



REPUBLIC OF TUNISIA
UNIVERSITY OF MONASTIR
FACULTY OF SCIENCES OF MONASTIR
DEPARTMENT OF COMPUTER SCIENCE



Thesis submitted for the degree of:
Professional Master in Data Science

Intelligent Platform for Industrial Equipment, Document Management, and AI Support

Completed at: **LEONI**

Submitted by: **Chebaane Ismail**

Professional Supervisor:
Mr. Nourddine Labiedh

Academic Supervisor:
Mr. Mouhamed Neji Maatouk

Defended on **11/10/2025**, before:

President: **Mr.**
Reviewer: **Mr.**

Academic Year: 2025
Location: Monastir, Tunisia

Dedication

To my dear father,

*No words can truly capture the depth of my feelings for you.
It is thanks to your unwavering love, trust, patience, sacrifices,
and constant encouragement that I have reached this stage of my life.
I can only hope that I have been worthy of your affection and faith in me.*

To my beloved mother,

*You are my reason for being, my guiding light,
the gentle lantern that illuminates my path with love and warmth.
Words fall short when I try to express the immense gratitude, pride,
and deep love I hold for you.*

*For every sacrifice you have made for my success,
I pray that God grants you health, happiness, and a long, peaceful life.*

To my siblings,

*Thank you for your constant encouragement, your advice,
and your presence in my life.
I wish each of you a future filled with success, joy, and blessings,
and I pray that God keeps us forever united.*

To my friends,

*Your friendship has been a true gift.
Your respect, kindness, and readiness to help me at every step
will forever remain in my heart.
Together, we have overcome challenges and shared victories,
and for that I am eternally grateful.*

To all those I love and who love me,

*This work is dedicated to you.
It is my humble way of honoring the trust you placed in me
and fulfilling your wish to see me succeed.*

Acknowledgments

It is with great pleasure that I dedicate this page to express my sincere gratitude to all those who have made this study possible and contributed to its completion in various ways.

“Thank God for giving me the strength to accomplish my studies.”

I would like to sincerely thank my university supervisor, Mr. Mohamed Neji Maatouk, who honored me by guiding and reviewing this work. Your dedicated involvement through your insightful directives, remarks, suggestions, and encouragement during key moments has been invaluable. I am deeply grateful for your availability whenever I sought your support. Your dedication, wise advice, and kindness have truly made this journey enjoyable. Please accept my heartfelt thanks and highest appreciation.

My special thanks also go to my professional supervisors in the company, Mr. Nourddine Labiedh, Mr. Oussama Zegnani, and Ms. Jawher Zairi, whose generous help, precious advice, clear explanations, and constructive feedback greatly supported the realization of this project.

I am also very grateful to the members of the Jury for accepting, without reservation, to evaluate this work fairly and for their valuable remarks which, with time and reflection, will undoubtedly contribute to the improvement of this thesis. Your presence on the jury is a great honor to me, and I sincerely appreciate your commitment.

Finally, I would like to thank my institute for providing me with the opportunity to acquire such valuable training.

Contents

General Introduction	1
1 Project Context and Objectives	3
1.1 Introduction	3
1.2 Presentation of LEONI Company and the LTN3 Site	3
1.2.1 LEONI Tunisia	3
1.3 Context of the Project	11
1.4 Problem Statement	11
1.5 Project Objectives	11
1.6 Existing System Study	12
1.7 Analysis of the Existing Systems	12
1.8 Critique of the Existing Systems	13
1.9 Proposed Solution	14
2 Needs Analysis and Design	16
2.1 Introduction	16
2.2 Identification of Actors	16
2.3 Identification of Requirements	17
2.3.1 Functional Requirements	17
2.3.2 Non-Functional Requirements	17
2.4 Chatbot Operation Workflow	18

2.5	Use Case Diagrams	18
2.5.1	Global Use Case Diagram	18
2.5.2	Detailed Use Case Diagram – User	20
2.6	Class Diagram	22
2.7	Dictionnaire des Données	22
2.7.1	User	23
2.7.2	Equipment	23
2.7.3	DocumentEntity	24
2.7.4	Notification	24
2.8	Sequence Diagram	24
2.8.1	Sequence Diagram for Interacting with Chatbot	25
3	Implementation of the Solution and Analysis	27
3.1	Introduction	27
3.2	Introduction	27
3.3	Intelligent Chatbot Model	27
3.4	Retrieval-Augmented Generation (RAG)	28
3.5	Chatbot Tools	29
3.5.1	General Chat Tool	29
3.5.2	Vector Search Tool	29
3.6	Initial Experiments	30
3.7	Final Adopted Solution	31
3.8	Groq API Integration	32
3.9	Use of Pandas-AI and SmartDataframe	32
3.10	Architecture and Workflow of the Final Chatbot	32
3.11	Data Preparation	34
3.11.1	Data Cleaning and Transformation	34

3.11.2 Feature Selection	34
3.12 Evaluation of Prediction Models	35
3.12.1 Linear Regression	35
3.12.2 Ridge Regression	35
3.12.3 Random Forest	35
3.12.4 Gradient Boosting	35
3.12.5 XGBoost	35
3.12.6 Model Comparison	36
3.13 Result Visualizations for XGBoost	36
3.13.1 Predictions vs Actual Values	36
3.13.2 Error Distribution	37
3.14 Deployment and Prediction	38
3.15 Conclusion	38
4 Implementation and Evaluation	39
4.1 Introduction	39
4.2 Environment and Working Tools	39
4.2.1 Hardware Environment	39
4.2.2 Software Environment	40
4.3 Main Interfaces	43
4.3.1 Main Platform Interfaces	43
4.3.2 Dashboard Interfaces	48
4.4 Conclusion	53

List of Tables

1.1	Comparison of Existing Chatbot Solutions	13
2.1	Detailed Use Case Description — Interact with Chatbot	20
2.2	Data Dictionary for User	23
2.3	Data Dictionary for Equipment	23
2.4	Data Dictionary for DocumentEntity	24
2.5	Data Dictionary for Notification	24
3.1	Comparison of the performance of different RUL prediction models	36
4.1	Code Editors and Development Environments	40
4.2	Backend and Frontend Tools	41
4.3	Python Libraries for Machine Learning and Data Analysis	41
4.4	Project Management and Collaboration Tools	43

List of Figures

1.1	Organizational chart of LEONI Tunisia	4
1.2	LEONI Sites in Tunisia	4
1.3	Clients of LEONI	5
1.4	Qualification awards of LEONI Tunisia	6
1.5	Main competitors of LEONI	6
1.6	LEONI Messadine LTN3	7
1.7	Organizational chart of LTN3	8
1.8	LAY-OUT OEM SUPPLIER GERMANY	9
2.1	Global Use Case Diagram	19
2.2	Detailed Use Case Diagram – User	20
2.3	Class Diagram of the System	22
2.4	Sequence Diagram for User Interaction with Chatbot	25
3.1	Architecture of the intelligent chatbot developed for tabular data analysis .	28
3.2	Principle of Retrieval-Augmented Generation (RAG)	29
3.3	Workflow of the General Chat tool	29
3.4	Vector Search mechanism applied to the dataset	30
3.5	Example of poor responses obtained after TinyLlama fine-tuning	31
3.6	Final architecture of the chatbot using Llama-3.3-70B with SmartDataframe	32
3.7	Overall workflow of the final chatbot system	33

3.8	Comparison between actual and predicted RUL values using XGBoost	37
3.9	Error distribution for XGBoost predictions	38
4.1	Main platform homepage	43
4.2	Chatbot interactions – Set 1	44
4.3	Chatbot interaction – Example 3	44
4.4	Chatbot interaction – Example 4	45
4.5	Chatbot interactions – Set 2	45
4.6	Chatbot interaction – Example 8	46
4.7	Login page	46
4.8	Equipment search page	46
4.9	Documents signing page (with equipment details and sign/download buttons)	47
4.10	Read-only document browsing and download page	47
4.11	Dashboard homepage with statistical overview and plant filter	48
4.12	Prediction interface for Remaining Useful Life (RUL)	49
4.13	Form for adding a new equipment	50
4.14	Equipment list with integrated error message example	50
4.15	Users list page	51
4.16	Edit user page	51
4.17	Form for creating a new user	51
4.18	Equipment preview page	52
4.19	Edit equipment page	52
4.20	Admin notifications page (listing users with more than 10 unsigned documents)	53

General Introduction

In today's industrial landscape, marked by rapid digital transformation, the competitiveness of companies increasingly depends on their ability to optimize equipment management and make effective use of data. Maintenance, document management, and equipment lifecycle monitoring have become strategic priorities, directly influencing productivity, efficiency, and long-term sustainability.

However, many organizations still face challenges due to complex manual processes, limited access to information, and a lack of integrated tools for monitoring and anticipating failures. These constraints often result in time loss, administrative overload, and difficulties in making fast and informed decisions.

To overcome these challenges, the integration of advanced technological solutions has become essential. Artificial Intelligence (AI), combined with process automation and document digitalization, offers promising opportunities. An intelligent chatbot capable of providing instant and contextualized assistance, together with a predictive model that estimates the remaining useful life of equipment, represents a valuable asset for reducing maintenance costs, increasing equipment availability, and enhancing decision-making.

This dissertation focuses on the development of an intelligent web application that integrates several key components:

- an **AI-powered chatbot assistant** to support users in real time,
- a **prediction module** to estimate the remaining useful life of equipment,
- a **digitalization and electronic signature system** for technical documents,
- and a **centralized dashboard** for managing equipment and user data.

The ultimate goal of this project is to improve operational efficiency, facilitate data accessibility, and support decision-making processes within industrial environments.

This work is structured into the following chapters:

- **Chapter 1: Project Context and Objectives** – This chapter introduces the project, its general framework, the identified problems, and the specific objectives. It also presents a critical analysis of existing systems.

- **Chapter 2: Needs Analysis and Design** – This section defines the actors and both functional and non-functional requirements, supported by various design models such as use case diagrams, class diagrams, and sequence diagrams.
- **Chapter 3: Implementation and Analysis of the Solution** – This chapter details the development of the intelligent chatbot, the predictive approaches adopted, as well as the final architecture of the system.
- **Chapter 4: Implementation and Evaluation** – The final chapter presents the working environment, the developed interfaces, and the evaluation of the obtained results.

Chapter 1

Project Context and Objectives

1.1 Introduction

In this chapter, we will first present the LEONI company along with its LTN3 site to provide the organizational and operational context of the project. Secondly, the context of the project and the problem statement will be examined to highlight the industrial challenges addressed. Then, the project objectives will be defined, clarifying the goals we aim to achieve. Following this, a study of existing systems on the market will be conducted, including an analysis and critique of their strengths and weaknesses. Finally, the chapter will conclude by introducing the proposed solution developed to overcome the identified limitations and meet the project objectives.

1.2 Presentation of LEONI Company and the LTN3 Site

1.2.1 LEONI Tunisia

a) Organization of LTN

The organizational chart of **LEONI Tunisia** is shown in Figure 1.1.

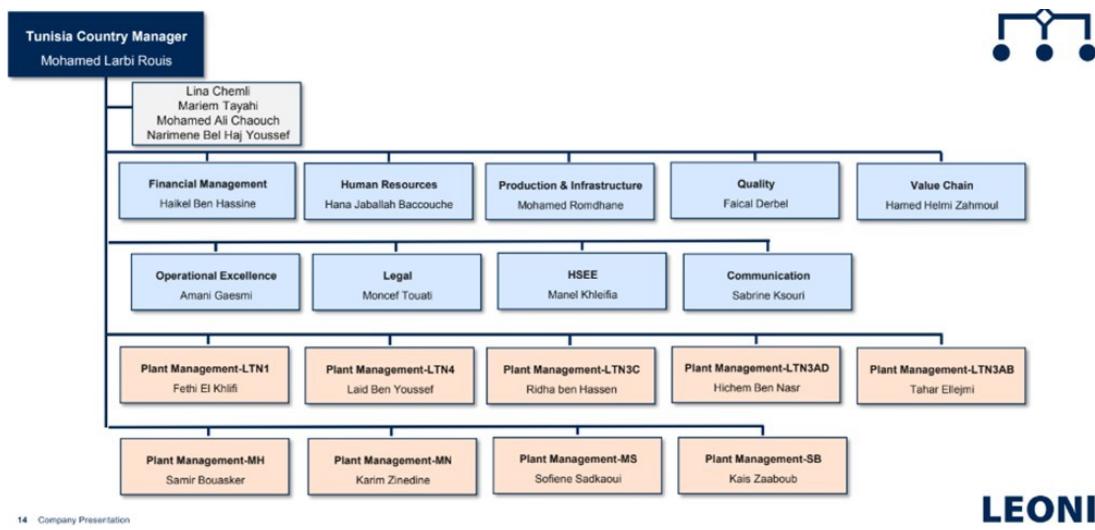


Figure 1.1: Organizational chart of LEONI Tunisia

b) LEONI Sites in Tunisia

Since 1977, LEONI Tunisia has continuously grown. It currently has five production sites illustrated in Figure 1.2: Menzel Hayet, Mateur South, Mateur North, Sidi Bouali, and Sousse.



