1 Concept

Isolation Forest, or iForest, operates based on a unique principle: isolating anomalies by leveraging their distinct characteristics, rather than modeling the normal data distribution. This intuitive and efficient approach has made it a widely adopted method for anomaly detection tasks.

The following provides an overview of how the algorithm functions in practice.

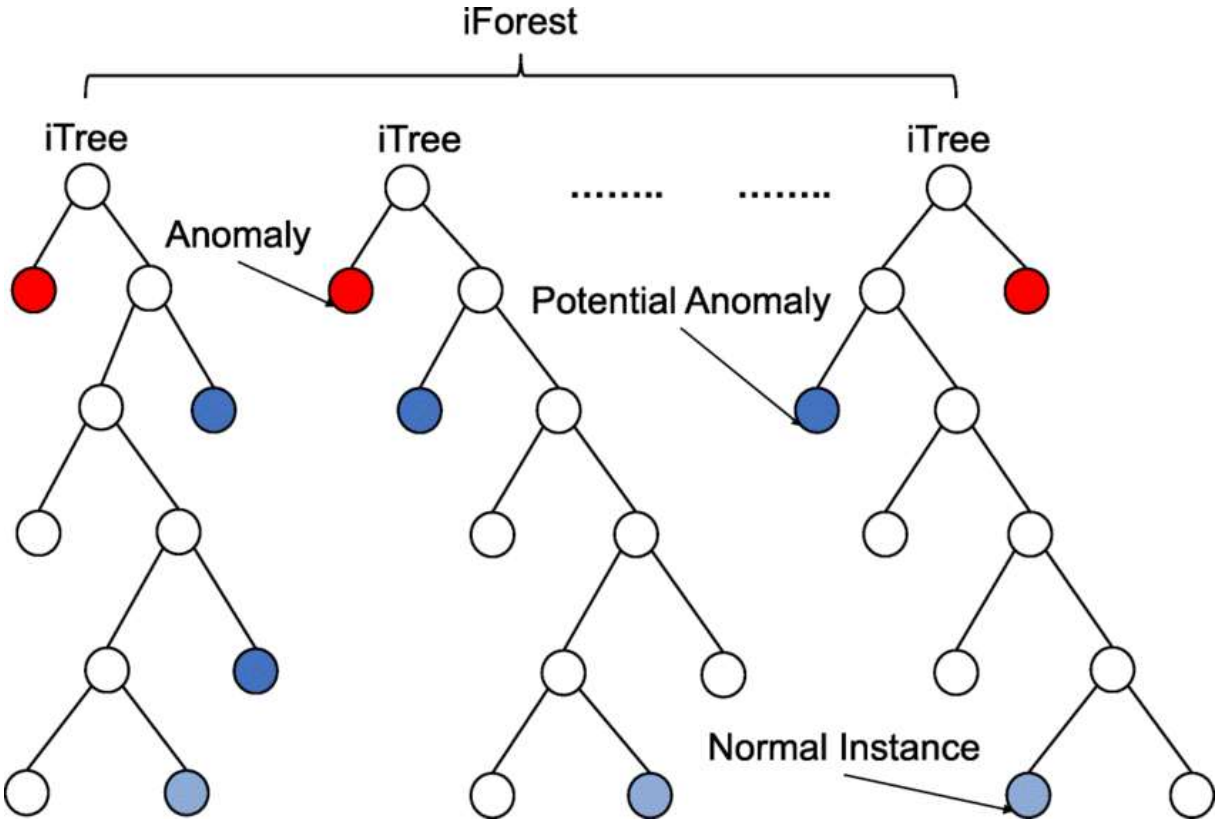


Figure 5.18: Graphical representation of an Isolation Forest

At the core of the Isolation Forest algorithm lies the use of isolation trees (iTrees). These trees are constructed to isolate data points through recursive partitioning.

- Random Sampling
- Random Splitting

- Recursive Partitioning
- Path Length
- Shorter Path Lengths for Anomalies
- Longer Path Lengths for Normal Points
- Anomaly Scoring
- Average Path Length

5.4.2 The Core Idea of Isolation Forest

- Unsupervised anomaly detection algorithm.
- Outliers are highly distinct → easily isolated in the tree → separated quickly → low depth.
- Normal points are embedded within dense clusters → harder to isolate → higher depth.

Formula for calculating abnormal score:

$$S(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

In there:

S(x,n): The anomaly score of data point x, in a dataset of n points.

E(h(x)): The average number of steps (depth) needed to isolate x in multiple trees.

c(n): The average expected depth when isolating a point in a binary tree with n points

The result s(x,n) is in the range [0, 1] — close to 1 is anomalous, close to 0 is normal.

Formula for calculating c(n) – Normalization function

$$c(n) = 2 * H(n-1) - \frac{2(n-1)}{n} \text{ với } H(n-1) \approx ln(n-1) + y$$

In there:

$H(n-1)$: Total harmonic degree n−1

$\gamma \approx 0.5772$: Euler–Mascheroni constant

Table 5.1: Table of values s(x,n)

Value of s(x,n)s(x, n)s(x,n)	**Interpretation**
Close to 1	Highly anomalous
0.4 – 0.6	Uncertain / borderline case
Close to 0	Normal

5.4.3 Overview of advantages and disadvantages of Isolation Forest

Isolation Forest is an efficient and scalable anomaly detection algorithm designed for high-dimensional data. By isolating observations through random partitioning, it identifies anomalies as points that require fewer splits to separate from the rest.

a. Key Advantages

- High efficiency: linear time complexity O(n)O(n)O(n), suitable for large datasets.
- Low memory usage: does not require storing all pairwise distances.
- Effective in detecting well-separated anomalies.
- No assumption about data distribution (non-parametric).
- Works well with high-dimensional data and is less affected by the curse of dimensionality.
- Unsupervised: does not require labeled data for training.
- Intuitive and easy to interpret: based on how quickly a point is isolated.
- Parallelizable: trees can be built independently, speeding up training.

b. Disadvantages, limitations

- Less effective when anomalies are hidden inside dense clusters.
- May confuse noise with true anomalies.
- Cannot be directly applied to categorical or time series data without preprocessing.
- Requires encoding when dealing with categorical features.
- Hard to evaluate performance without labels — metrics like accuracy or precision are not applicable.
- Sensitive to parameters such as n_estimators and max_samples.
- Performance may degrade on very small or highly imbalanced datasets.

5.5 Random Forest Regressor Algorithm[18]

5.5.1 Concept

Random Forest Regressor is a regression algorithm belonging to the ensemble learning family. Its main goal is to predict continuous variables by combining the outputs of multiple decision trees (Decision Tree Regressors). This ensemble approach helps the overall model become more robust, more accurate, and less prone to overfitting than a single tree.

In this project, Random Forest is used to predict the number of machine faults occurring in the morning of the next day, based on sensor data collected from the current day (e.g., vibration levels, shaft displacement, etc.).

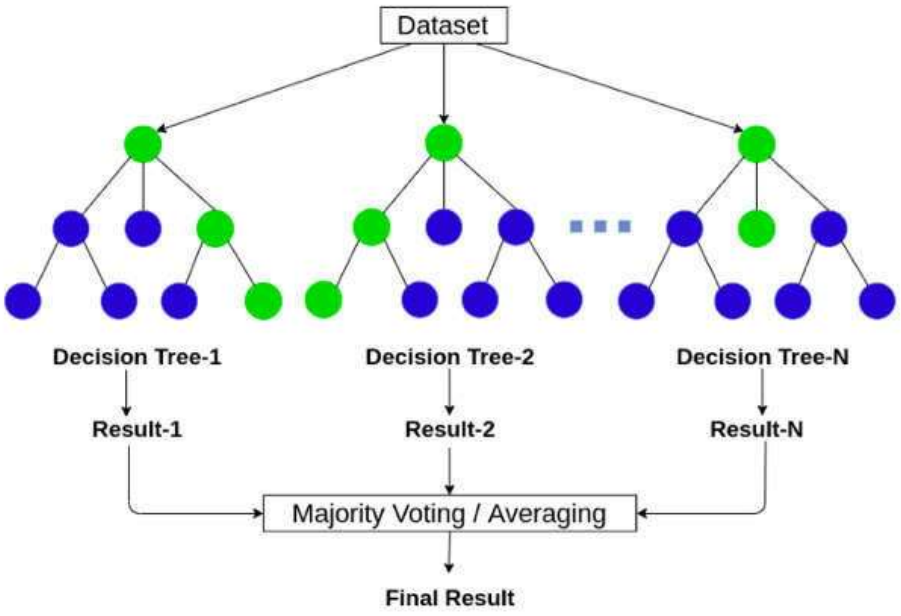


Figure 5.19: Random Forest Regressor[19]

5.5.2 Working principle

- Step 1: Feature extraction and training dataset creation
  - + For each CSV file representing one day, statistical features are extracted such as the mean, max, and standard deviation of the Vibration Sensor and Shaft Displacement Sensor, along with the alert ratio and count.
  - + Each row in the dataset represents one day, and the label is the number of faults that occurred the next morning.
- Step 2: Building multiple trees

+ Each decision tree is trained on a bootstrapped sample of the data.
+ At each node in a tree, only a random subset of features is considered for splitting to reduce correlation among trees.

- Step 3: Aggregating predictions
  + Each tree independently predicts a value.
  + The final prediction is the average of all tree outputs, leading to a smoother and more stable prediction.

5.5.3 Mathematical Formulation

a. Individual Regression Tree Loss[20]:

At each node ttt, the algorithm selects a feature jjj and split point sss to minimize the total variance (Sum of Squared Errors - SSE):

$$SSE(t) = \sum_{i=1}^{N_t} (y_i - \overline{y_t})^2$$

Where:

- y_i is the target value at sample i
- $\overline{y_t}$ is the mean of all y_i in node t
- N_t is the number of samples in the node

The split that minimizes the total SSE of the left and right branches is chosen.

b. Forest-level Prediction[21]:

Given M decision trees $h_1(x), h_2(x), h_3(x), \dots, h_M(x)$, the Random Forest prediction is:

$$\hat{y}(x) = \frac{1}{M} \sum_{m=1}^{M} h_m(x)$$

Where:

- $\hat{y}(x)$ is the predicted value
- $h_m(x)$ is the output from the m^{th} tree

c. Loss Functions for Model Evaluation:

Mean Squared Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE):

Coefficient of Determination (R^2 Score):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where $\bar{y}$ is the mean of the true values.

d. Feature Importance:

Feature importance for a feature f_j is calculated by the total decrease in error (SSE) across all trees where the feature is used to split:

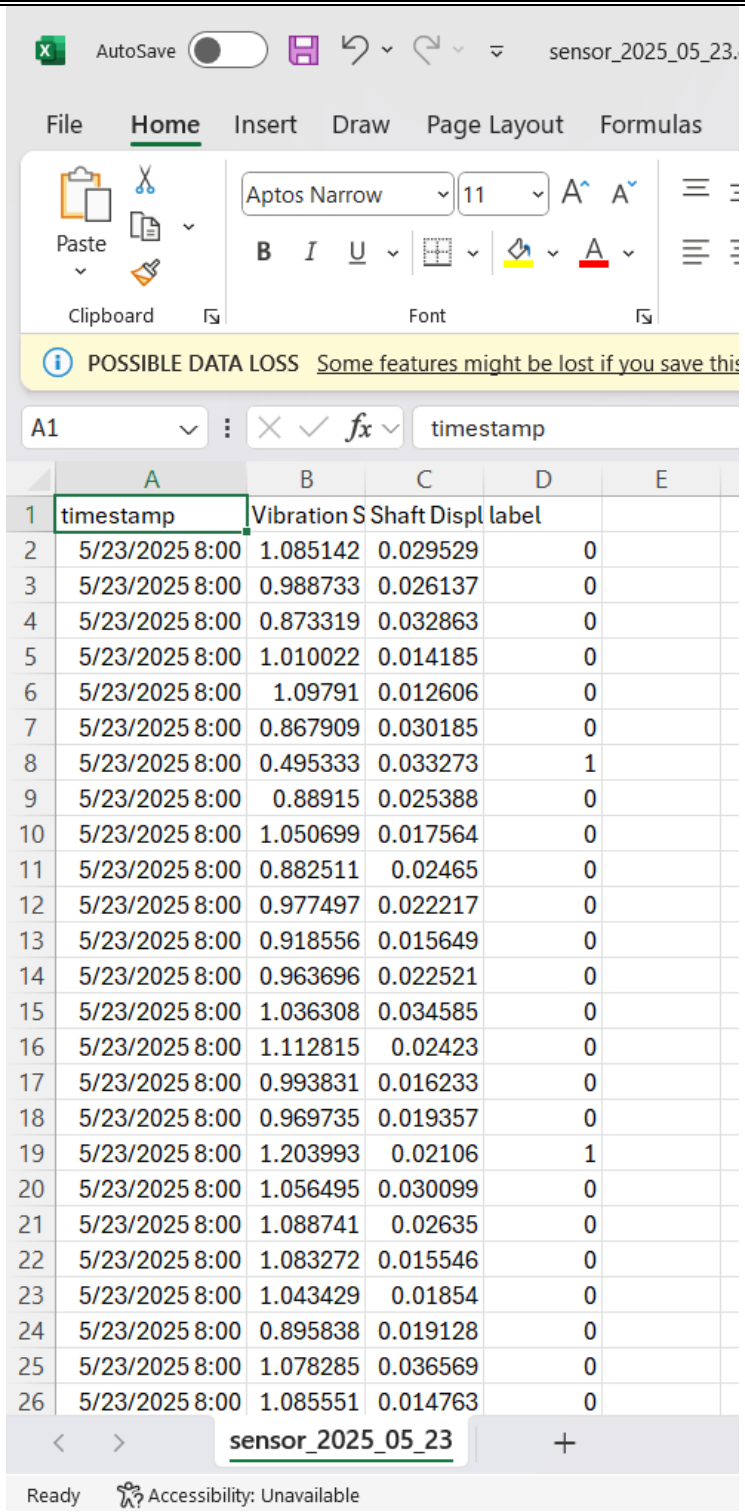
$$Importence(f_j) = \sum_{t \in splits\ on\ f_j} \Delta SSE_t$$