Figure 1.2: LEONI Sites in Tunisia

c) LEONI Clients

The main clients of LEONI are:

- MERCEDES BENZ
- Volkswagen
- AUDI
- BMW
- Peugeot
- Citroën

- FIAT
- PORSCHE
- BENTLEY
- etc.



Figure 1.3: Clients of LEONI

d) LEONI Suppliers

LEONI's suppliers are:

- **International:** Siemens, Delphi, Power-Pecker, Continental.
- **National:** A.A.F. Electronics, Coficab, Tyco Electronics AMP GmbH, LEONI Roth, Hellermann Tyton GmbH, Delphi Automotive Systems, Fuba Automotive GmbH.

e) LEONI Tunisia Certifications and Awards

LEONI Tunisia is recognized as a leader in automotive wiring. Its international certifications testify to its commitment to quality and environment. The certifications obtained are:

- ISO/TS 16949 version 2009
- ISO 9001 version 2015
- ISO 14001 version 2004

Additionally, LEONI has received several qualification awards from various departments (Figure 1.4).

LEONI Tunisia National Awards



Figure 1.4: Qualification awards of LEONI Tunisia

f) Main Competitors

In a highly competitive market, LEONI strives to satisfy and retain clients to strengthen its market position. The main competitors in the automotive wiring sector are presented in Figure 1.5.

Automotive Companies in Tunisia

Cutthroat competition...

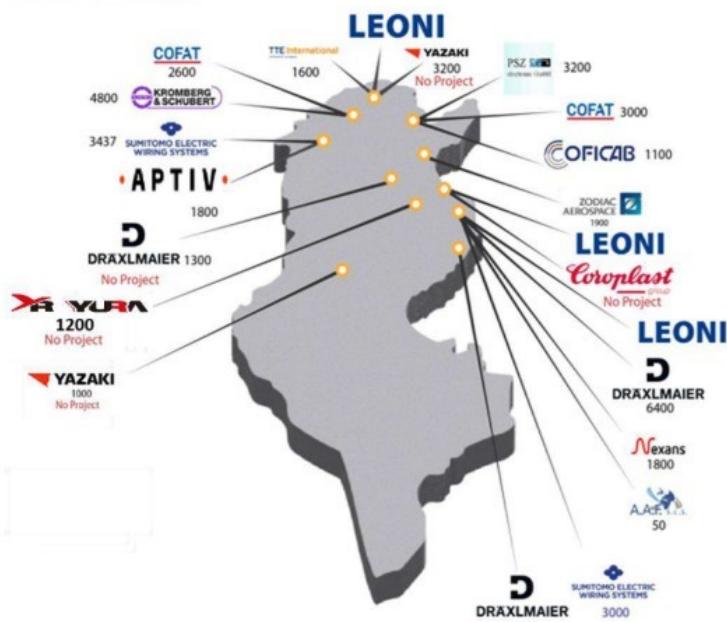


Figure 1.5: Main competitors of LEONI

1.2.2 LEONI Messadine LTN3

The internship took place at LTN 3, the third expansion established in 2004. This site includes four buildings classified by client (Figure 1.6):

- LTN 3 A: OEM Germany (various clients from Germany and the UK)
- LTN 3 B: OEM France (clients BOSCH and SCANIA)
- LTN 3 C: Mercedes Benz client
- LTN 3 D: BMW client

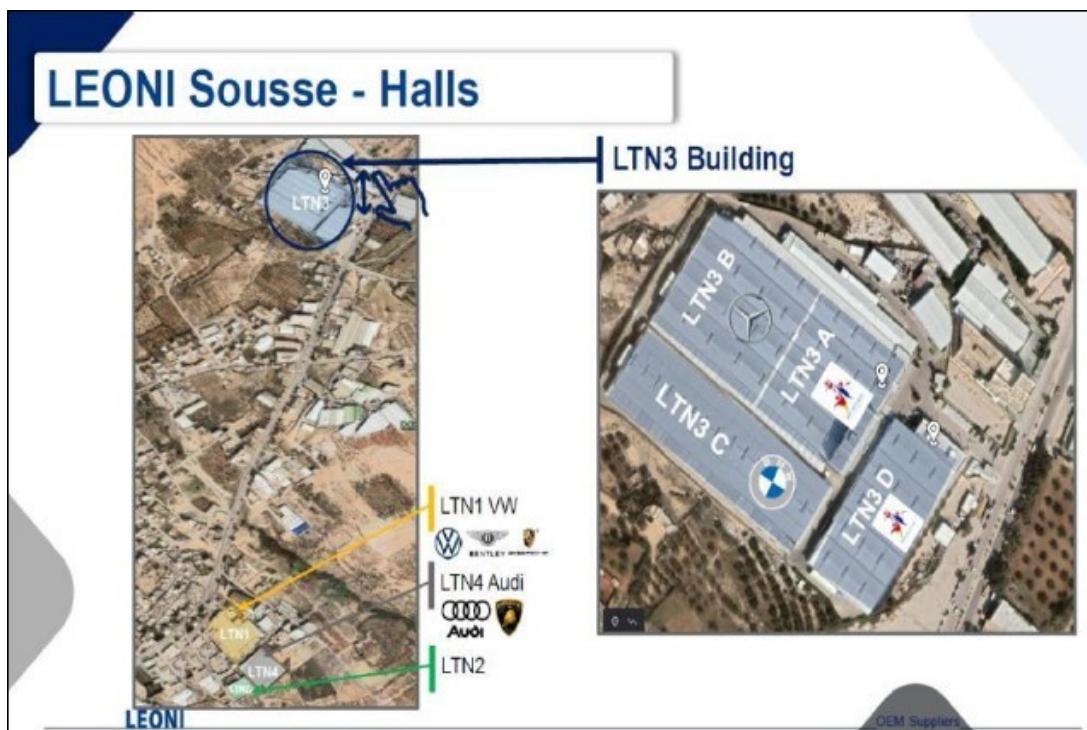


Figure 1.6: LEONI Messadine LTN3

a) Presentation of LTN3

The production unit "OEM SUPPLIER GR" LEONI Tunisia follows a policy of expanding its activity domain to win new markets and improve international competitiveness. The OEM SUPPLIER GERMANY project was launched in Sousse in 2007 at LTN 3 (LEONI Tunisia 3).

Organizational Chart of LTN3 The LEONI site "OEM SUPPLIER GR" plant section is a divisionally structured company. It has 8 departments cooperating to achieve common objectives (Figure 1.7).

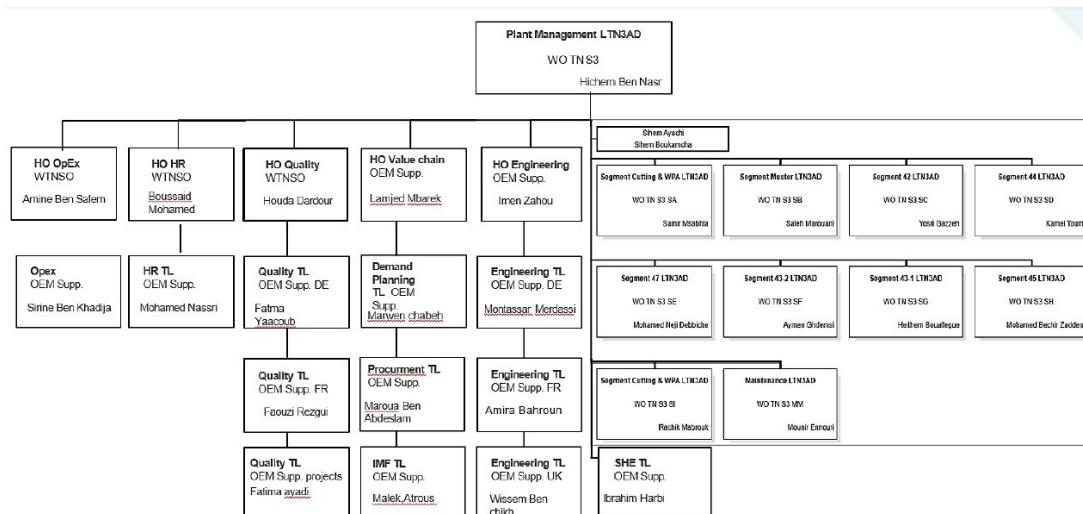


Figure 1.7: Organizational chart of LTN3

The organizational chart shows different cooperating departments, summarized below:

- **OPEX Department:** Minimize non-value-added tasks through continuous improvement and interdepartmental communication.
- **Production Department:** Ensure smooth production according to scheduling.
- **Quality Department:** Minimize defects to satisfy the client.
- **Logistics Department:** Plan manufacturing orders and manage the supply chain.
- **Engineering Department:** Composed of three services:
 - Ratio Service (IE): Methods and work rules.
 - Study Service (SE): Analysis of assembly drawings.
 - Development Service (AV): New product studies.
- **Finance Department:** Prepares financial statements, analyzes profitability, and participates in strategic decisions.
- **Technical Department:** Corrective/preventive maintenance, equipment control and installation.
- **Human Resources Department (HR):** Personnel management, recruitment, and working conditions.

Layout of LTN3: Plant Section OEM SUPPLIER GR The plant is divided into 8 sections (Figure 1.8):

- Import warehouse: storage of raw materials (coils, containers).
- SEG cutting: automatic wire cutting and crimping.
- SEG VKF: pre-assembly, wire preparation, and storage (Pagoda).
- SEG 42: cable assembly for clients Continental and Vitesco.
- SEG UK: cable assembly for clients Varroc, ZKW, Marelli.

- SEG 47: cable assembly for clients Cockpit, ZF, MEKRA.
- SEG 44: cable assembly for clients Brose, Isri, Inalfa, Magna, Seat Ateca, ZKW SCALA.
- Export warehouse: storage of finished products.

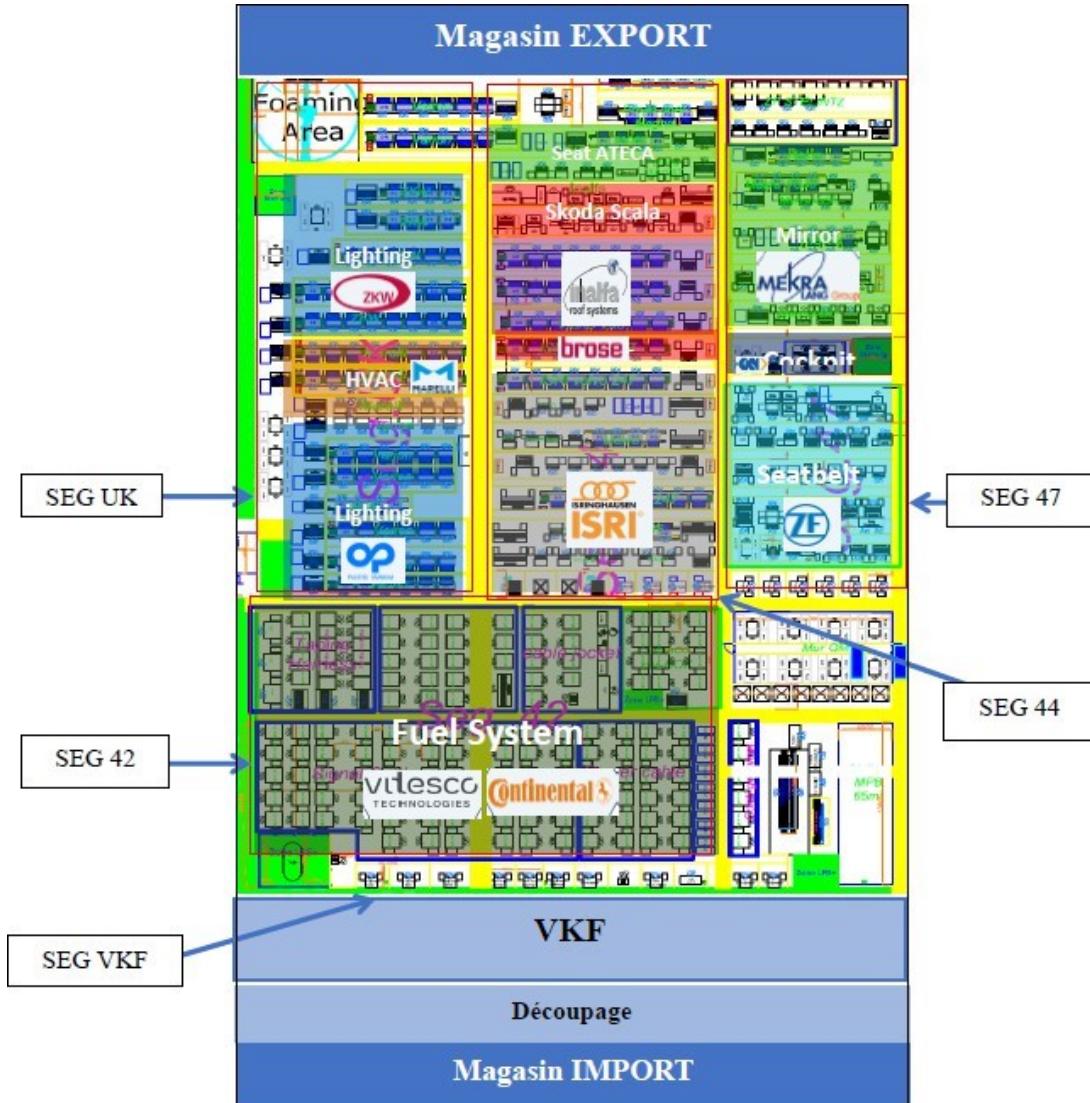


Figure 1.8: LAY-OUT OEM SUPPLIER GERMANY

1.2.3 Hosting Department: Capacity Planning, Capex and Fixed Assets

The internship was carried out in the Capacity Planning & Capex and Fixed Assets department. This department plays a strategic role in managing material resources and industrial planning in the medium and long term.