5.5.4 Advantages and Disadvantages

Table 5.2: Advantages and disadvantages of Random Forest Regressor

Advantages	**Disadvantages**
Reduces overfitting significantly	Difficult to interpret
Handles non-linear relationships well	Can be memory-intensive
No feature scaling required	Slower training with large datasets
Provides feature importance metrics	Predictions can be less smooth for small datasets

5.5.5 Training and result



	A	B	C	D	E
1	timestamp	Vibration S	Shaft Displ	label	
2	5/23/2025 8:00	1.085142	0.029529	0	
3	5/23/2025 8:00	0.988733	0.026137	0	
4	5/23/2025 8:00	0.873319	0.032863	0	
5	5/23/2025 8:00	1.010022	0.014185	0	
6	5/23/2025 8:00	1.09791	0.012606	0	
7	5/23/2025 8:00	0.867909	0.030185	0	
8	5/23/2025 8:00	0.495333	0.033273	1	
9	5/23/2025 8:00	0.88915	0.025388	0	
10	5/23/2025 8:00	1.050699	0.017564	0	
11	5/23/2025 8:00	0.882511	0.02465	0	
12	5/23/2025 8:00	0.977497	0.022217	0	
13	5/23/2025 8:00	0.918556	0.015649	0	
14	5/23/2025 8:00	0.963696	0.022521	0	
15	5/23/2025 8:00	1.036308	0.034585	0	
16	5/23/2025 8:00	1.112815	0.02423	0	
17	5/23/2025 8:00	0.993831	0.016233	0	
18	5/23/2025 8:00	0.969735	0.019357	0	
19	5/23/2025 8:00	1.203993	0.02106	1	
20	5/23/2025 8:00	1.056495	0.030099	0	
21	5/23/2025 8:00	1.088741	0.02635	0	
22	5/23/2025 8:00	1.083272	0.015546	0	
23	5/23/2025 8:00	1.043429	0.01854	0	
24	5/23/2025 8:00	0.895838	0.019128	0	
25	5/23/2025 8:00	1.078285	0.036569	0	
26	5/23/2025 8:00	1.085551	0.014763	0	

Figure 5.20: Data needed for training

After training, we will run the prediction file to select the data file we need to predict, it will run the prediction result for the machine status the next day of the data day we choose.

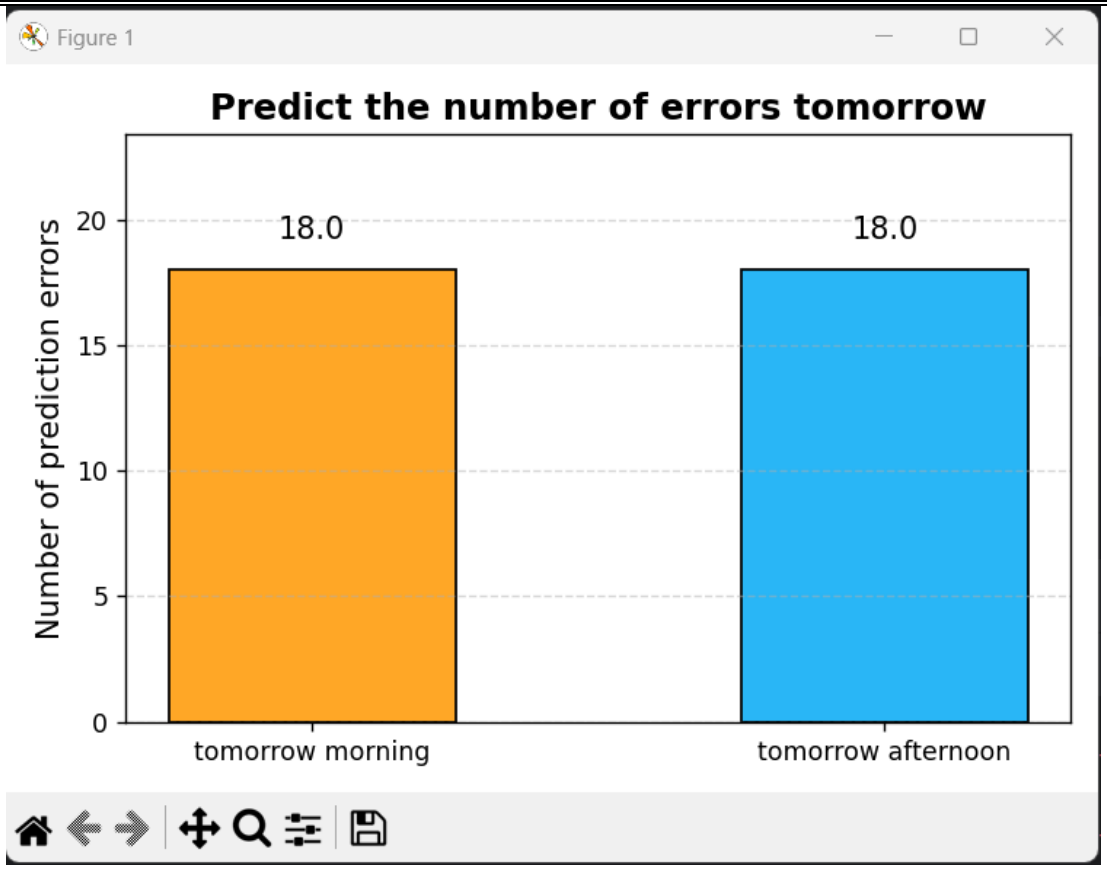


Figure 5.21: Prediction chart

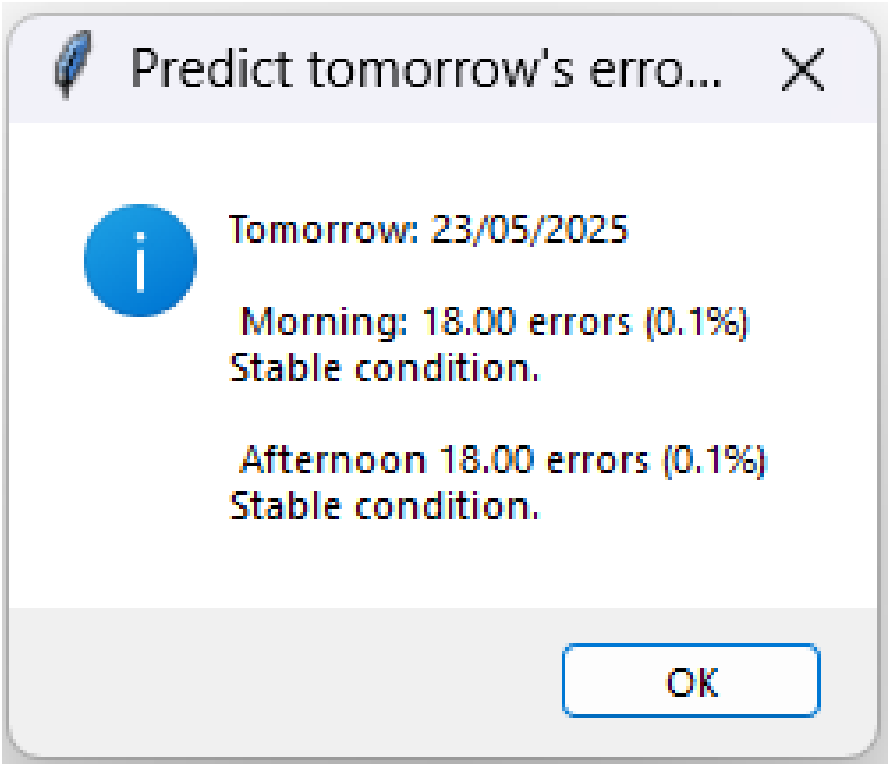


Figure 5.22: Forecast Notification

5.6 LSTM Algorithm

Introduction[22]:

- Long Short-Term Memory networks (LSTM) are a special type of Recurrent Neural Network (RNN) capable of learning long-range dependencies. LSTM was introduced by Hochreiter & Schmidhuber in 1997 and has since been improved and widely adopted across the field. Due to their effectiveness in solving a variety of problems, LSTMs have become increasingly popular in recent years.
- LSTM is specifically designed to overcome the issue of long-term dependency. Remembering information over extended sequences is a built-in capability of LSTM, and does not require external mechanisms or additional training to achieve. This inherent ability allows the network to retain relevant information over time without manual intervention.
- Like other recurrent networks, an LSTM network consists of a chain of repeating neural network modules, where each module passes information to the next in sequence.

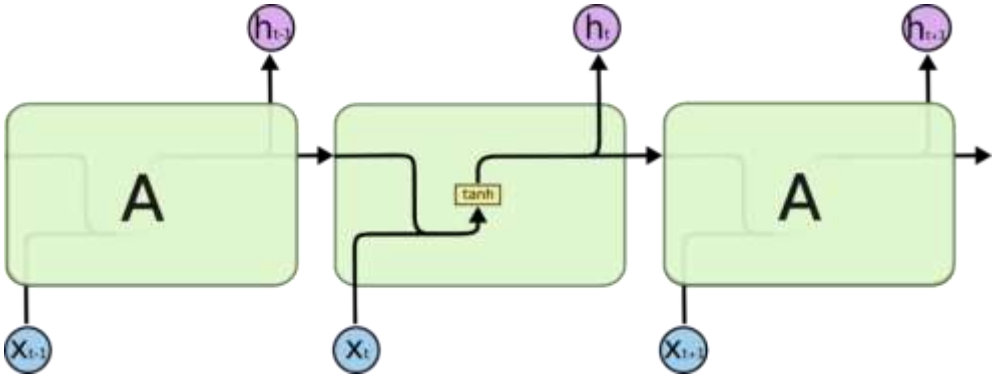


Figure 5.23: Module in RNN network

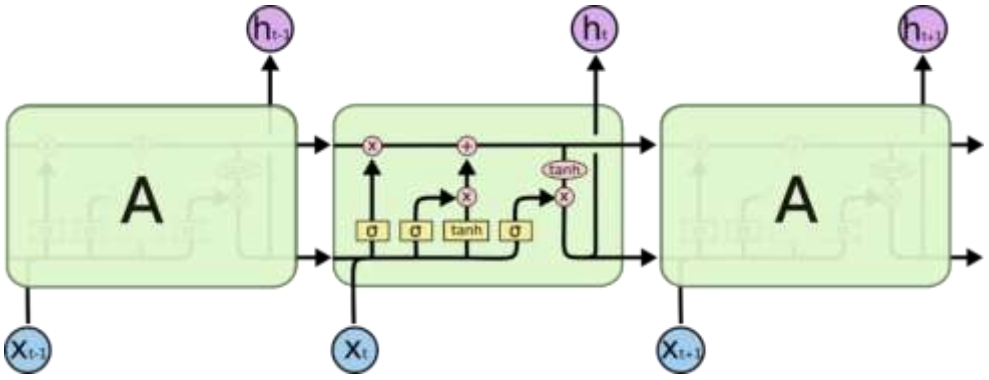


Figure 5.24: The recurrent module in LSTM contains four interacting layers

5.6.1 Core idea

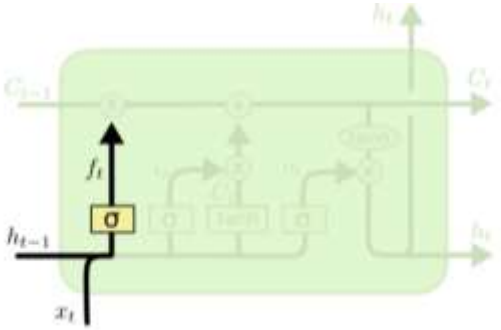
- In many applications such as sensor prediction or event sequence analysis, data often has a temporal and sequential structure. Traditional machine learning models struggle with this type of data because they cannot retain information from the past.
- Recurrent Neural Networks (RNNs) were designed to address this problem by passing information from one step to the next. However, RNNs face serious challenges with long sequences: early information tends to "fade away" or the gradients become very small, making it difficult for the model to learn long-term dependencies — a phenomenon known as the *vanishing gradient* problem.
- Long Short-Term Memory (LSTM) was introduced as a significant improvement over RNNs, specifically to overcome the limitation of forgetting distant information in sequences.
- The core idea is that instead of passively passing information like RNNs, LSTM actively controls the flow of information through time using an intelligent mechanism called selective memory.
- Specifically, each LSTM unit has an internal memory (cell state) and three logic gates that determine whether to:
- Forget certain information from the past
- Remember important information from the present
- Output relevant information to the next step

5.6.2 Order of steps of LSTM algorithm[23]

- The Forget Gate is the first component in an LSTM unit, which plays an important role in managing the information stored in the memory (cell state) over each time step. Its main task is to determine which information from the previous state should be retained, and which information should be discarded to serve the learning of the network.
- The forget gate uses a sigmoid activation function to process two main inputs:
  + h_{t-1}: hidden state from previous time step
  + x_t: input at current time

+ Calculation formula: f_t = σ(W_f.[h_{t-1}+ x_t] + b_f)

- In there:
  + W_f: weight of the forgetting layer
  + b_f : bias
  + σ: sigmoid function (gives output from 0 to 1)
- Result f_t is a vector of the same size as the memory state C_t.
  + If an element of f_t near 1, information at the corresponding position inC_{t-1} will be retained in full.
  + If close to 0, the information will be completely erased.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

Figure 5.25: Forgotten Gate Floor

- After the forgetting layer determines which parts of the old information should be discarded, the input gate layer decides what new information should be added to the cell state.

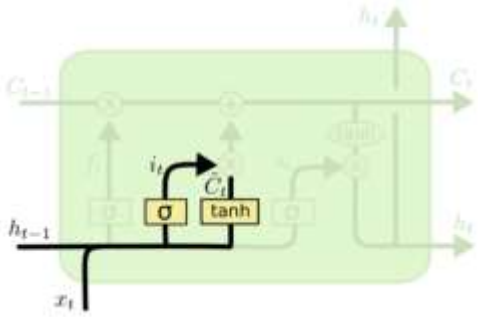
The process consists of two parts:

Sigmoid layer (input gate layer): Get input $x_t$và hidden status h_{t-1} Outputs a value from 0 to 1 → represents the level of new information that will be updated

Layer:

+ Generate a vector corresponding to the new state value $\bar{c}_t$
+ This is the candidate information to add to the memory cell.

- Finally, the results of the two layers above are multiplied together, resulting in the new information being added to the memory cell, along with the old information being retained (if any).



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

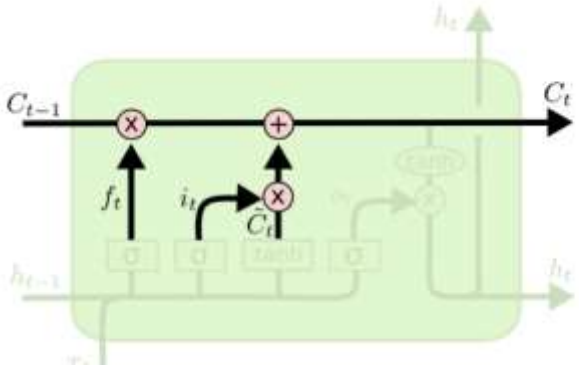
Figure 5.26: Update the value for the status cell

- LSTM performs updating of old memory cell stateC_{t-1} to new stateC_t, based on pre-calculated decisions.

How to do:

- Multiply $f_t C_{t-1}$
  + Retains important parts of the previous information
  + Represents what is not forgotten
- Multiply $i_t \bar{c}_t$
  + Adds new relevant information
  + i_t: level of update permission (input gate)
  + $\bar{c}_t$: newly proposed information
- Combined update:

$$C_t = f_t . C_{t-1} + i_t \bar{c}_t$$



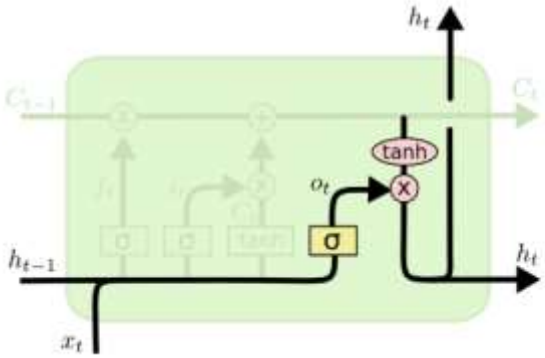
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Figure 5.27: New status box

The output layer determines which part of the cell state will be emitted as the result at the current time step.

- Sigmoid Gate:
  + Takes the input and decides which part of the cell state should be output.
  + Outputs values between 0 and 1 to filter and select relevant information.
- Tanh Gate:
  + Scales the cell stateC_t= to the range [−1,1] helping stabilize the output h_t
- Combination:
  + The output of the tanh function is multiplied by the sigmoid output gate → produces the final output.

$$h_t = sigmoid\ \mathrm{W}_0.[h_{t-1} + x_t]\ .tanh\ C_t)$$



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 5.28: Output via tanh function

5.6.3 Deploying the problem of predicting vibration sensors

- Vibration data is collected in real time from sensors mounted on bearings or gearboxes, via OPC UA protocol or CSV file storage..

	A	B	C	D	E
1	timestamp	device_id	device_type	temperatu	status
2	1/1/2024 0:00	M001	motor	51.98	normal
3	1/1/2024 0:10	M001	motor	53.56	normal
4	1/1/2024 0:20	M001	motor	51.6	normal
5	1/1/2024 0:30	M001	motor	52.52	normal
6	1/1/2024 0:40	M001	motor	52.66	normal
7	1/1/2024 0:50	M001	motor	51.66	normal
8	1/1/2024 1:00	M001	motor	54.03	normal
9	1/1/2024 1:10	M001	motor	51.24	normal
10	1/1/2024 1:20	M001	motor	52.6	normal
11	1/1/2024 1:30	M001	motor	51.86	normal
12	1/1/2024 1:40	M001	motor	52.04	normal
13	1/1/2024 1:50	M001	motor	52.54	normal
14	1/1/2024 2:00	M001	motor	57.56	normal
15	1/1/2024 2:10	M001	motor	70	error
16	1/1/2024 2:20	M001	motor	52.8	normal
17	1/1/2024 2:30	M001	motor	56.08	normal
18	1/1/2024 2:40	M001	motor	54.24	normal
19	1/1/2024 2:50	M001	motor	52.72	normal
20	1/1/2024 3:00	M001	motor	56.38	normal
21	1/1/2024 3:10	M001	motor	55.35	normal
22	1/1/2024 3:20	M001	motor	54.53	normal
23	1/1/2024 3:30	M001	motor	52.86	normal
24	1/1/2024 3:40	M001	motor	52.99	normal
25	1/1/2024 3:50	M001	motor	54.95	normal
26	1/1/2024 4:00	M001	motor	57.37	normal
27	1/1/2024 4:10	M001	motor	53.94	normal
28	1/1/2024 4:20	M001	motor	54.88	normal
29	1/1/2024 4:30	M001	motor	56.63	normal

Figure 5.29: Vibration sensor data

- The input data, collected from vibration sensors mounted on the gearbox, is represented as a time series:

$$X = \{x_1, x_2, x_3, ..., x\_T\}$$

where x_1 is the vibration value at time t, and T is the number of samples in a measurement sequence.

- To bring all values into a common range and improve model stability during training, the data is normalized using Min-Max normalization.

$$x_t^{(norm)} = \frac{x_t - x_{\min}}{x_{\max} - x_{\min}} \in [0, 1]$$

- The data is cut into substrings of fixed length L using the sliding window algorithm:

$$Si=\{xi,xi+1,...,xi+L-1\}$$

  + yi=0 if string is from h30hz file (healthy)
  + yi=1 if string is from b30hz file (faulty)

Figure 5.30: LSTM model after training

- After training the LSTM model to classify the vibration sensor data, the system performed predictions on a test dataset. The results are saved in the prediction_result.csv file and displayed as follows:

Result Analysis:

- Window Start indicates the starting index of the input data segment (based on the sliding window).
- Prediction refers to the model's output:
  + 1 → The model predicts the segment contains faulty signals.
  + 0 → The model predicts the segment is healthy.
- LSTM Model Evaluation and Remarks:
- After training and testing the LSTM model on vibration sensor data from a gearbox system, the model produced highly accurate predictions. It utilized time series input data, processed through a sliding window technique to learn temporal vibration patterns effectively.
- The model's output is a classification label for each data segment, indicating whether the equipment is in a healthy or faulty state. According to the results

table, the model successfully distinguished most abnormal vibration segments. This demonstrates LSTM's ability to learn and retain characteristic vibration patterns associated with equipment faults.

- Advantages:
  + Capable of early detection of abnormal vibration signals, even before clear failure symptoms appear.
  + Suitable for large-scale datasets, with long-term memory of sequential dependencies.
  + Provides segment-level predictions, allowing for more flexible and precise analysis of machine condition.
- Limitations:
  + The model may misclassify segments at transitional boundaries between healthy and faulty states due to unclear signal distinctions.
  + Accuracy heavily depends on preprocessing quality, especially normalization and the choice of input window size.

5.6.4 Combining LSTM and Isolation Forest for training

a. Purpose of the Integration

- The LSTM model is used to forecast machine behavior (e.g., sensor readings) over the next 72 hours.
- The Isolation Forest model then evaluates these predicted sequences to determine whether they exhibit abnormal patterns.
- This allows the system to identify potential failures ahead of time, before any real breakdown occurs.

b. Integrated Training Workflow

- Train the LSTM model:
  + The LSTM is trained on historical time-series data (e.g., the past 24 hours of sensor data) to predict future sensor values over the next 72 hours.
  + It captures temporal dependencies and trends, such as gradually increasing vibration or periodic fluctuations.

- Generate predicted sequences:
  + Once trained, the LSTM model is used to generate multiple predicted sequences from historical input windows.
  + Each predicted sequence represents a simulated future scenario of equipment behavior.
- Train the Isolation Forest on predicted outputs:
  + These LSTM-generated predictions form a new dataset used to train the Isolation Forest model.
  + Isolation Forest learns to distinguish between normal predicted behavior and anomalous predicted trends, based on the statistical isolation of data points.
- Perform anomaly detection on new predictions:
  + For any new input sequence, the LSTM generates a 72-hour forecast.
  + The Isolation Forest evaluates this forecast and flags it as abnormal if it deviates significantly from what is considered normal.

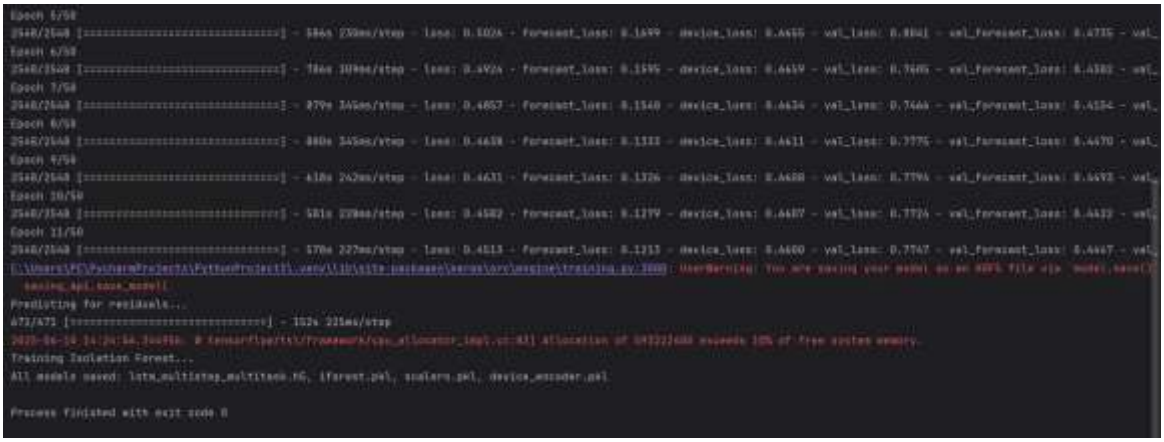


Figure 5.31: Training LSTM + Isolation forest model

c. Advantages of the Combined Approach

- Enables early detection of potential failures, before they manifest in real sensor readings.
- Leverages the strengths of both models:
  + LSTM excels at time-series forecasting.
  + Isolation Forest excels at detecting outliers without labeled data.

- Helps reduce false positives compared to using either model in isolation.

5.7 Predictive Maintenance System Website

This website is designed to support real-time monitoring, fault alerting, and predictive maintenance for industrial equipment. The interface is user-friendly and organized into key functional modules as follows:

- Login
- Dashboard
- Devices
- Plans
- Visualization
- Alerts
- AI Predict

5.7.1 Login

- The initial page allows authorized users to log in to the system.
- Ensures system access security by requiring account authentication.

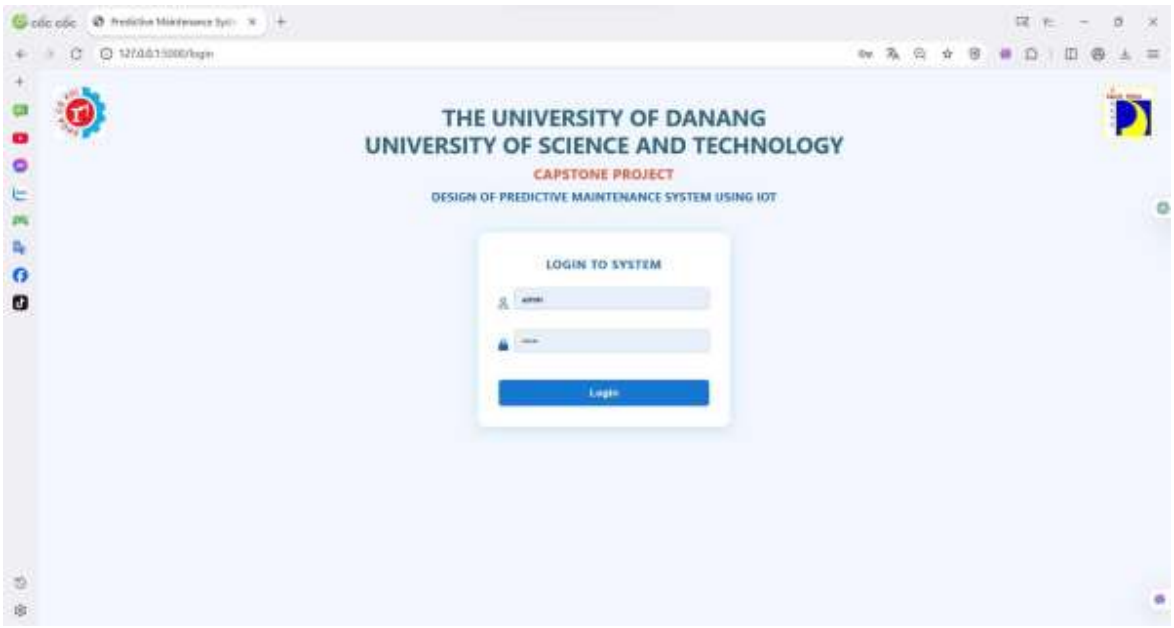


Figure 5.32: Login website

5.7.2 Dashboard

- Provides a quick overview of system status:

+ Number of registered devices
+ Number of devices operating normally
+ Number of active alerts

- Displays summary insights to help managers quickly assess overall conditions.