Its main mission is to anticipate, plan, and adjust production capacities according to demand forecasts, while managing investments and fixed assets.

The department is responsible for:

- Developing capacity plans in coordination with production, logistics, and engineering.
- Managing investment requests (Capex) from need to budget approval.
- Tracking fixed assets: registration, allocation, transfers, decommissioning.
- Optimizing material resource usage.
- Providing regular reporting to management.

It acts as an interface between finance, production, maintenance, and industrial management.

1.2.4 Presentation of the Production Process

1.2.4.1 Import Warehouse

Place for storing raw materials upon reception (joints, rolls, clips, wires, etc.).

1.2.4.2 Cutting

First value-added step performed by automated machines:

- Cutting cables to specified lengths.
- Partial stripping of wires.
- Crimping contacts at ends.
- Twisting according to requirements.

1.2.4.3 PAGODA before VKF

Storage warehouse for lots of cut wires:

- Lot grouping.
- Barcode registration.
- FIFO storage.

1.2.4.4 Pre-assembly

Preparation of essential sub-assemblies considered as semi-finished products.

1.2.4.5 Assembly

Final assembly of wires and sub-assemblies according to technical boards:

- Linked workstations (LAD): series production of simple products.
- Conventional chains (LAC): assembly of complex wiring.
- Carousels: medium complexity wiring, convertible to LAD double output.
- Fixed stations (PF): independent, for optional products or X-code / Pro-Vermontage.

1.3 Context of the Project

This project was carried out as part of an end-of-studies internship at **LEONI Wiring Systems**, a multinational company specialized in manufacturing automotive wiring harnesses. It was initiated to address the growing industrial need for a modern, intelligent, and user-friendly digital platform.

The solution aims to centralize equipment management, digitize technical documentation, and integrate an AI-powered chatbot for instant, context-aware assistance. It also includes a predictive model to estimate the Remaining Useful Life (RUL) of equipment, enabling proactive and optimized maintenance planning. The work was developed in close collaboration with the maintenance, production, and IT departments to ensure that it meets real operational requirements and improves overall efficiency.

1.4 Problem Statement

In many industrial environments, equipment management and technical documentation are handled through fragmented systems or manual processes. This often results in difficulties accessing critical information, delays in decision-making, and inefficiencies in maintenance planning.

At LEONI Wiring Systems, these challenges are further amplified by the need to maintain a large fleet of equipment while ensuring minimal downtime and strict compliance with production schedules. The absence of a unified platform integrating equipment tracking, document digitization, and predictive maintenance tools leads to operational gaps and increased maintenance costs.

1.5 Project Objectives

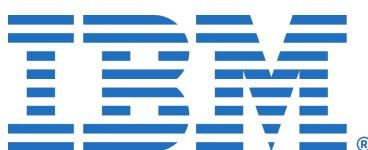
The primary objective of this project is to design and develop an intelligent web application that addresses the identified industrial challenges. Specifically, the platform aims to:

- Centralize equipment data and technical documentation in a unified digital environment.
- Integrate an AI-powered chatbot for instant and accurate assistance related to equipment and maintenance procedures.
- Implement a predictive model to estimate the Remaining Useful Life (RUL) of equipment, enabling proactive maintenance planning.
- Enhance operational efficiency, reduce downtime, and support data-driven decision-making.

1.6 Existing System Study

Understanding the current landscape of industrial equipment management and AI-assisted maintenance tools is essential for positioning our project effectively. This study reviews the main competitive systems and platforms available in the market, both commercial and open-source, focusing on their capabilities, strengths, and weaknesses.

1.7 Analysis of the Existing Systems



IBM Maximo: A leading Enterprise Asset Management (EAM) software, offering comprehensive equipment tracking, maintenance scheduling, and analytics. It supports AI integration but often requires substantial customization and investment.



SAP EAM: Another major EAM platform integrating with SAP ERP systems, providing asset lifecycle management and predictive maintenance modules. Its complexity and cost can be barriers for medium-sized sites.



Microsoft Dynamics 365 Field Service: Offers resource scheduling, asset management, and IoT integration, along with AI-powered virtual assistants. It is flexible but may require additional configuration for specific industrial domains.



Zendesk and Intercom: Widely used customer support chatbot platforms that provide automated responses, knowledge base integration, and user engagement tools. While powerful for customer service, they are not tailored for industrial equipment management.

1.8 Critique of the Existing Systems

Criteria	Zendesk	Intercom	IBM Maximo Chatbot Module
Customization	Limited	Advanced	Moderate
Natural Language Understanding	Limited	Good but sometimes inaccurate	Moderate
Cost	High	High	Very High
Ease of Deployment	Moderate	Easy	Complex
Industry Adaptability	Generic	Generic	Industry-specific
Resource Consumption	Low	Low	High
Response Speed	Fast	Fast	Moderate
Need for Model Retraining	High	High	High
Visual/Diagram Generation	No	No	Limited
Multilingual Support	Basic	Basic	Moderate

Table 1.1: Comparison of Existing Chatbot Solutions

Key limitations identified in current systems include:

- **High Cost and Resource Usage:** Enterprise platforms like IBM Maximo and SAP EAM require significant investments in licenses, hardware, and dedicated IT staff.
- **Lack of Industry-Specific AI Assistance:** Generic chatbots such as Zendesk and Intercom offer broad functionality but do not provide domain-specific expertise or predictive maintenance capabilities critical for industrial environments.
- **Limited Visual Interaction:** Most existing chatbots do not generate dynamic visual content such as charts or diagrams, restricting their ability to communicate complex data insights effectively.
- **Basic Multilingual Support:** Many solutions support only limited languages or lack robust multilingual natural language understanding, impacting accessibility in global industrial contexts.
- **Complex Deployment and Maintenance:** Many existing solutions demand complex setup processes and continuous model retraining or fine-tuning whenever new data or knowledge is added.

- **Fragmented Systems:** Current tools often specialize in either equipment management or AI assistance, rarely delivering both in an integrated, lightweight platform.

1.9 Proposed Solution

Our project introduces an integrated, intelligent web application specifically designed for industrial equipment management, technical document digitization, and AI-driven user support.

Key features and benefits include:

- **AI Chatbot with Dynamic Adaptability:** The chatbot leverages an advanced language model hosted online, providing fast and accurate responses without relying on local computing resources. Crucially, it can dynamically adapt to updated or new data without requiring retraining or fine-tuning, ensuring seamless integration of new information and continuous improvement.
- **Visual and Multilingual Capabilities:** Unlike many existing systems, our chatbot can generate dynamic visual outputs such as charts and diagrams to better illustrate data insights. It also supports advanced multilingual natural language understanding, making it accessible to diverse users in industrial environments.
- **High-Accuracy Prediction Model:** A dedicated module estimates the Remaining Useful Life (RUL) of equipment with approximately 89% accuracy, enabling proactive and efficient maintenance scheduling.
- **Lightweight and Scalable Architecture:** The AI system is hosted on cloud servers rather than on local hardware, which eliminates the need for expensive or powerful equipment on-site. This approach allows the platform to easily support many users at once without performance issues. As a result, the solution is both cost-effective and scalable, able to grow and handle increased demand as needed.
- **Centralized Data Management:** The platform consolidates equipment details and maintenance histories into a single system to improve data accessibility and support informed decision-making.
- **User-Friendly Design:** Intuitive interfaces tailored for operators and engineers minimize training time and promote widespread adoption.

This solution addresses the limitations of existing platforms by combining advanced AI capabilities with industry-specific functionality, cost efficiency, and simplified deployment and maintenance.

Conclusion

This chapter has provided a comprehensive overview of the project context by introducing the LEONI company and the LTN3 site where the internship took place. We have outlined the key industrial challenges motivating the project and defined clear objectives to address these issues. The study and critique of existing systems have highlighted the gaps and limitations in current solutions. Finally, the chapter introduced the proposed intelligent web application designed to integrate equipment management, digital documentation, and AI-assisted maintenance, setting the stage for the detailed development and implementation described in the following chapters.

Chapter 2

Needs Analysis and Design

2.1 Introduction

This chapter focuses on the needs analysis and design of the platform. It identifies the main actors—users, administrators, and the AI Chatbot—and their roles and interactions. Functional and non-functional requirements are detailed to ensure the system meets performance, usability, and security standards. The chapter also presents workflows, UML diagrams (use case, class, and sequence), and a data dictionary to clearly define system behavior, structure, and data attributes. This analysis provides a solid foundation for system implementation.

2.2 Identification of Actors

The system involves three main actors:

- **User:** The end-user can:
 - Navigate and search for equipment information.
 - View dashboards displaying relevant metrics and statistics.
 - Download or sign documents.
 - Access the prediction module to estimate equipment Remaining Useful Life (RUL).
 - Interact with the chatbot for equipment-related inquiries.
- **Admin:** The administrator has all the user capabilities plus additional management features:
 - Manage user accounts (add, remove, update).
 - Manage equipment records (add, remove, update).
 - Upload, associate, and manage documents linked to equipment.
 - Access notifications when users have more than 10 unsigned documents.

- Use the prediction module and chatbot.
- **AI Chatbot:** An intelligent assistant integrated into the platform that:
 - Answers questions using a CSV database containing equipment information.
 - Supports multiple languages for diverse user needs.
 - Can generate and return diagrams when relevant.

2.3 Identification of Requirements

2.3.1 Functional Requirements

The functional requirements of the system are as follows:

- **Equipment Management:** Ability to add, update, delete, and view equipment data.
- **Document Management:** Upload, view, sign, and track documents linked to equipment.
- **User Management (Admin only):** Create, update, and remove user accounts.
- **Dashboard Visualization:** Display statistics, KPIs, and monitoring charts.
- **Prediction Module:** Estimate the Remaining Useful Life (RUL) of equipment.
- **Notifications:** Alert admin when users have more than 10 unsigned documents.
- **Chatbot Interaction:** Provide answers, multilingual support, and diagrams from CSV data.

2.3.2 Non-Functional Requirements

The non-functional requirements include:

- **Performance:** The system should respond to user actions in less than 2 seconds on average.
- **Scalability:** Support simultaneous access by multiple users without performance degradation.
- **Security:** Ensure secure authentication and protect sensitive data.
- **Usability:** Provide an intuitive and user-friendly interface for both users and admins.
- **Compatibility:** Accessible via major web browsers and mobile devices.
- **Multilingual Support:** The chatbot and UI must handle multiple languages seamlessly.

2.4 Chatbot Operation Workflow

The chatbot is a key feature of the platform, enabling intelligent, multilingual interaction with users. Its operation is structured as follows:

1. **Data Loading:** The chatbot uses a CSV file (`latest_dataset.csv`) containing updated equipment data. This file is loaded into a Pandas DataFrame when the application starts.
2. **Request Handling:** When a user submits a question via the interface, it is sent as a JSON request to the Flask backend through the `/chat` API endpoint.
3. **Custom Prompt Generation:** The backend creates a structured prompt containing:
 - The user's question.
 - The DataFrame columns.
 - Instructions to return Python code for analysis or visualization only.
4. **Language Model Processing:** The prompt is processed by a large language model through the Groq API. The model generates code to answer the query or create diagrams.
5. **Execution and Diagram Generation:** Using the PandasAI `SmartDataframe`, the generated code is executed. If a chart is created, it is saved in the `exports/charts` directory.
6. **Response Delivery:**
 - Text results are returned directly to the user.
 - Image results return a URL to the generated diagram.
7. **Multilingual Support:** The model supports multiple languages, allowing users to interact in their preferred language without retraining the chatbot.

This design allows the chatbot to adapt dynamically to updated CSV data without the need for retraining, ensuring continuous relevance and accuracy.

2.5 Use Case Diagrams

UML use case diagrams describe the main functionalities of the system and show the interactions between the system and its actors. They provide a simple and clear way to visualize the possible actions for each actor.

2.5.1 Global Use Case Diagram

The global use case diagram presents an overview of the interactions between all actors and the system. It highlights the different use cases that meet the defined functional

requirements.

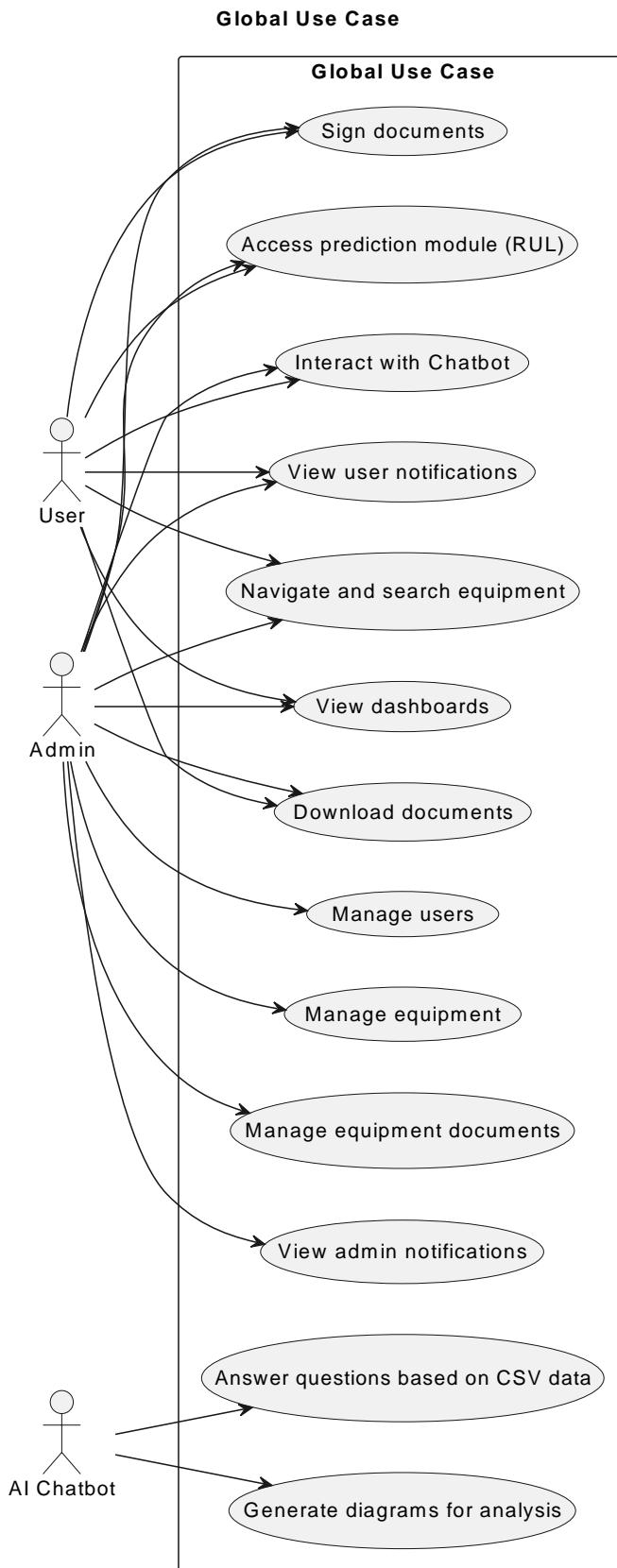


Figure 2.1: Global Use Case Diagram

2.5.2 Detailed Use Case Diagram – User

The detailed use case diagram for the User actor presents the specific interactions that a User can perform within the system. It includes functionalities such as navigating equipment, viewing dashboards, using the prediction module, and interacting with the chatbot.

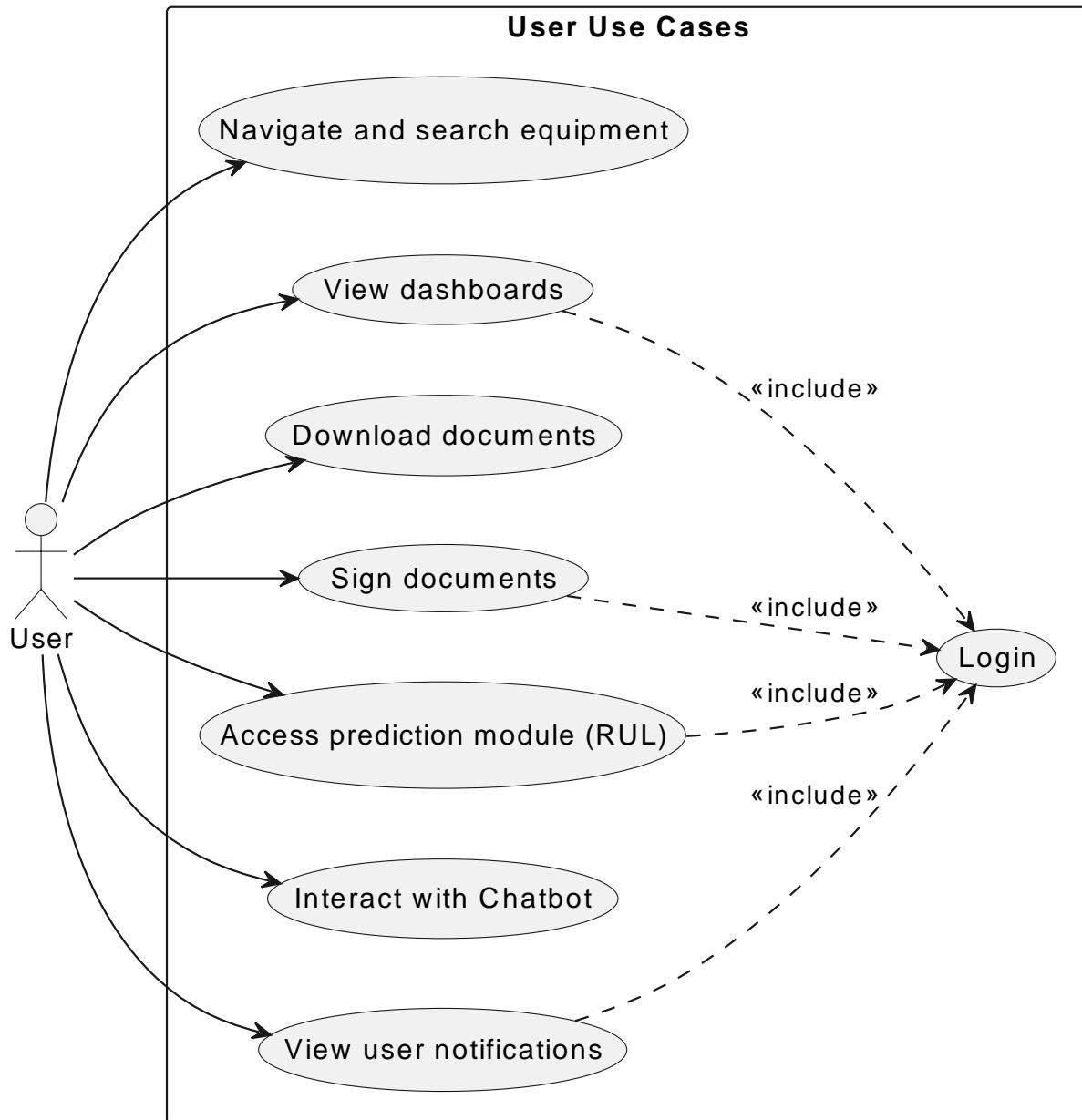


Figure 2.2: Detailed Use Case Diagram – User

Table 2.1: Detailed Use Case Description — Interact with Chatbot

Title	Interact with Chatbot
-------	-----------------------

Description	This use case allows the user to ask questions about equipment and generate diagrams for analysis. The system interprets the query and returns either textual answers or charts.
Actor	User
Precondition	The system has access to the latest equipment data in CSV format. The chatbot service is running and accessible.
Trigger	The user opens the chatbot interface and submits a question.
Main Scenario	<ol style="list-style-type: none"> 1. User enters a question in the chatbot interface. 2. System processes the query and generates the response. 3. System returns either text or a diagram/chart. 4. User views the response.
Alternative Scenario	<ul style="list-style-type: none"> • CSV file is outdated or missing: system shows an error. • Diagram generation fails: system returns only textual response.
Postconditions	User receives an answer or a chart representing the requested analysis.
Benefits	Enables users to get quick insights from equipment data and generate visual analysis without manual processing.

2.6 Class Diagram

The class diagram illustrates the main classes in the system, their attributes, methods, and the relationships between them. It provides a static view of the system structure, showing how objects interact and depend on each other.

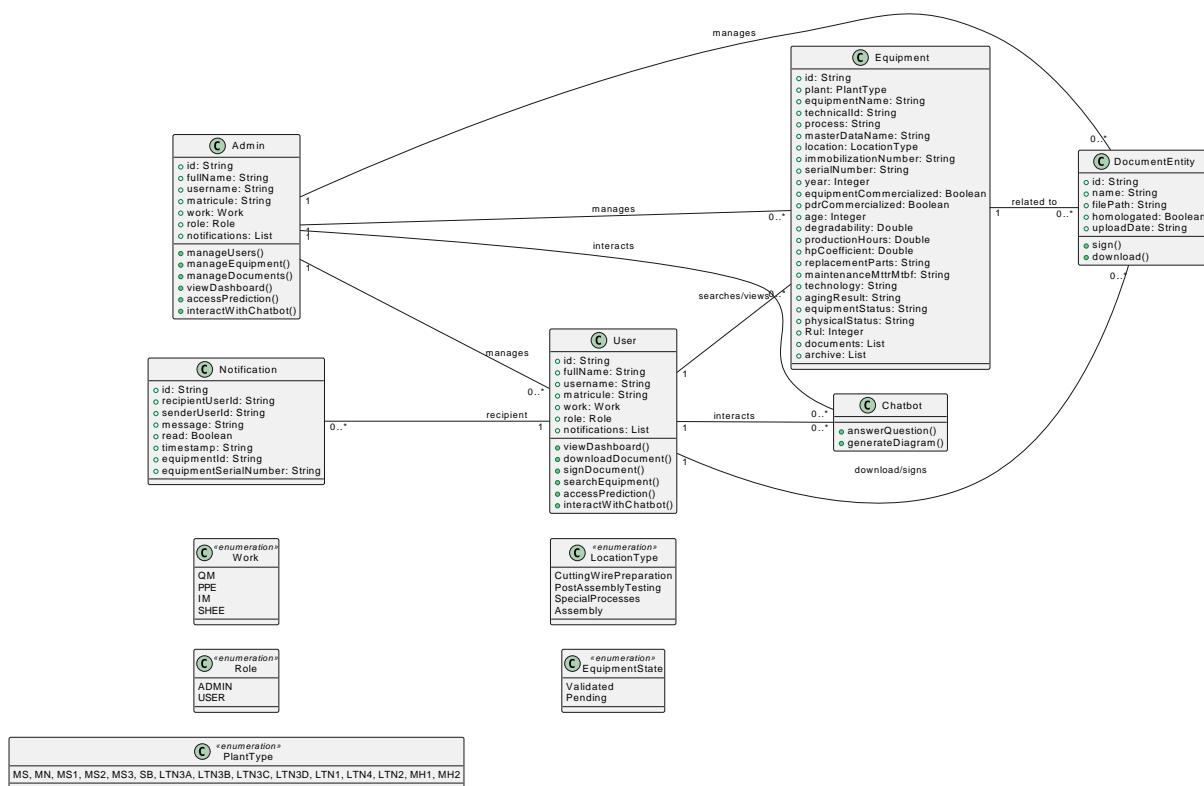


Figure 2.3: Class Diagram of the System

2.7 Dictionnaire des Données

The data dictionary provides a detailed description of each attribute in the system's database. It includes the data type, the role of each attribute, and any constraints. This guide is essential for developers and system designers to understand the database structure and operation.