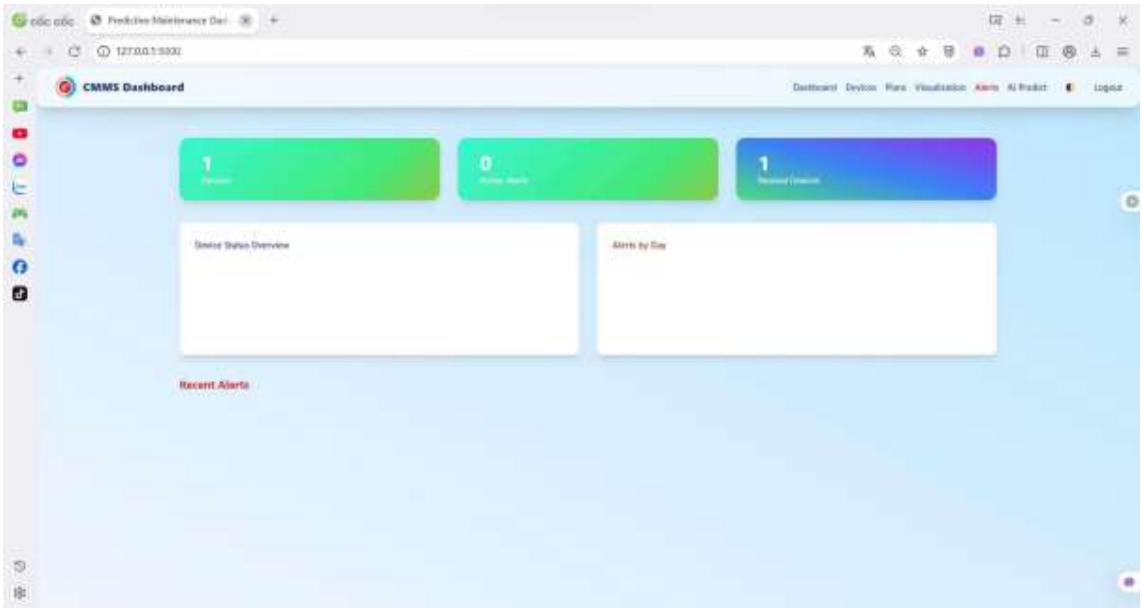


Figure 5.33: Dashboard

5.7.3 Devices

- Displays a list of monitored devices with details such as name, type, installation date, and status.
- Allows adding new devices or removing existing ones.
- Devices are linked to real-time sensor data for monitoring and diagnostics.

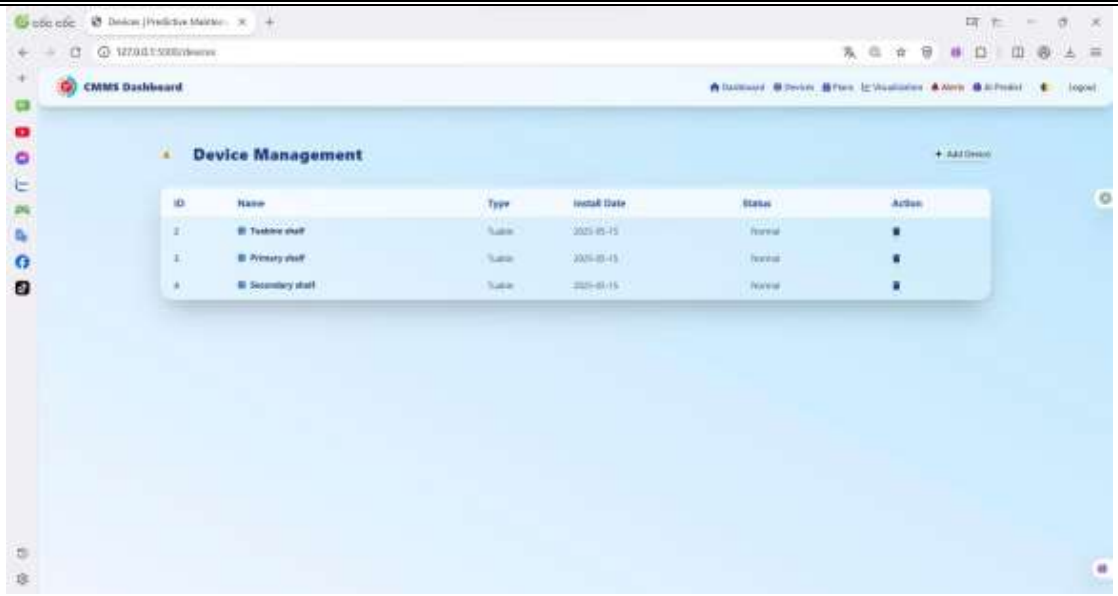


Figure 5.34: Devices Management

5.7.4 Plans

- Manages all maintenance tasks categorized into:
  + Preventive Maintenance
  + Emergency Maintenance
  + AI-based Maintenance
- Displays maintenance type, schedule, assignee, email notifications, and task status (Planned/Completed).

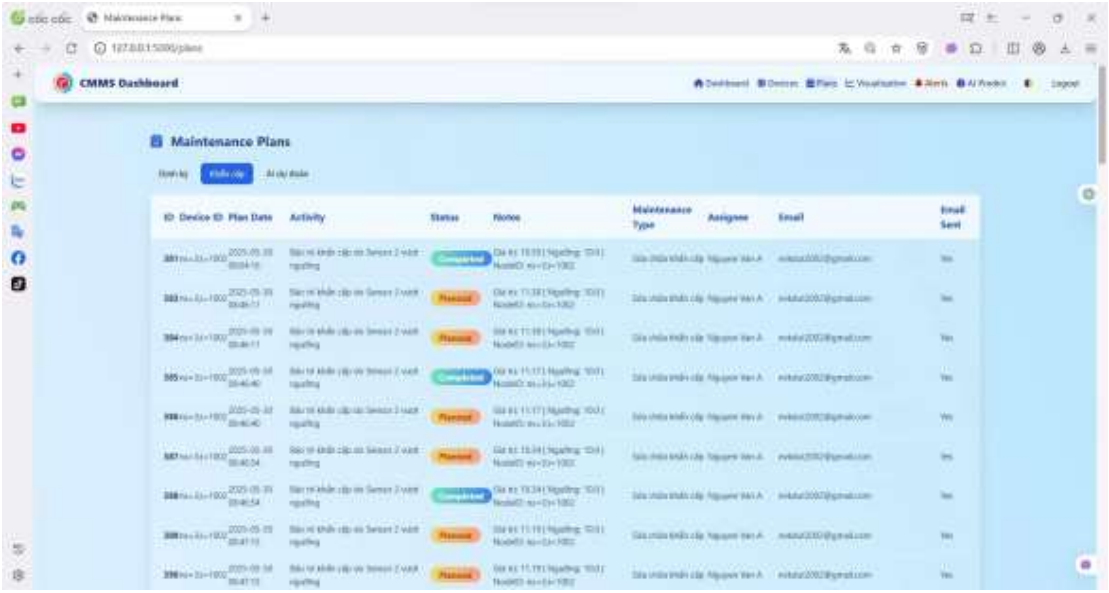


Figure 5.35: Emergency Maintenance

5.7.5 Visualization

- Real-time line charts show live sensor readings from multiple sensors.
- Users can define alert thresholds for each sensor (e.g., sensor1, sensor2, sensor3).
- If a sensor value exceeds its threshold, the system automatically generates alerts.

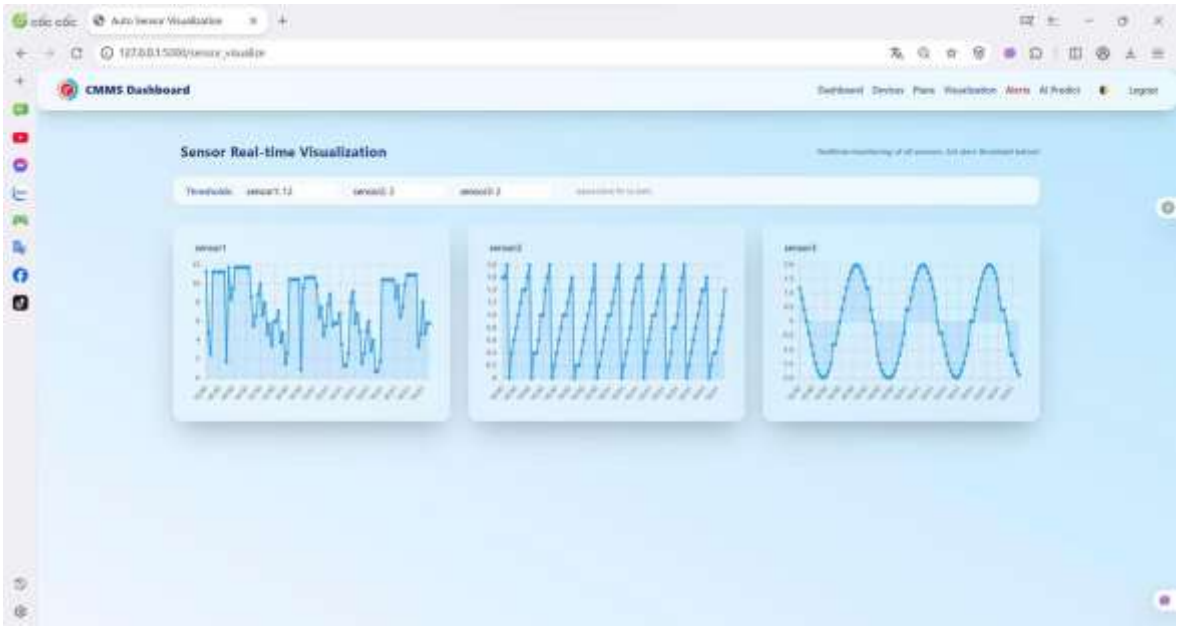


Figure 5.36: Sensor Real-time Visualization

5.7.6 Alerts

- Lists recent and active alerts triggered by sensor data anomalies.
- Helps users monitor device conditions and act promptly to prevent failures.

5.7.7 AI Predict

- Allows users to upload sensor data in .csv format for fault prediction using machine learning models:
  + LSTM for predicting equipment condition over the next 3 days
  + Random Forest Regressor for estimating the number of possible faults the next day
- Supports proactive maintenance planning and early risk detection.

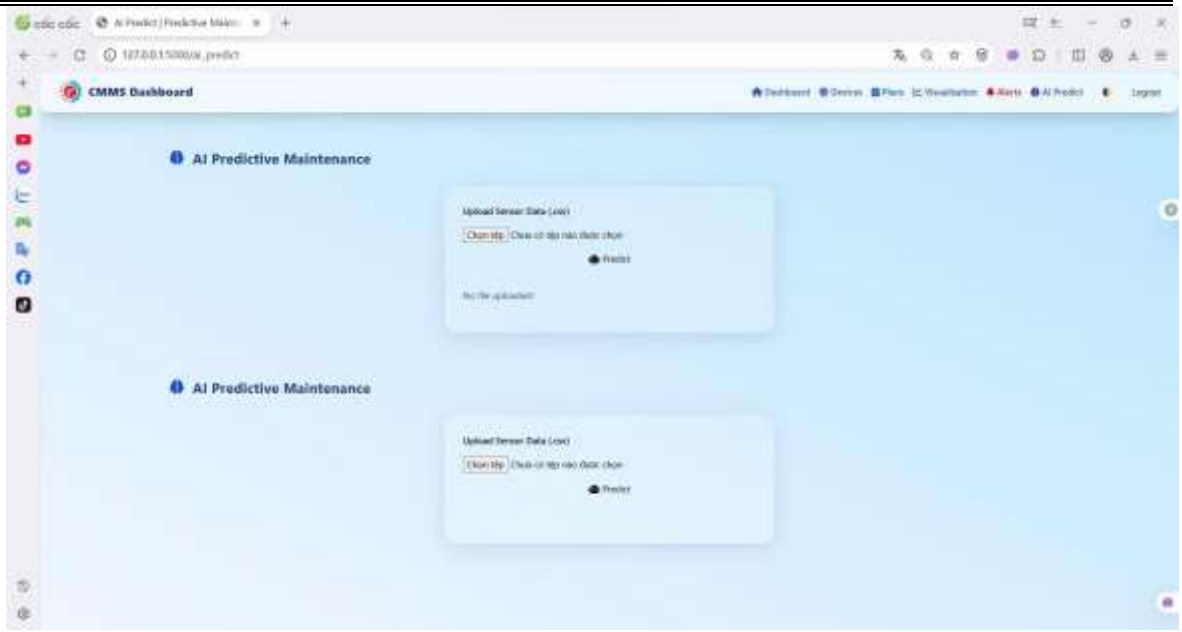


Figure 5.37: AI Predict

CHAPTER 6 CONCLUSION

6.1 Results achieved

After a period of research, design, and implementation, the project "Design of Predictive Maintenance System Using IoT" has yielded the following notable results:

- Successfully designed and developed a predictive maintenance system based on IoT technology, capable of collecting sensor data from industrial equipment, such as vibration, shaft displacement, temperature, and current sensors.
- Built a fully functional CMMS software (Computerized Maintenance Management System) that integrates multiple features, including device management, maintenance scheduling, maintenance history tracking, sensor data visualization, AI model training, and failure prediction.
- Applied the Random Forest Regressor machine learning algorithm to predict the number of potential failures that may occur in the next morning, based on daily sensor data statistics such as mean, standard deviation, and alert frequency.
- Presented prediction results in intuitive bar charts, accompanied by warning messages categorized by severity levels (safe, mild alert, or critical).
- Implemented an automated email alert system, which notifies technicians when any sensor signal exceeds a predefined threshold, enabling timely intervention.
- Successfully tested the prediction model on real-world data, achieving relatively high accuracy ($R^2 > 0.85$), demonstrating its feasibility for industrial applications.