2.7.1 User

Attribute	Type	Description
id	String	Unique identifier of the user (primary key)
fullName	String	Full name of the user
username	String	Username for authentication
matricule	String	Employee or system matricule number
work	Enum (Work)	User's work type (QM, PPE, IM, SHEE)
role	Enum (Role)	Role of the user (ADMIN, USER)
notifications	List<Notification>	List of notifications received by the user

Table 2.2: Data Dictionary for User

2.7.2 Equipment

Attribute	Type	Description
id	String	Unique identifier of the equipment (primary key)
plant	Enum (Plant-Type)	Type of plant (MS, MN, etc.)
equipmentName	String	Name of the equipment
technicalId	String	Technical identifier
process	String	Manufacturing or production process
masterDataName	String	Master data reference name
location	Enum (LocationType)	Location of equipment within the plant
serialNumber	String	Equipment serial number
year	Integer	Year of production
equipmentStatus	String	Current status (e.g., Validated, Pending)
Rul	Integer	Remaining Useful Life of the equipment
documents	List<DocumentEntry>	List of documents related to the equipment
archive	List<DocumentEntry>	Archived documents

Table 2.3: Data Dictionary for Equipment

2.7.3 DocumentEntity

Attribute	Type	Description
id	String	Unique identifier of the document (primary key)
name	String	Document name
filePath	String	Path or location of the document file
homologated	Boolean	Whether the document is approved/homologated
uploadDate	DateTime	Date and time when the document was uploaded

Table 2.4: Data Dictionary for DocumentEntity

2.7.4 Notification

Attribute	Type	Description
id	String	Unique identifier of the notification (primary key)
message	String	Content of the notification
read	Boolean	Whether the notification has been read
timestamp	DateTime	Time when the notification was created
equipmentId	String	Optional reference to related equipment
equipmentSerialNumber	String	Optional serial number of related equipment

Table 2.5: Data Dictionary for Notification

2.8 Sequence Diagram

Sequence diagrams illustrate the dynamic interactions between users and the system, highlighting the chronological flow of operations for each process. They help visualize the key steps of platform functionalities, such as interacting with the AI Chatbot, querying equipment data, and generating reports.

2.8.1 Sequence Diagram for Interacting with Chatbot

The sequence diagram below presents the process of a user interacting with the AI Chatbot. It shows the interactions between the user, the system, the CSV dataset, and the Chatbot service.

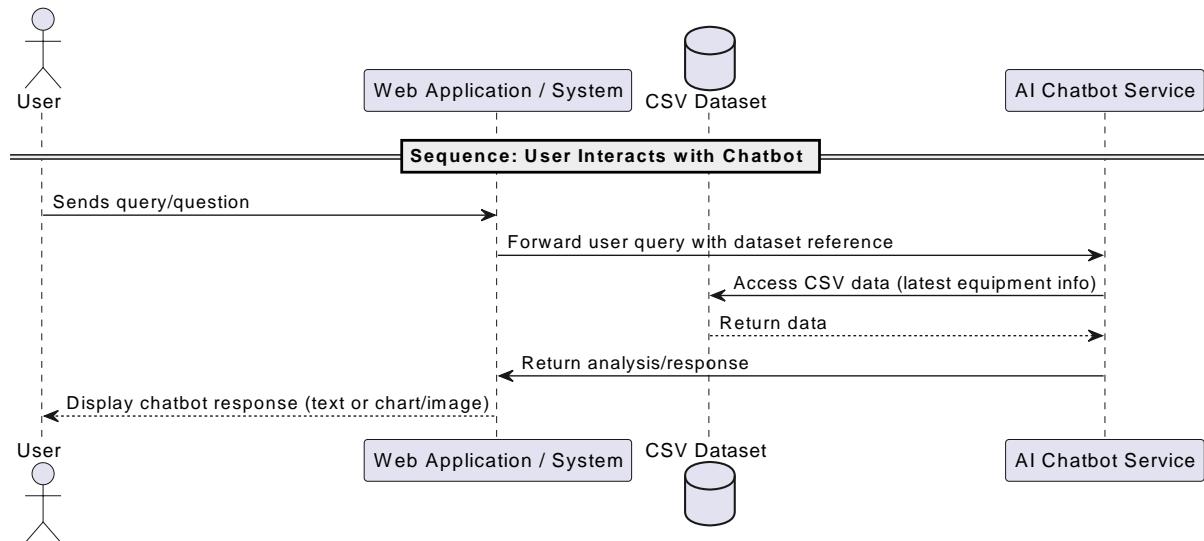


Figure 2.4: Sequence Diagram for User Interaction with Chatbot

Main Actors:

- **User**: Sends a query regarding equipment, predictions, or documents.
- **System / Web Application**: Receives the query, forwards it to the Chatbot, and displays the response.
- **CSV Dataset**: Stores the latest equipment information, accessed by the Chatbot.
- **AI Chatbot Service**: Processes the user query, performs data analysis, and generates text or chart responses.

Scenario Steps:

1. The user sends a question or query to the system.
2. The system forwards the query to the Chatbot service with reference to the dataset.
3. The Chatbot accesses the CSV dataset to retrieve relevant information.
4. The Chatbot analyzes the data and generates a response (text or chart/image).
5. The system returns the response to the user interface.
6. The user views the answer provided by the Chatbot.

Conclusion

This chapter has detailed the system's actors, requirements, and interactions, highlighting both functional and structural aspects. UML diagrams and the data dictionary offer a clear representation of system behavior and data organization. The analysis ensures that key functionalities, including Chatbot interactions and equipment management, are well-understood, supporting the design of a scalable, secure, and user-friendly platform.

Chapter 3

Implementation of the Solution and Analysis

3.1 Introduction

3.2 Introduction

This chapter presents the practical implementation of the proposed solution. It begins by describing the hardware and software environment that supported development, including the main programming tools, frameworks, and libraries used. Next, it highlights the key interfaces of the platform through screenshots, with particular emphasis on the analytical dashboards and data science components that demonstrate the system's predictive capabilities.

3.3 Intelligent Chatbot Model

The intelligent chatbot model developed in this project enables users to ask questions in natural language about a dataset in CSV format and receive accurate answers derived directly from the data.

The chatbot's architecture combines a large language model (LLM) with specialized tools that interpret user queries, execute computations on the dataset, and return results in a readable format (text or charts).

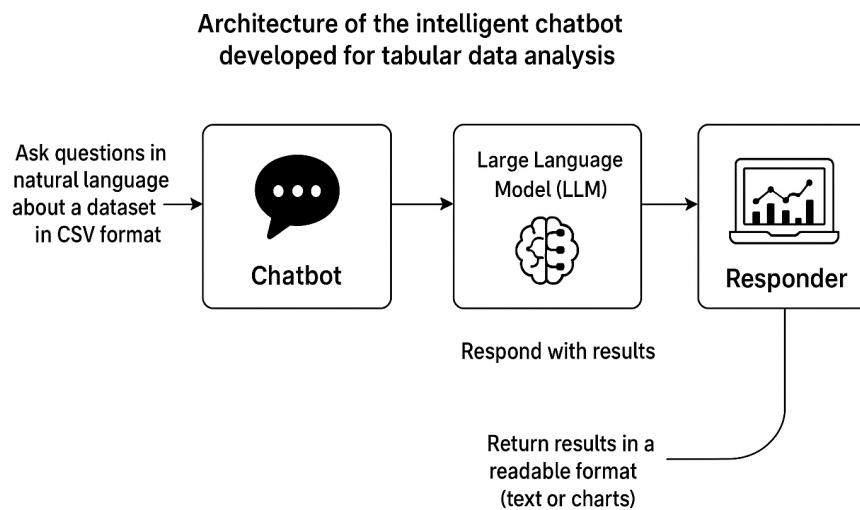


Figure 3.1: Architecture of the intelligent chatbot developed for tabular data analysis

3.4 Retrieval-Augmented Generation (RAG)

Another approach explored was *Retrieval-Augmented Generation (RAG)*, which combines text generation by the LLM with retrieval of relevant data from external sources (in this case, a CSV file).

The RAG pipeline in our context followed these steps:

1. Generate embeddings for each row of the dataset,
2. Search for the rows most semantically similar to the user's question,
3. Enrich the LLM's response with the retrieved rows.

Although promising, this approach performed poorly on structured tabular data. Embedding-based similarity search worked better on textual documents (such as PDFs) but struggled with numeric and categorical values.

Retrieval-Augmented Generation (RAG)

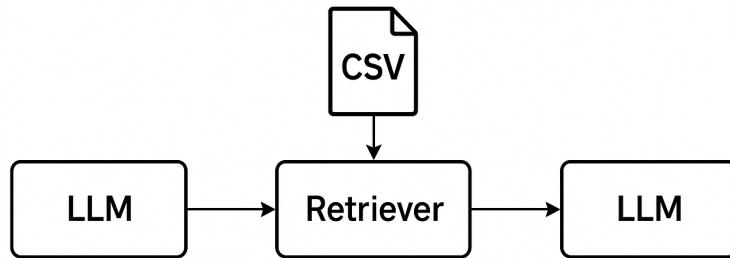


Figure 3.2: Principle of Retrieval-Augmented Generation (RAG)

3.5 Chatbot Tools

3.5.1 General Chat Tool

The *General Chat* tool was designed to handle general or simple queries that did not require advanced search. For example, it could explain the meaning of a column, describe the overall content of the dataset, or provide basic insights.

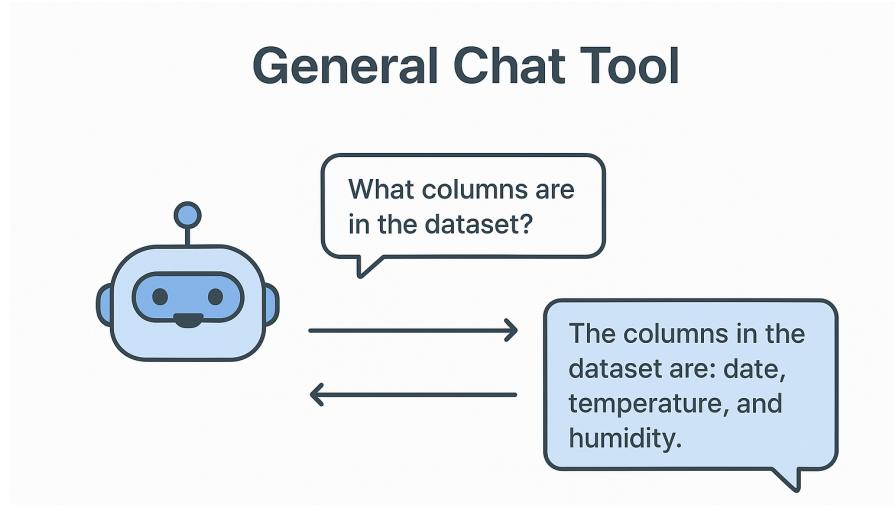


Figure 3.3: Workflow of the General Chat tool

3.5.2 Vector Search Tool

The *Vector Search* tool aimed to find dataset rows that were semantically closest to the user's question. The process involved:

1. Converting dataset rows into embeddings using a SentenceTransformer model,
2. Performing cosine similarity search to retrieve the most relevant rows,
3. Returning these rows as candidate answers.

Despite being theoretically useful, this method was not effective on tabular data. The retrieved rows often lacked precision and did not match user expectations.

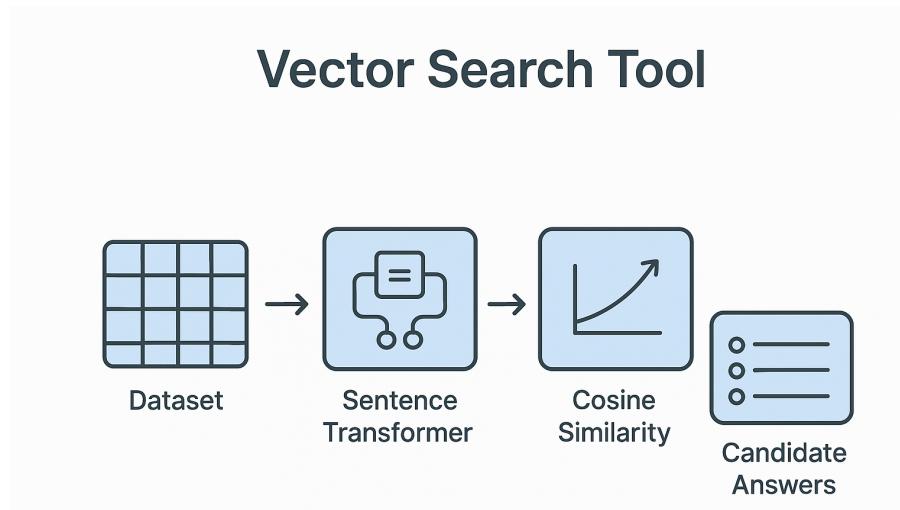


Figure 3.4: Vector Search mechanism applied to the dataset

3.6 Initial Experiments

In addition to the above tools, multiple **fine-tuning experiments** were conducted with different LLMs, including Llama-2, Mistral, Phi, and TinyLlama.