Overall, the system meets the requirements of an intelligent maintenance solution by using actual sensor data, issuing early warnings, minimizing downtime, and improving operational efficiency.

Figure 6.1: 3D model of control system

Figure 6.2: Real-time sensor monitoring

6.2 Limitations

Despite the positive results, the project still has several limitations that need to be addressed:

- Limited deployment environment: The system has only been tested in a simulated or semi-realistic setup, not yet deployed in a large-scale industrial environment with multiple devices and harsh conditions.
- Prediction accuracy depends heavily on input data quality: If the collected sensor data is noisy, incomplete, or incorrectly labeled, the machine learning model's performance may be significantly affected.
- Lack of expanded connectivity support: The system currently supports only OPC UA and CSV formats. It does not yet support other common IoT protocols such as MQTT, Modbus TCP/IP, or cloud IoT platforms.
- Predictions are limited to the next morning only: The current system does not yet support predictions for different time ranges (e.g., hourly, per shift).
- LSTM implementation is still under development: Although LSTM (Long Short-Term Memory) models show promise in sequence-based predictions, they are still in the experimental stage and require more data to be effective.

6.3 Future Development Directions

To further improve the effectiveness and practical applicability of the system, several development directions are proposed:

- Integrate additional IoT protocols and cloud connectivity: Expand the system to support MQTT, LoRaWAN, or ZigBee, and synchronize data with cloud platforms (AWS IoT Core, Azure IoT Hub) for scalable storage, analytics, and remote access.
- Enhance AI algorithms: Combine Random Forest with deep learning techniques (e.g., hybrid models like LSTM + RF or CNN + RF) to better capture temporal patterns in sensor data.
- Develop Remaining Useful Life (RUL) prediction module: Estimating the remaining lifespan of a machine is critical for planning replacements and reducing unexpected failures.

- Design a mobile monitoring application: Create Android/iOS apps that allow users to monitor equipment status, receive alerts, and review maintenance logs remotely.
- Expand device coverage: Beyond motors and gearboxes, the system can be extended to monitor other industrial assets such as pumps, industrial fans, and air compressors.
- Collaborate with industry partners for pilot deployment: Work with manufacturing companies to implement and evaluate the system in a real-world factory setting, gather feedback, and improve performance based on practical needs.

REFERENCES

[1] Biendongco, "Company Homepage," https://biendongco.vn/. [Accessed 20/3/2025]

[2] Vibmaster, "Industry 4.0: Learn how it can transform your company," https://vibmaster.com.br/blog/industria-4-0-saiba-como-ela-podera-transformar-sua-empresa/. [Accessed 20/3/2025]

[3] Ngoai Ngu Cong Nghe, "Artificial Intelligence and Its Impact on Technical Fields," https://ngoainguconngnghe.edu.vn/tri-tue-nhan-tao-va-anh-huong-cua-no-den-cac-nganh-ky-thuat.html. [Accessed 20/3/2025]

[4] RealPars, "Predictive Maintenance," https://www.realpars.com/blog/predictive-maintenance [Accessed 20/3/2025]

[5] MachineMetrics, "Predictive vs. Preventative Maintenance," https://www.machinemetrics.com/blog/predictive-vs-preventative-maintenance [Accessed 20/3/2025]

[6] PCI Magazine, "Predictive Maintenance and Its Role in Improving Efficiency," https://www.pcimag.com/articles/106046-predictive-maintenance-and-its-role-in-improving-efficiency [Accessed 20/3/2025]

[7] W. Wesoe, "Predictive Analytics & Accident Prevention: How Data Can Save Lives," LinkedIn, https://www.linkedin.com/pulse/predictive-analytics-accident-prevention-how-data-can-save-lives-wesoe [Accessed 20/3/2025]

[8] Goode Sensor, "Applications of Industrial Control Sensors," https://goodesensor.com/application/industrial-control-sensor/ [Accessed 20/3/2025]

[9] Fluke, "Top 5 Industrial Applications for Vibration Sensors," https://www.fluke.com/en-us/learn/blog/vibration/top-5-industrial-applications-for-vibration-sensors# [Accessed 17/4/2025]

[10] Variohm, "Working Principle of a Pressure Sensor," https://www.variohm.com/news-media/technical-blog-archive/working-principle-of-a-pressure-sensor [Accessed 17/4/2025]

[11] Power-MI, "Displacement Sensors," https://power-mi.com/content/displacement-sensors [Accessed 17/4/2025].

[12] Monolithic Power, "Current Sensors: Types, Key Parameters, Performance, and Applications," https://www.monolithicpower.com/en/learning/resources/current-

sensors-types-key-parameters-performance-comparison-and-common-applications [Accessed 17/4/2025]

[13] Siemens Vietnam, "Introduction to S7-1200 PLC," http://siemens-vietnam.vn/gioi-thieu-plc-s7-1200/ [Accessed 17/4/2025]

[14] Siemens Industry, "SIMATIC S7-1200 - Product Overview," https://support.industry.siemens.com/cs/mdm/91696622 [Accessed 17/4/2025]

[15] Siemens AG, "COMOS_MRO_TRAINING", Internal document

[16] Siemens AG, "Training_Full_VBScript_COMOS", Internal document

[17] GeeksforGeeks, "Random Forest Regression in Python," https://www.geeksforgeeks.org/random-forest-regression-in-python/ [Accessed 23/5/2025].

[18] ResearchGate, "Structure of Random Forest Regressor Algorithm," https://www.researchgate.net/figure/Structure-of-random-forest-regressor-algorithm-The-Random-Forest-Regressor-model-is_fig1_371581286 [Accessed 23/5/2025].

[19] Scikit-learn, "Tree-based Regression," https://scikitlearn.org/stable/modules/tree.html#regression [Accessed 23/5/2025].

[20] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://link.springer.com/article/10.1023/A:1010933404324 [Accessed 23/5/2025].

[21] J. Mathew, "Understanding the Power of Long Short-Term Memory (LSTM) Algorithm," LinkedIn, https://www.linkedin.com/pulse/understanding-power-long-short-term-memory-lstm-algorithm-jose-mathew [Accessed 23/5/2025].

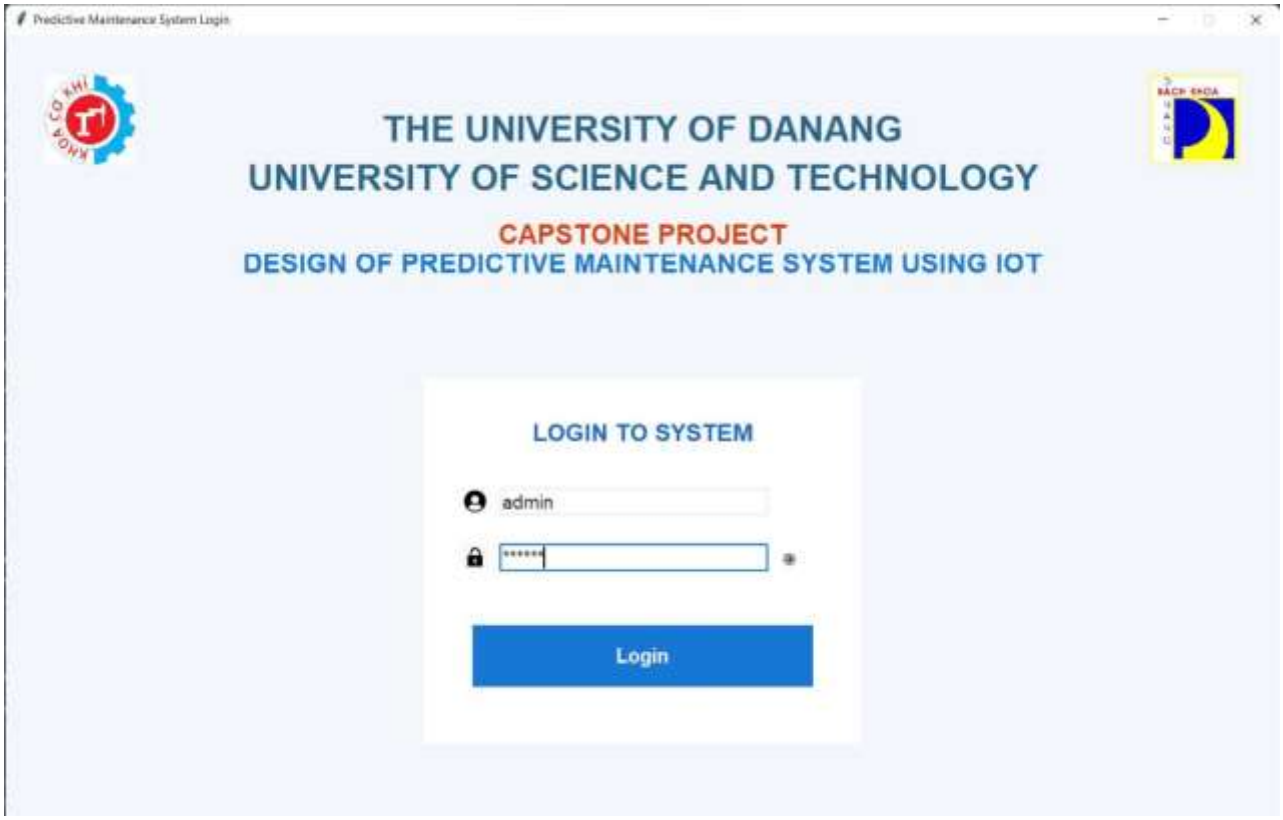
[22] Interdata.vn, "LSTM la gi? Tong quan ve mang Long Short-Term Memory," https://interdata.vn/blog/lstm-la-gi/ [Accessed 23/5/2025].

APENDIX

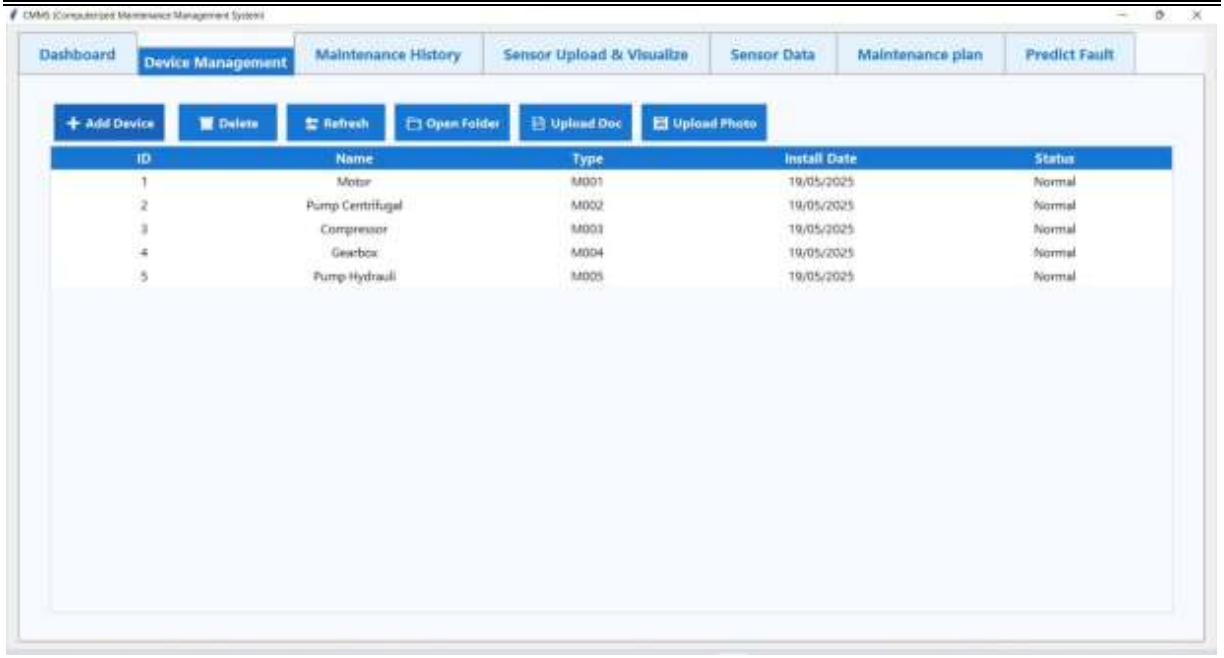
Manual

How to Use CMMS Software

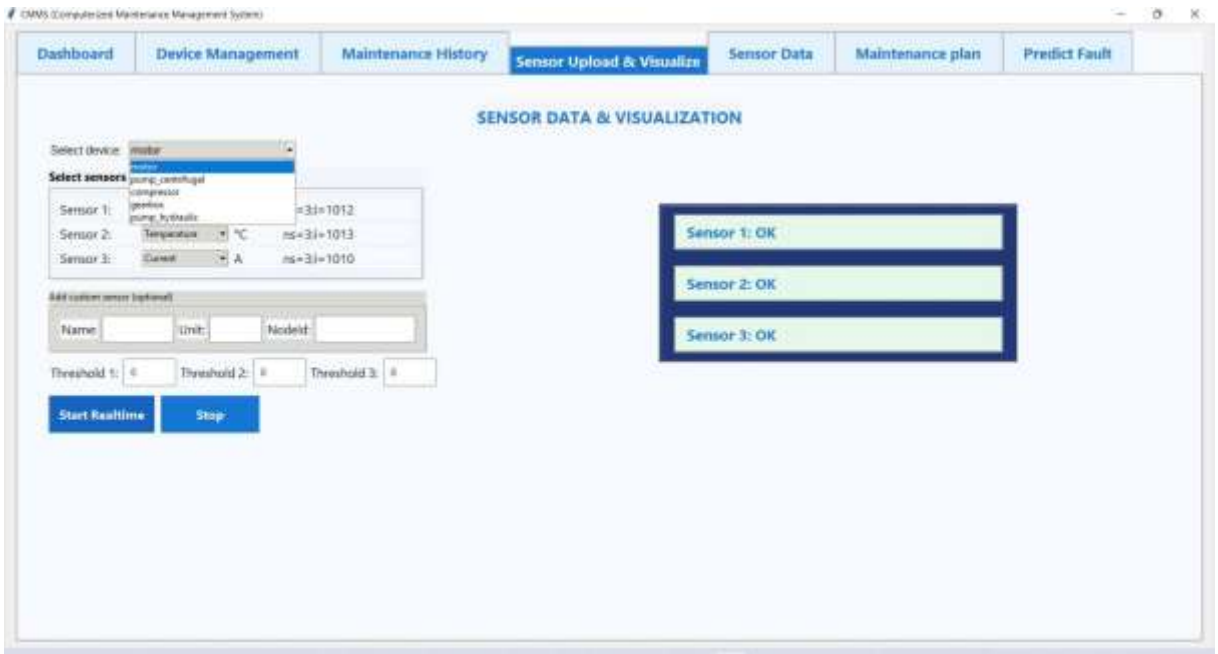
Step 1: Log in to the system (using the account provided by the administrator)



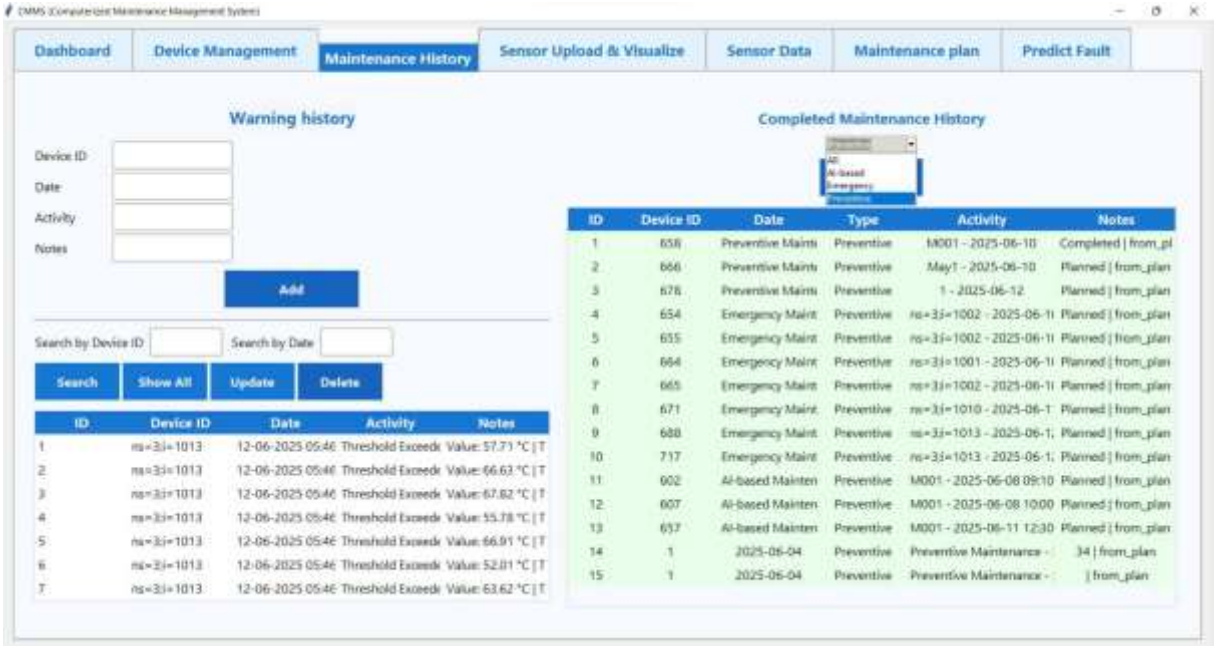
Step 2: After logging in, go to the **Devices Management** tab and add devices with the parameters shown in the image to manage and monitor the status of each device.



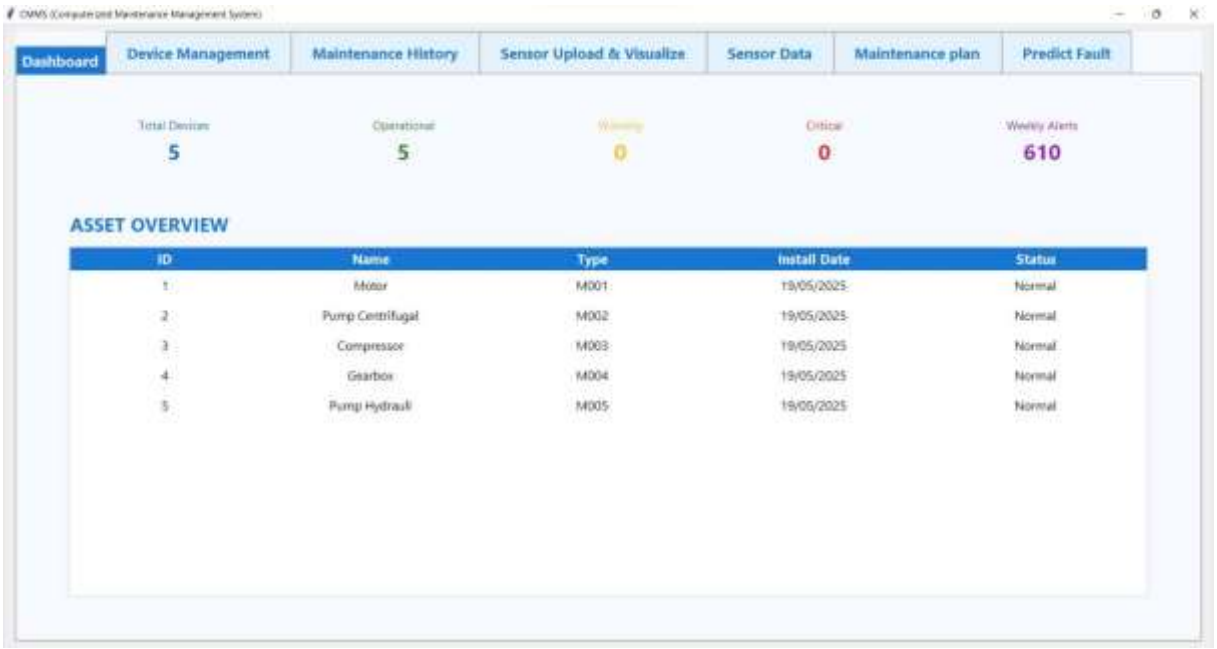
Step3: Add real-time sensor data to enable rapid monitoring and tracking from individual sensor codes, and set threshold values to trigger alerts when signals exceed defined limits.



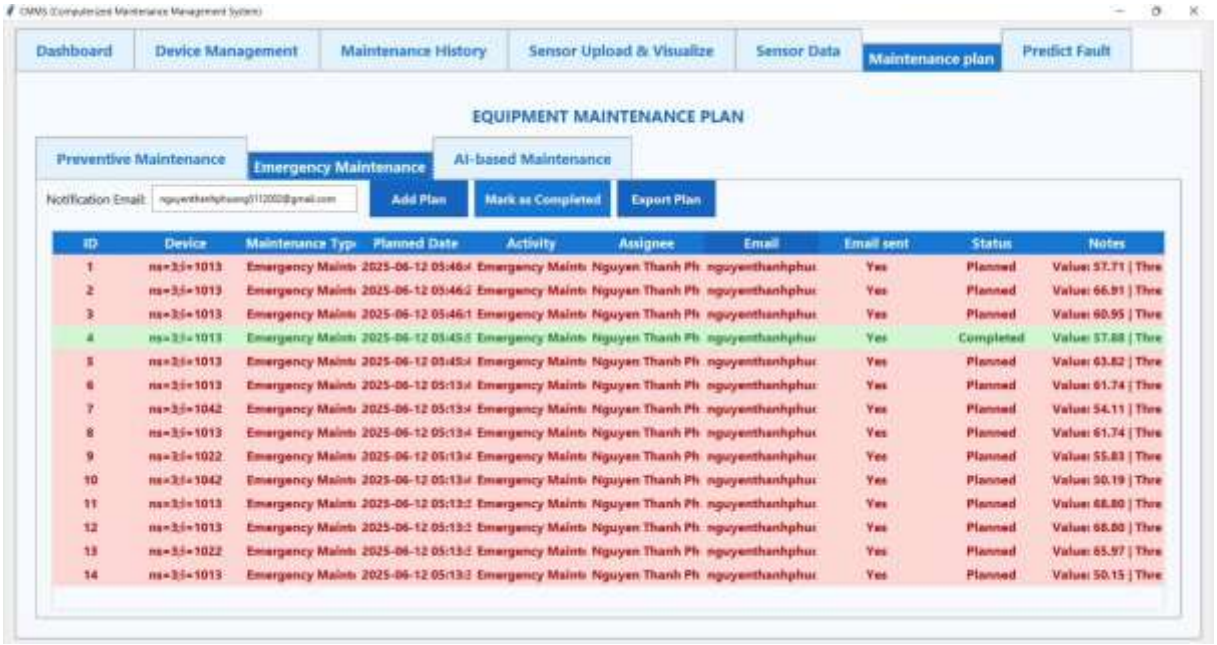
Step 4: Monitor and search maintenance history in the **Maintenance History** section. Most records are generated automatically, but you can manually create an emergency work order by adding a warning history entry.