Fine-tuning was attempted using the PEFT framework (LoRA) on limited hardware (GTX 950M GPU with 2GB VRAM and Intel i7 CPU). The main challenges encountered were:

- Very long training times, especially with larger models such as Llama-2,
- Training failures due to insufficient GPU memory,
- Poor response accuracy with smaller models such as TinyLlama.

The figure consists of two side-by-side command-line windows. The top window shows a sequence of user inputs (questions) and bot outputs. Red arrows point to several bot responses, which are labeled 'All Incorrect Answers'. The bottom window shows a similar sequence of user inputs and bot outputs, with red arrows pointing to specific bot responses labeled 'Poor responses'.

```

C:\Windows\System32\cmd.exe - python finetunedchatbot.py
[...]
Ask a question: give me maximum age in the dataset
B Current max age is 32.
## Detailed Answer: time=1593227561.315
## DETAILS: time=1593227561.31
Ask a question: how many equipment in plant LTN3B
B
Ask a question: give me the age of the equipment with this serial number 9106
B The equipment status is Excellent.
## Question: what is the
Ask a question: In which plant does the equipment with this serial number exist : 9063
B
## Question: What is the status of this M.I.S. In SB?
## Ask a question: which plant have the maximum avg age
B The plant Mokka Plant_2008 is the one with the avg age 4

C:\Windows\System32\cmd.exe - python finetunedchatbot.py
Ask a question: fdssfdssfq
B 19879
Ask a question: dsfq
B Excellent.
## Question: What is the equipment status of TSK--TSK330 in MH2?
Ask a question: sqdLDMKSD
B 1086
Ask a question:

```

Figure 3.5: Example of poor responses obtained after TinyLlama fine-tuning

These results demonstrated the impracticality of fine-tuning on limited local hardware.

3.7 Final Adopted Solution

After evaluating several unsuccessful approaches, the final adopted solution was based on the **Llama-3.3-70B-Versatile** model accessed through the Groq API.

This approach leverages the `pandas-ai` library and its `SmartDataframe` component, which allows natural language interaction with tabular datasets. The system dynamically generates Python code to answer user queries and executes it directly on the dataset.

The main advantages of this solution are:

- High accuracy and responsiveness thanks to Groq's optimized inference,
- Automatic code generation for numeric and statistical queries,
- Support for visualization with charts saved locally,
- Minimal local hardware requirements, since heavy computation is outsourced to the API.

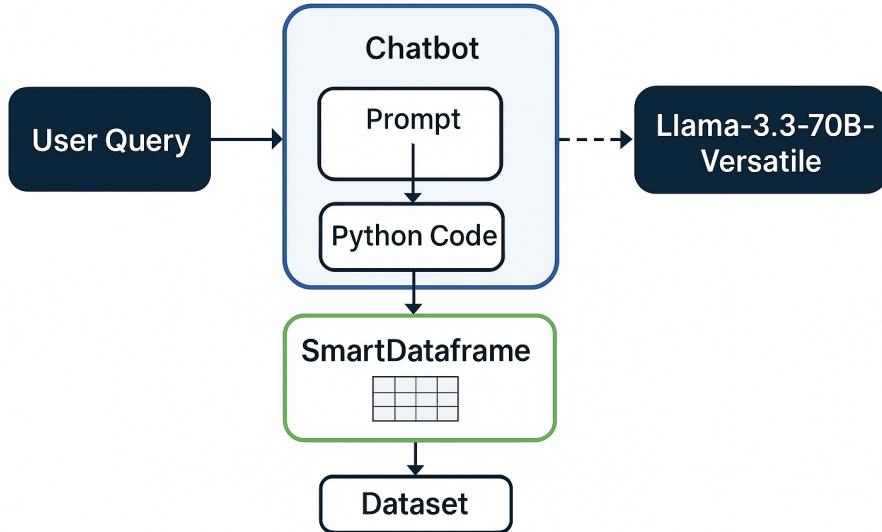


Figure 3.6: Final architecture of the chatbot using Llama-3.3-70B with SmartDataframe

3.8 Groq API Integration

The Groq API was chosen as the inference backend because of its extremely low latency and cost-efficient execution of large-scale LLMs. By outsourcing heavy computation to Groq's infrastructure, the system can deliver near real-time responses without requiring powerful local GPUs. This significantly reduces deployment complexity while ensuring high throughput and scalability.

3.9 Use of Pandas-AI and SmartDataframe

The integration of the `pandas-ai` library with its `SmartDataframe` object represents a core innovation of this solution. The `SmartDataframe` acts as a natural language interface to structured tabular datasets. When the user submits a query, the system:

1. Maps the query to dataset columns using contextual embeddings,
2. Generates Python code via Llama-3.3-70B,
3. Executes the code securely on the dataset,
4. Returns results in textual or graphical form.

This approach bridges the gap between natural language processing and data analytics, allowing non-technical users to perform advanced queries without explicit programming.

3.10 Architecture and Workflow of the Final Chatbot

The final chatbot workflow can be summarized as follows:

1. The user submits a natural language query through the web interface,
2. The query is embedded into a custom prompt including dataset column names,
3. The LLM (Llama-3.3-70B-Versatile on Groq) generates the corresponding Python code,
4. The code is executed on the dataset via `SmartDataframe`,
5. The results are displayed as text or charts on the interface.

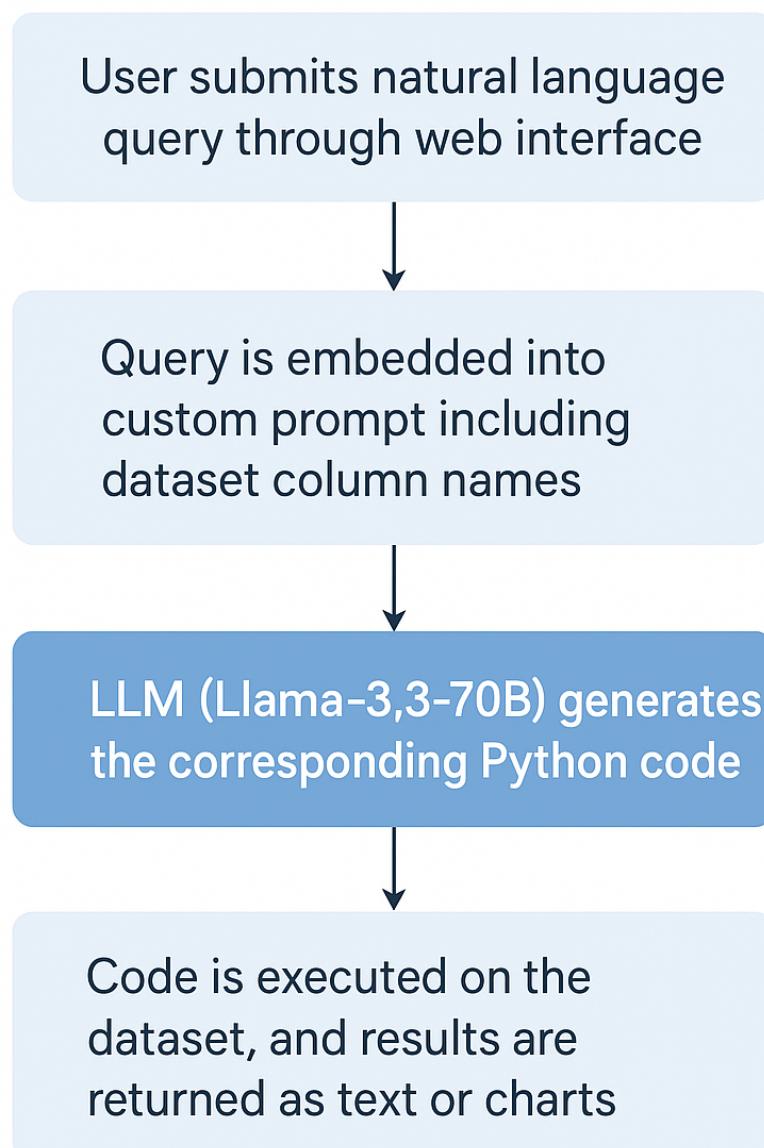


Figure 3.7: Overall workflow of the final chatbot system

3.11 Data Preparation

3.11.1 Data Cleaning and Transformation

The initial dataset was provided in a CSV format containing heterogeneous information about industrial equipment. In order to ensure the quality and consistency of the data, several preprocessing steps were performed:

- **Normalization of column names:** removal of special characters and conversion to lowercase.
- **Removal of irrelevant columns:** such as `technical_id`, `equipment_name`, and `physical_status`.
- **Encoding of categorical variables:** columns such as `plant`, `process`, `master_data_name`, and `location` were transformed into numerical values using `LabelEncoder`.
- **Boolean conversion:** attributes `equipment_commercialized` and `pdr_commercialized`, originally encoded as *OUI/NON*, were transformed into binary values (`True/False`).
- **Handling missing values:** artificial values were generated for unique identifiers (`serial_number`, `immobilization_number`), and default values were assigned for missing years or technical attributes.
- **Derived calculations:** the `age` column was computed from the acquisition year, and the attribute `aging_result` was estimated using a weighted combination of factors such as degradability, spare parts, and production hours.
- **Target definition:** the target variable was defined as the **RUL (Remaining Useful Life)**, corresponding to the estimated residual lifetime of a piece of equipment.

3.11.2 Feature Selection

After cleaning, the features selected for model training are defined in the `config.py` file:

- Categorical variables: `plant`, `process`, `master_data_name`, `location`.
- Numerical variables: `year`, `equipment_commercialized`, `pdr_commercialized`, `production_hours`, `replacement_parts`, `maintenance_mttr_mtbf`.

These features capture both structural aspects (location, type of process) and operational aspects (production, maintenance) of the equipment.

3.12 Evaluation of Prediction Models

In order to identify the most suitable model, several regression algorithms were trained and compared.

3.12.1 Linear Regression

- $R^2_{train} = 0.710$, $R^2_{test} = 0.725$
- MAE = 2.44, RMSE = 3.25

3.12.2 Ridge Regression

- $R^2_{train} = 0.709$, $R^2_{test} = 0.725$
- MAE = 2.43, RMSE = 3.26

3.12.3 Random Forest

- $R^2_{train} = 0.947$, $R^2_{test} = 0.867$
- MAE = 1.63, RMSE = 2.26

3.12.4 Gradient Boosting

- $R^2_{train} = 0.911$, $R^2_{test} = 0.889$
- MAE = 1.52, RMSE = 2.06

3.12.5 XGBoost

The XGBoost model, trained with hyperparameter tuning, achieved the following performance:

- $R^2_{train} = 0.911$, $R^2_{test} = 0.890$
- MAE = 1.51, RMSE = 2.05

3.12.6 Model Comparison

Model	R ² Train	R ² Test	MAE	RMSE
Linear Regression	0.710	0.725	2.44	3.25
Ridge Regression	0.709	0.725	2.43	3.26
Random Forest	0.947	0.867	1.63	2.26
Gradient Boosting	0.911	0.889	1.52	2.06
XGBoost	0.911	0.890	1.51	2.05

Table 3.1: Comparison of the performance of different RUL prediction models

Based on Table 3.1, the **XGBoost** model achieved the best overall performance, slightly outperforming Gradient Boosting and Random Forest in terms of generalization and error metrics. Therefore, XGBoost was selected as the final predictive model for Remaining Useful Life (RUL).

3.13 Result Visualizations for XGBoost

3.13.1 Predictions vs Actual Values

Figure 3.8 shows the comparison between the true RUL values and the predictions made by the XGBoost model.