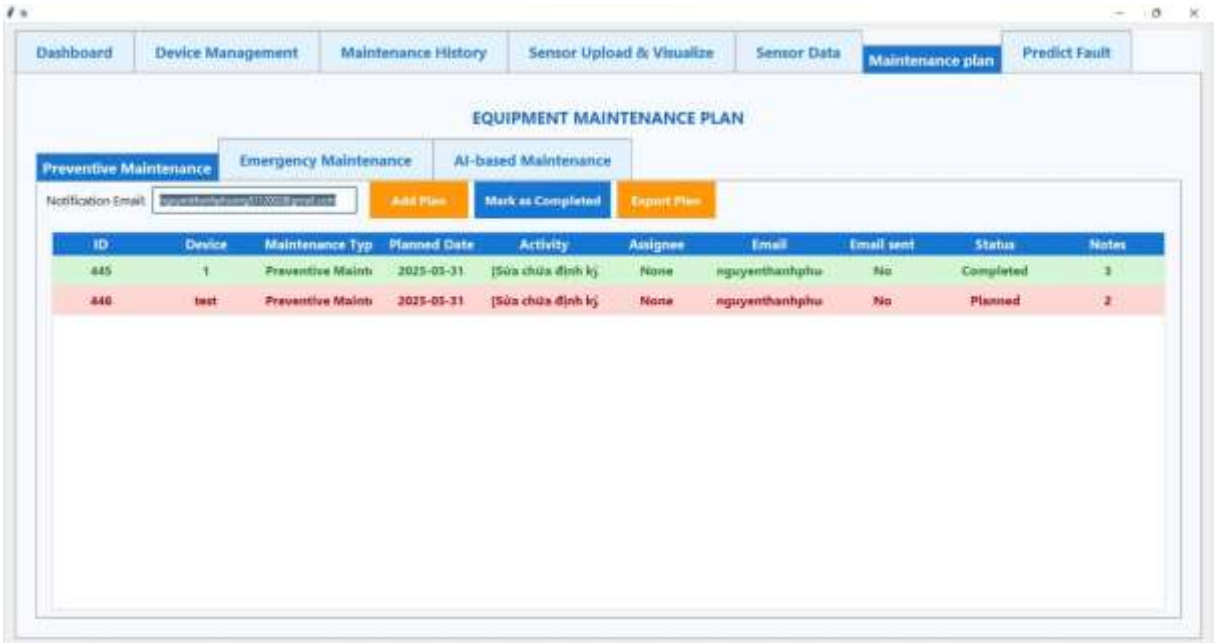


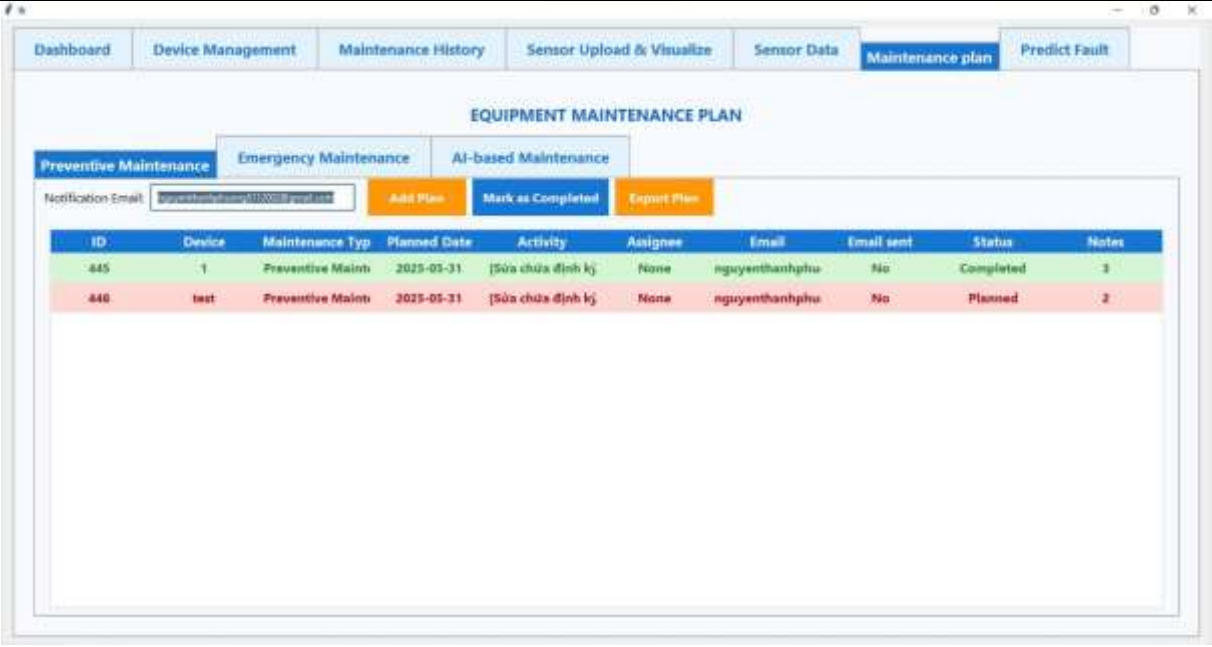
Step 5: The parameters and status of the equipment are displayed on the **Dashboard**.



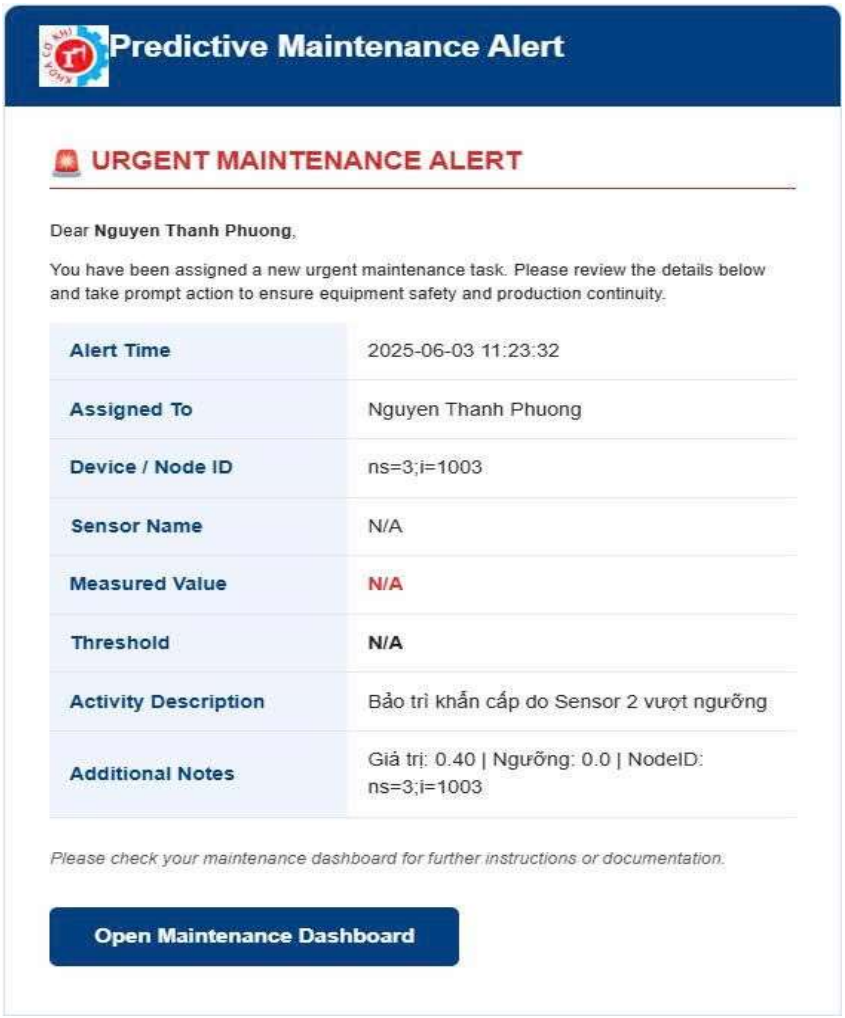
Step 6: Preventive, corrective, and AI-based predictive maintenance orders are fully displayed here. After maintenance is completed, click on the order to change its status to **Completed**.



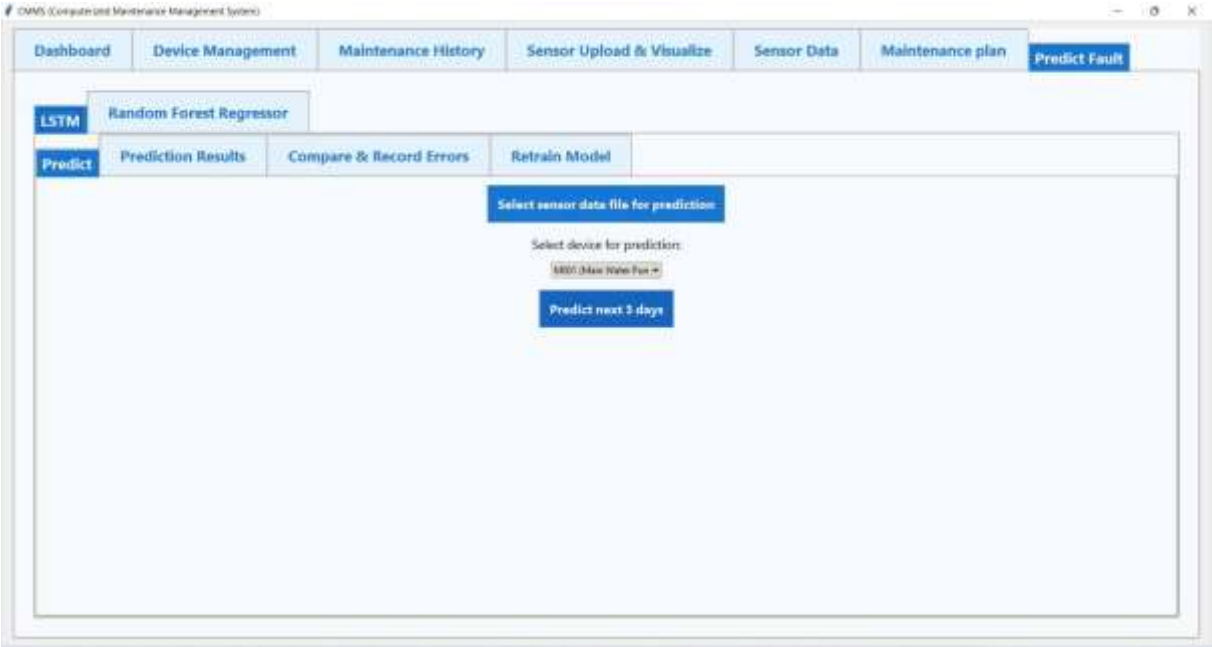




When a sensor signal exceeds the defined threshold, an email will be sent to the technician notifying of the issue and requesting a repair.



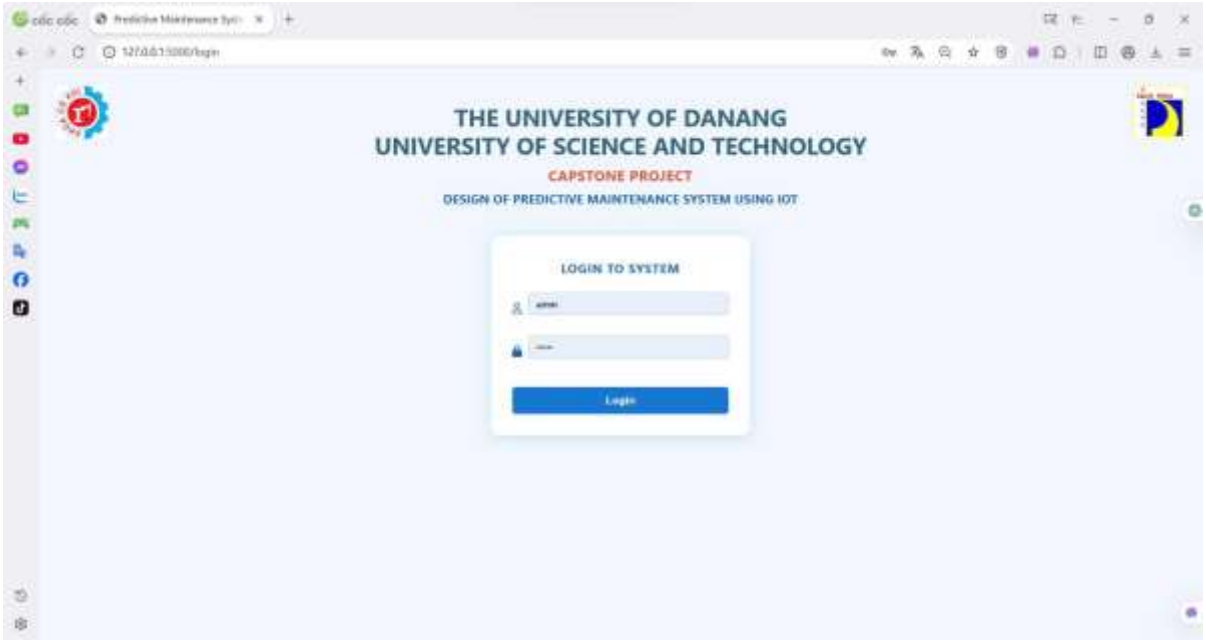
Step 7: In the **Predict Fault** tab, you can upload the latest sensor datasets. Using pre-trained models, the software will process the data with the **LSTM algorithm** and generate predictions for the next three days to determine whether the equipment is likely to fail, providing timely alerts if necessary.



User Interface

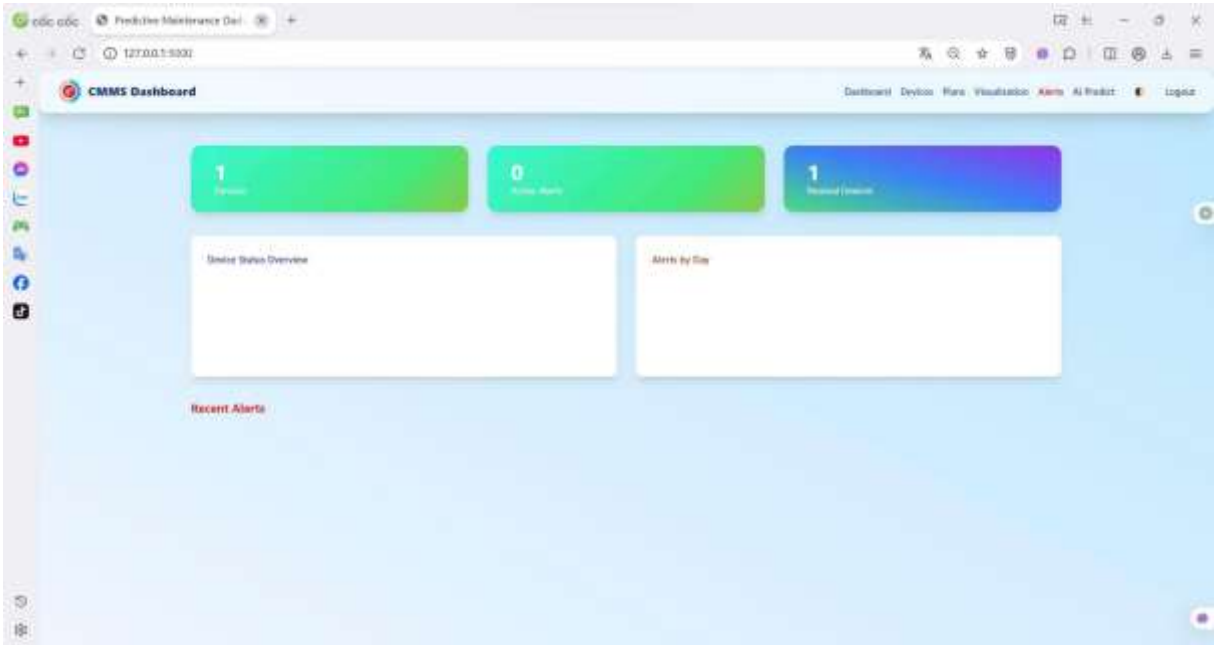
The user interface will also display the same device parameters as on the software.

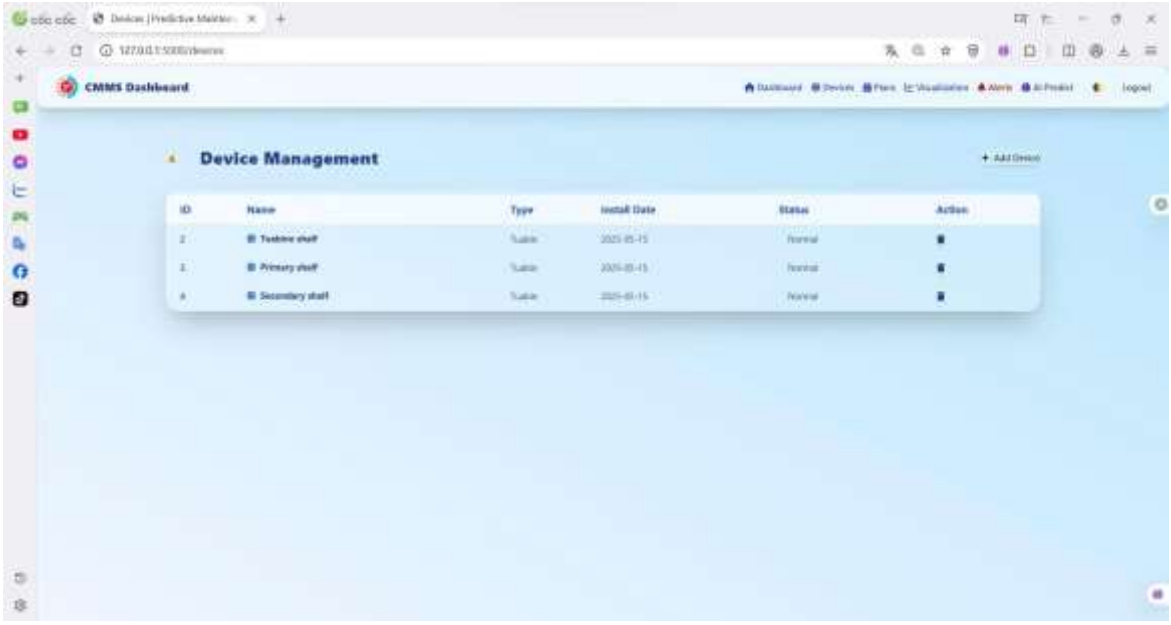
LLog in with the provided account



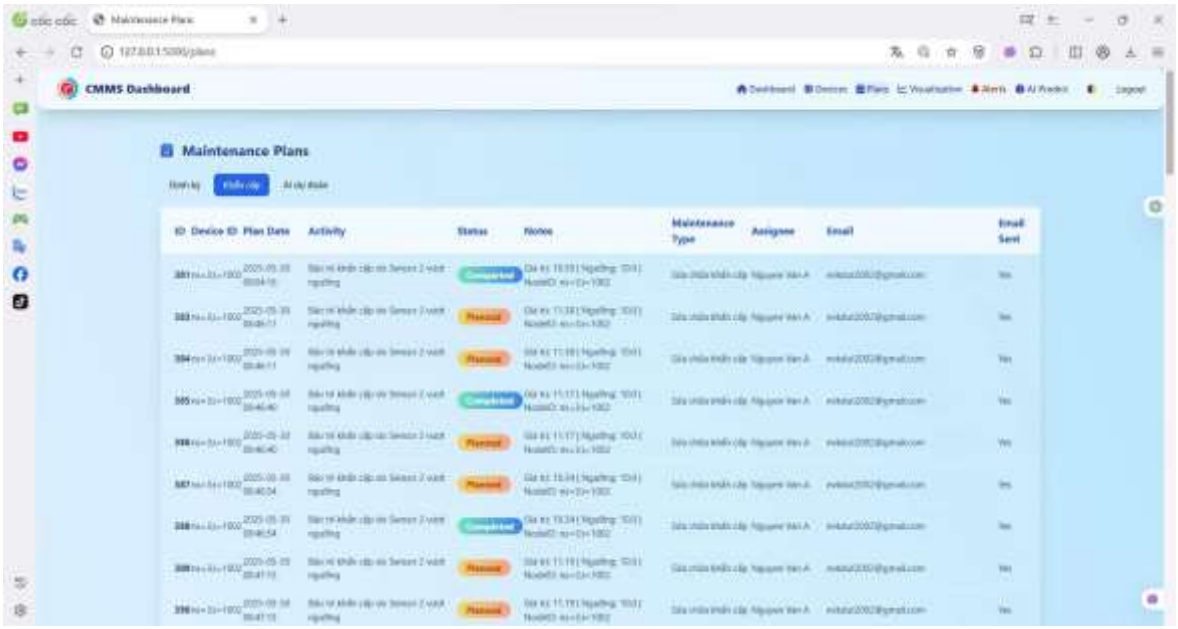
Dashboard

Dashboard displays machine specifications and error status

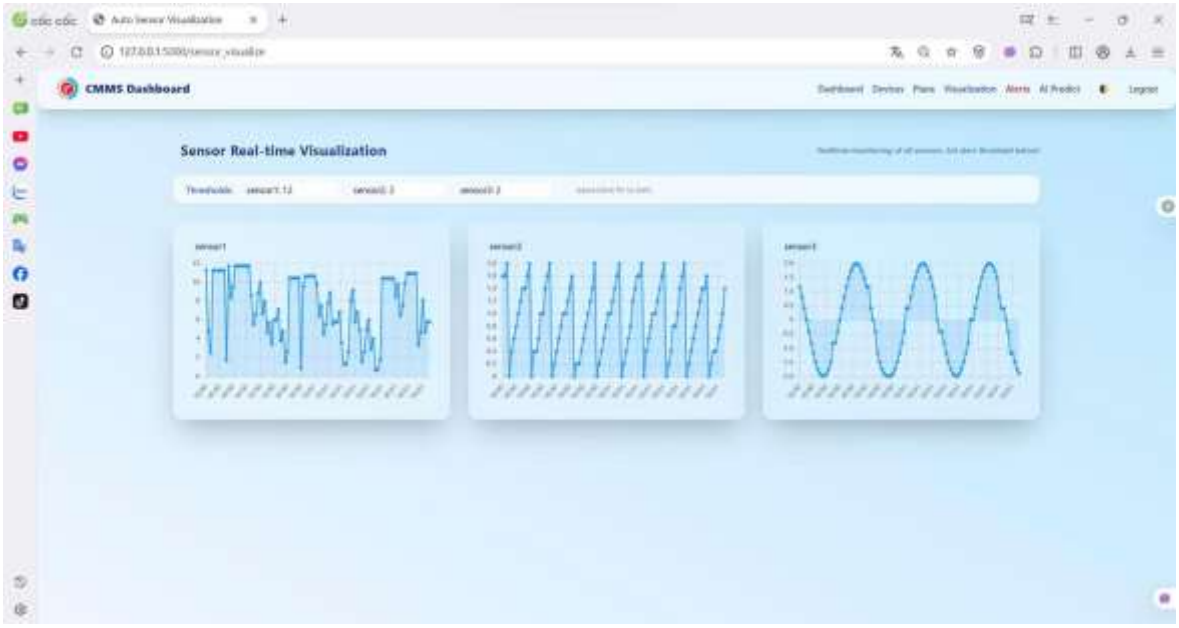




Display history of scheduled, emergency, maintenance and AI



Display 3 sensor graphs over time and set warning thresholds so that when the signal is exceeded, an email will be sent.



APENDIX 2

CODE RANDOM FOREST REGRESSOR

TRAINING

```
import pandas as pd
import os
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import joblib
import glob

# === B1: Đọc và xử lý các file CSV liên tục ===
csv_files = sorted(glob.glob("sensor_2025_05_*.csv"))

features_all = []
for i in range(len(csv_files) - 1):
    today_df = pd.read_csv(csv_files[i])
    next_df = pd.read_csv(csv_files[i + 1])
    today_df['timestamp'] = pd.to_datetime(today_df['timestamp'])
    next_df['timestamp'] = pd.to_datetime(next_df['timestamp'])

    feat = {
        'mean_vib': today_df['Vibration Sensor'].mean(),
        'max_vib': today_df['Vibration Sensor'].max(),
        'std_vib': today_df['Vibration Sensor'].std(),
        'mean_shaft': today_df['Shaft Displacement Sensor'].mean(),
        'max_shaft': today_df['Shaft Displacement Sensor'].max(),
        'std_shaft': today_df['Shaft Displacement Sensor'].std(),
```

```
        'count_alerts': (today_df['label'] == 1).sum(),
        'alert_ratio': (today_df['label'] == 1).sum() / len(today_df),
        'morning_faults_tomorrow': ((next_df['timestamp'].dt.hour < 12) & (next_df['label'] == 1)).sum()
    }
    features_all.append(feat)

df = pd.DataFrame(features_all)

# === B2: Huấn luyện mô hình ===
X = df.drop(columns=['morning_faults_tomorrow'])
y = df['morning_faults_tomorrow']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# === B3: Đánh giá ===
y_pred = model.predict(X_test)
print("MSE:", mean_squared_error(y_test, y_pred))
print("R²:", r2_score(y_test, y_pred))

# === B4: Lưu mô hình ===
joblib.dump(model, "regressor_model.pkl")
with open("regressor_features.txt", "w") as f:
    f.write(",".join(X.columns))
```

PREDICT

```
import pandas as pd
import joblib
import tkinter as tk
from tkinter import filedialog, messagebox
import matplotlib.pyplot as plt
import threading

# Ẩn cửa sổ chính Tkinter
root = tk.Tk()
root.withdraw()

# Chọn file CSV cảm biến hôm nay
file_path = filedialog.askopenfilename(
    title="Select today's data file to predict tomorrow morning and afternoon",
    filetypes=[("CSV Files", "*.csv")]
)

if not file_path:
    messagebox.showinfo("Notification", "You have not selected any files.")
else:
    try:
        # === Nạp mô hình và đặc trưng ===
        model = joblib.load("regressor_model.pkl")
        with open("regressor_features.txt", "r") as f:
            feature_names = f.read().split(',')

        # === Đọc dữ liệu và xử lý ===
        df = pd.read_csv(file_path)
```

```
df['timestamp'] = pd.to_datetime(df['timestamp'])
df['hour'] = df['timestamp'].dt.hour
data_morning = df[df['hour'] < 12]
data_afternoon = df[df['hour'] >= 12]

def extract_features(data):
    return {
        'mean_vib': data['Vibration Sensor'].mean(),
        'max_vib': data['Vibration Sensor'].max(),
        'std_vib': data['Vibration Sensor'].std(),
        'mean_shaft': data['Shaft Displacement Sensor'].mean(),
        'max_shaft': data['Shaft Displacement Sensor'].max(),
        'std_shaft': data['Shaft Displacement Sensor'].std(),
        'count_alerts': (data['label'] == 1).sum(),
        'alert_ratio': (data['label'] == 1).sum() / len(data) if len(data) > 0 else 0
    }

# === Dự đoán sáng và chiều mai ===
features_morning = pd.DataFrame([extract_features(data_morning)])[feature_names]
features_afternoon = pd.DataFrame([extract_features(data_afternoon)])[feature_names]
pred_morning = round(model.predict(features_morning)[0])
pred_afternoon = round(model.predict(features_afternoon)[0])

# === Ngày mai + % lỗi ===
ngay_hom_nay = df['timestamp'].dt.date.iloc[0]
ngay_mai = ngay_hom_nay + pd.Timedelta(days=1)
total_today = len(df)
percent_morning = (pred_morning / total_today) * 100
percent_afternoon = (pred_afternoon / total_today) * 100
```

```
    # === Tạo cảnh báo theo mức độ ===
    def generate_warning(percent):
        if percent > 10:
            return "High risk of failure – stop machine and service!"
        elif percent > 1:
            return "The machine shows signs of minor errors."
        else:
            return "Stable condition."

    warning_morning = generate_warning(percent_morning)
    warning_afternoon = generate_warning(percent_afternoon)

    # === Vẽ biểu đồ song song ===
    def show_chart():
        plt.figure(figsize=(6, 4))
        bars = plt.bar(["tomorrow morning", "tomorrow afternoon"],
                       [pred_morning, pred_afternoon],
                       color=["#FFA726", "#29B6F6"], edgecolor='black', width=0.5)

        for bar in bars:
            yval = bar.get_height()
            plt.text(bar.get_x() + bar.get_width()/2, yval + 1,
                     f"{yval:.1f}", ha='center', va='bottom', fontsize=12)

        plt.title("Predict the number of errors tomorrow", fontsize=14, weight='bold')
        plt.ylabel("Number of prediction errors", fontsize=12)
        plt.ylim(0, max(pred_morning, pred_afternoon) * 1.3 if max(pred_morning,
pred_afternoon) > 5 else 10)
```

```
            plt.grid(axis='y', linestyle='--', alpha=0.5)
            plt.tight_layout()
            plt.show()

        threading.Thread(target=show_chart).start()

        # === Hiển thị thông báo đầy đủ ===
        messagebox.showinfo(
            "Predict tomorrow's errors",
            f"Tomorrow: {ngay_mai.strftime('%d/%m/%Y')}\n\n"
            f" Morning: {pred_morning:.2f} errors
({percent_morning:.1f}%)\n{warning_morning}\n\n"
            f" Afternoon {pred_afternoon:.2f} errors
({percent_afternoon:.1f}%)\n{warning_afternoon}"
        )

        # Hiển thị song song với biểu đồ
        threading.Thread(target=show_chart).start()
        show_custom_alert()

    except Exception as e:
        messagebox.showerror("Error", f"An error occurred.:\n{e}")
```