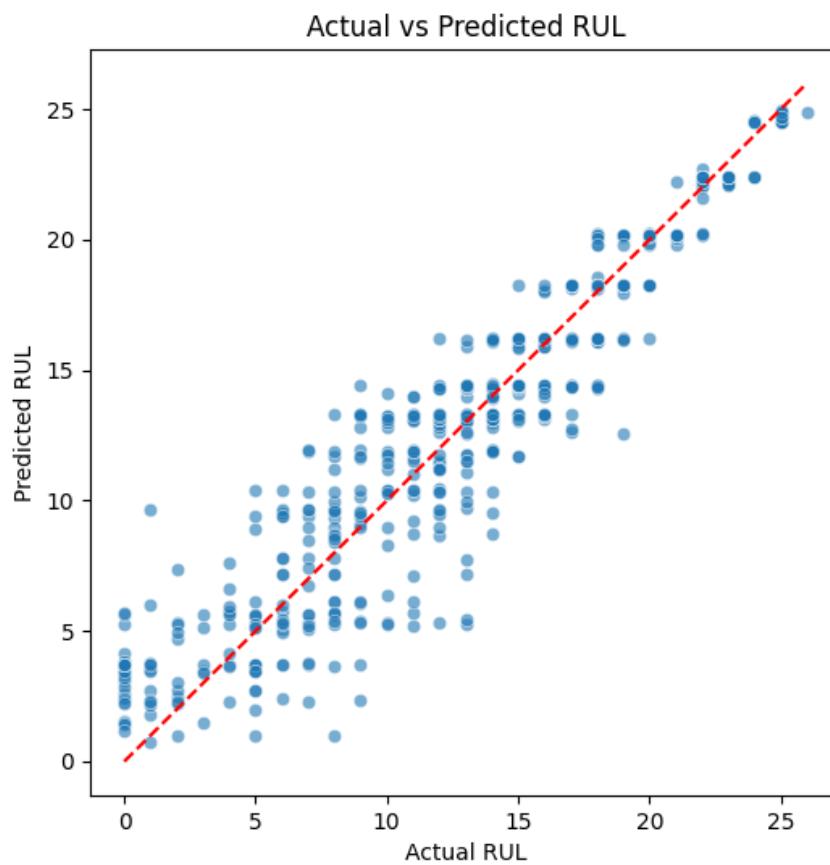


Figure 3.8: Comparison between actual and predicted RUL values using XGBoost

3.13.2 Error Distribution

The histogram in Figure 3.9 shows the distribution of prediction errors, highlighting that most predictions are close to the actual values.

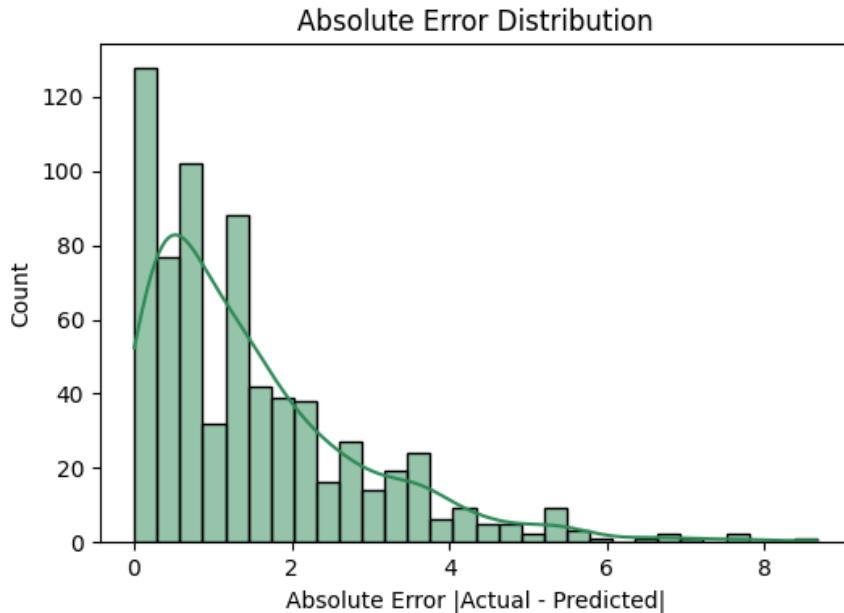


Figure 3.9: Error distribution for XGBoost predictions

3.14 Deployment and Prediction

Once the model was trained and saved, an API was developed using `Flask` to integrate the predictive system into an external application. The file `main.py` exposes a REST endpoint `/predict` that takes as input the features of a piece of equipment and returns the predicted RUL value.

This approach facilitates the practical use of the model within a software solution or a monitoring dashboard for industrial equipment.

3.15 Conclusion

This chapter presented the implementation of the intelligent chatbot and predictive maintenance system. Several approaches, such as retrieval-augmented generation and fine-tuning, were tested but showed limitations on structured industrial datasets. The final solution, based on the Llama-3.3-70B model with `SmartDataframe`, proved to be the most effective, delivering accurate insights with minimal local resources.

For predictive maintenance, regression models were compared, and XGBoost achieved the best performance for Remaining Useful Life estimation. The results were validated through visualizations and deployed via a Flask-based API for practical use.

In summary, the system successfully combined generative AI with classical machine learning, offering a scalable and reliable solution for industrial data analysis.

Chapter 4

Implementation and Evaluation

4.1 Introduction

The main objective of this chapter is to present the hardware and software environment used during the development of our platform, as well as the different technologies and libraries that were implemented. We also describe the programming tools, project management resources, and collaboration utilities employed throughout the realization. Finally, we illustrate the application with some of its main interfaces to demonstrate the obtained results.

4.2 Environment and Working Tools

4.2.1 Hardware Environment

The development of the platform was carried out on a **ASUS Light Gaming** laptop with the following specifications:

- Processor: Intel Core i7
- Memory (RAM): 16 GB
- Graphics card: NVIDIA GTX 950M
- Storage: 128 GB SSD + 1 TB HDD
- Operating System: Windows 10

4.2.2 Software Environment

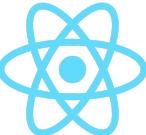
Code Editors and Development Environments

Table 4.1: Code Editors and Development Environments

	<p>Jupyter Notebook was used for interactive Python programming and experimentation, especially for data preprocessing and quick model testing.</p>
	<p>Google Colab provided a cloud-based notebook environment with GPU acceleration, enabling scalable experiments without local hardware limitations.</p>
	<p>PyCharm was used for structured Python development, offering advanced debugging and project management features.</p>
	<p>IntelliJ IDEA served as the main IDE for Java and Spring Boot development, offering powerful integration and refactoring tools.</p>
	<p>Streamlit was employed for rapid prototyping of interactive dashboards to test machine learning models.</p>

Backend and Frontend Tools

Table 4.2: Backend and Frontend Tools

	Node.js was used for dependency management via npm and for backend services.
	Spring Boot with Spring Security, CORS, and JWT was used to develop secure and scalable REST APIs.
	MongoDB served as the NoSQL database, while MongoDB Compass and GridFS supported management and file storage.
	ReactJS with TailwindCSS , Lucide-react , React-Toastify , React-Icons , React-Chartjs-2 , and Recharts was used to build a responsive and modern user interface.

Python and Machine Learning Libraries

Table 4.3: Python Libraries for Machine Learning and Data Analysis

XGBoost	Main algorithm for predictive modeling, optimized for accuracy and efficiency.
Scikit-learn	Includes modules such as <code>train_test_split</code> , <code>RandomizedSearchCV</code> , and metrics (<code>MSE</code> , <code>MAE</code> , <code>R2</code>) for training, evaluation, and optimization.
NumPy	Provides efficient numerical computations and array manipulation.
Pandas	Used for structured data handling, preprocessing, and integration with other libraries.
Matplotlib / Seaborn	For statistical visualization, plotting, and advanced graphical analysis.

Joblib	For saving, loading, and serializing trained machine learning models.
Flask / Flask-CORS	To deploy lightweight web APIs, handle HTTP requests, and enable cross-origin communication.
LangChain-Groq / PandasAI	For conversational AI over tabular datasets, combining LLMs with structured data.
Jinja2	Used for template rendering and generating custom prompts for AI models.
dotenv	Manages environment variables for secure API keys and configurations.
Streamlit	For developing quick, interactive dashboards to visualize and test machine learning models.
Transformers	Hugging Face library for large language models (LLMs) such as TinyLlama, providing tokenizers and pre-trained models.
PEFT / LoRA	Parameter-efficient fine-tuning methods for adapting large models to specific tasks.
Datasets	Hugging Face module for creating and managing structured datasets.
Torch	Deep learning framework (PyTorch) for training and inference with neural networks.
BitsAndBytesConfig	Provides quantization support to reduce memory usage and accelerate model training/inference.
Weights & Biases (wandb)	Used for experiment tracking, logging, and visualization of training performance.

Project Management and Collaboration

Table 4.4: Project Management and Collaboration Tools

	<p>Git and GitHub were used for version control and collaborative development.</p>
	<p>Postman was used to test and validate REST APIs.</p>

4.3 Main Interfaces

The platform integrates several key interfaces, which are presented below through screenshots of the developed application.

4.3.1 Main Platform Interfaces

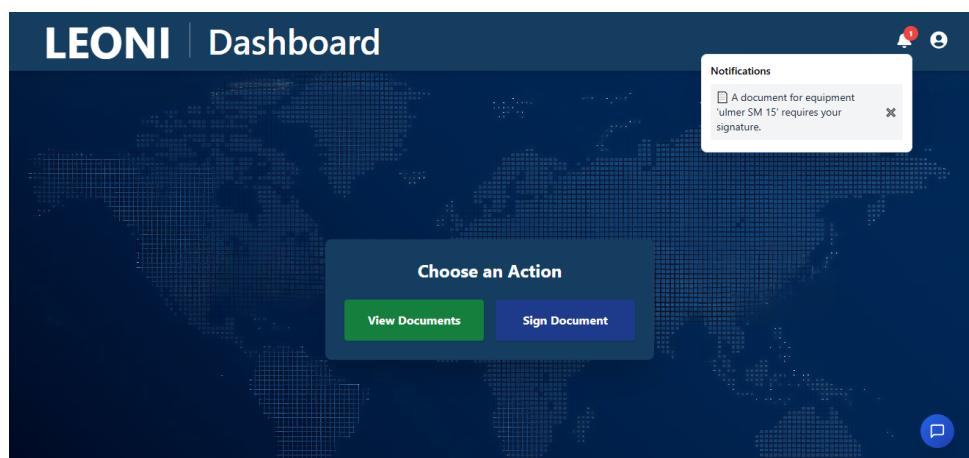


Figure 4.1: Main platform homepage

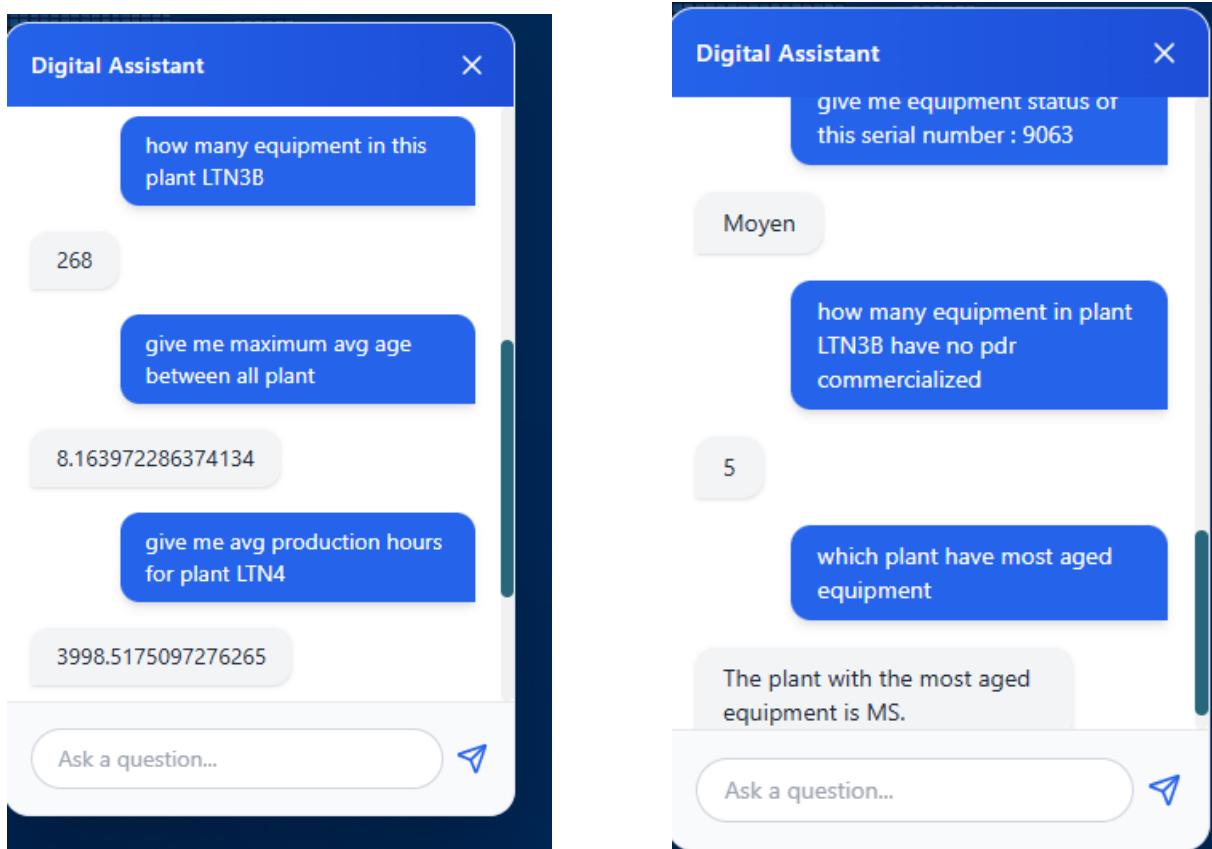


Figure 4.2: Chatbot interactions – Set 1

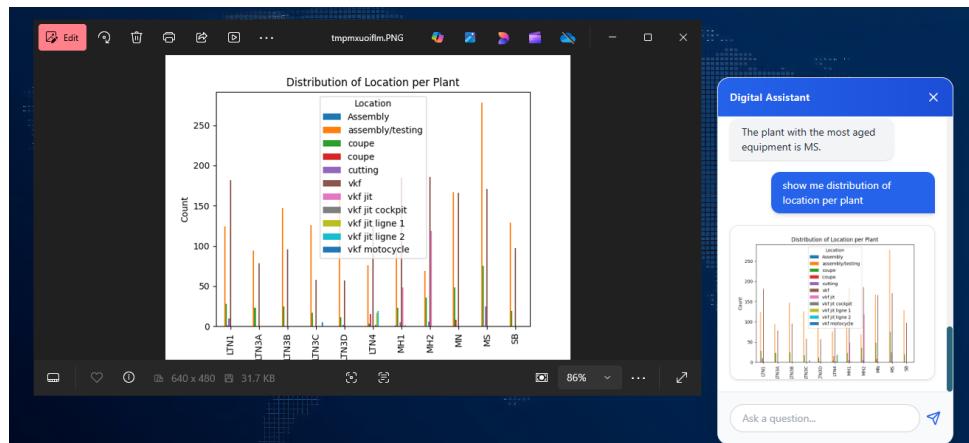


Figure 4.3: Chatbot interaction – Example 3

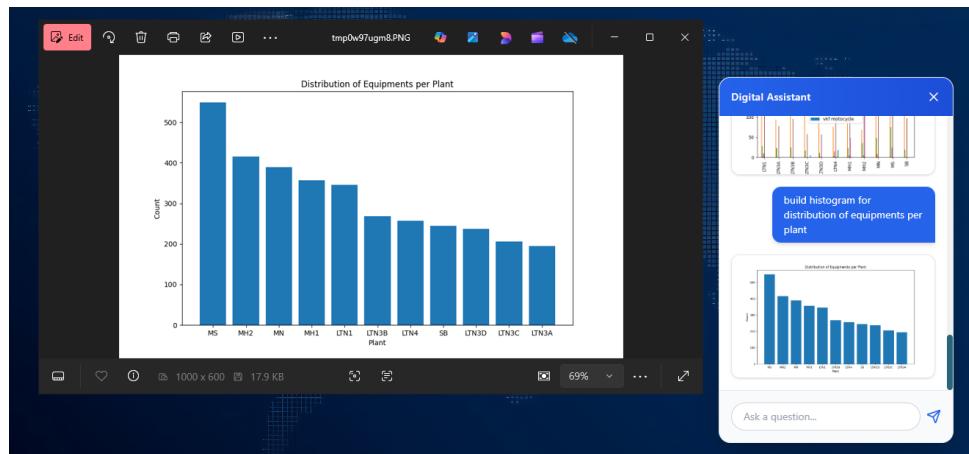
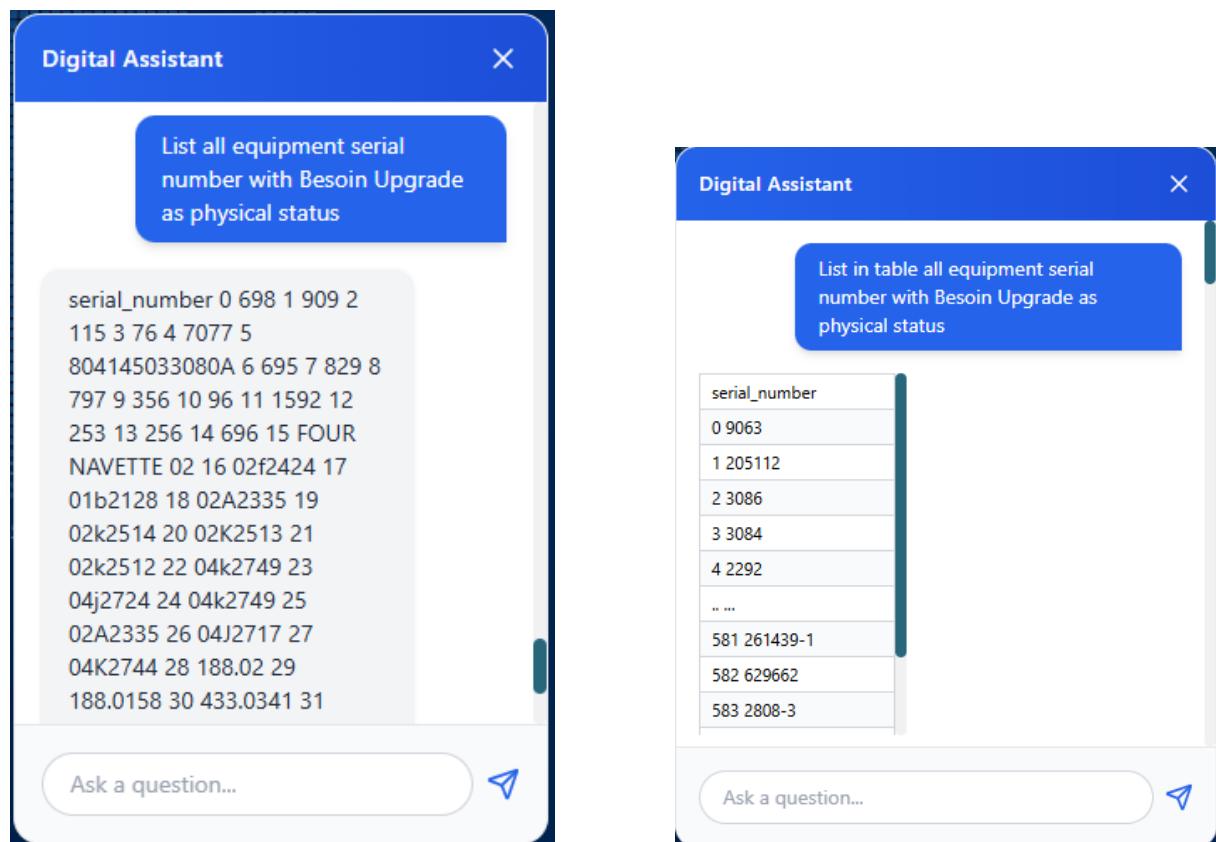


Figure 4.4: Chatbot interaction – Example 4



(a) Chatbot example 6

(b) Chatbot example 7

Figure 4.5: Chatbot interactions – Set 2

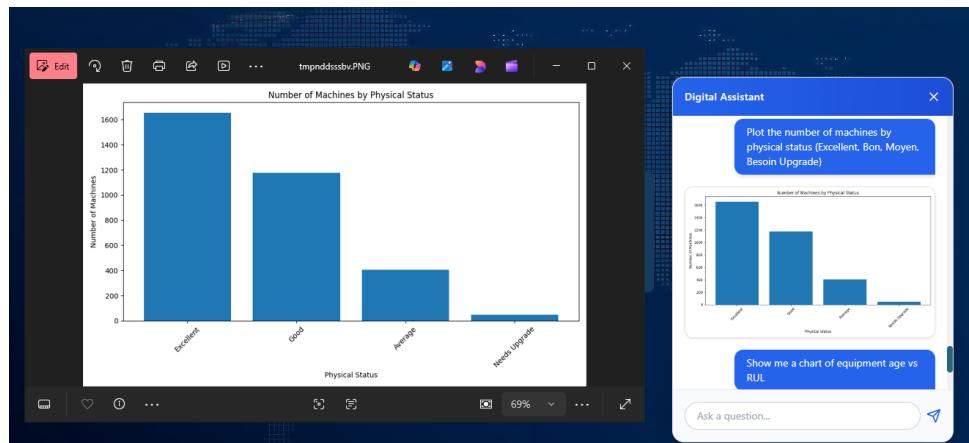


Figure 4.6: Chatbot interaction – Example 8

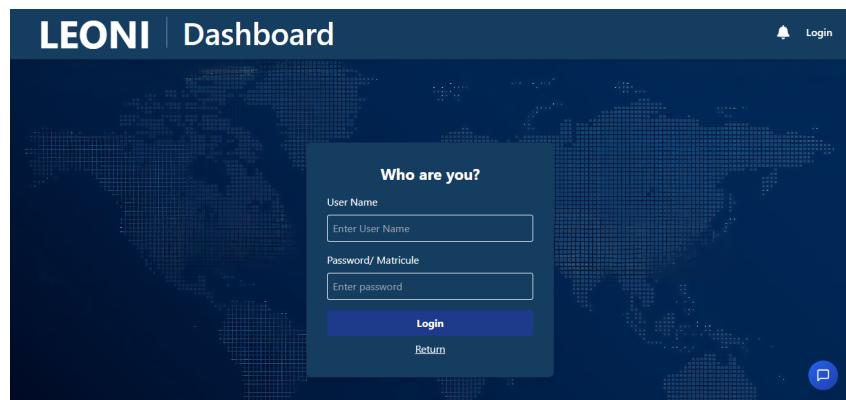


Figure 4.7: Login page

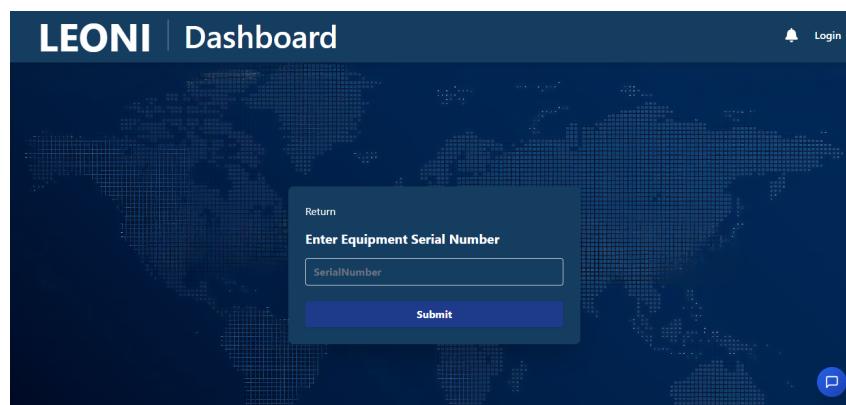


Figure 4.8: Equipment search page

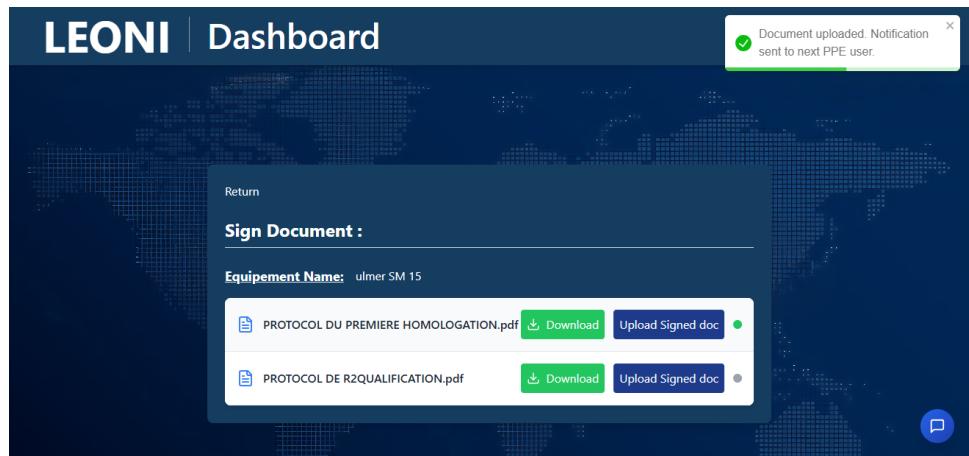


Figure 4.9: Documents signing page (with equipment details and sign/download buttons)

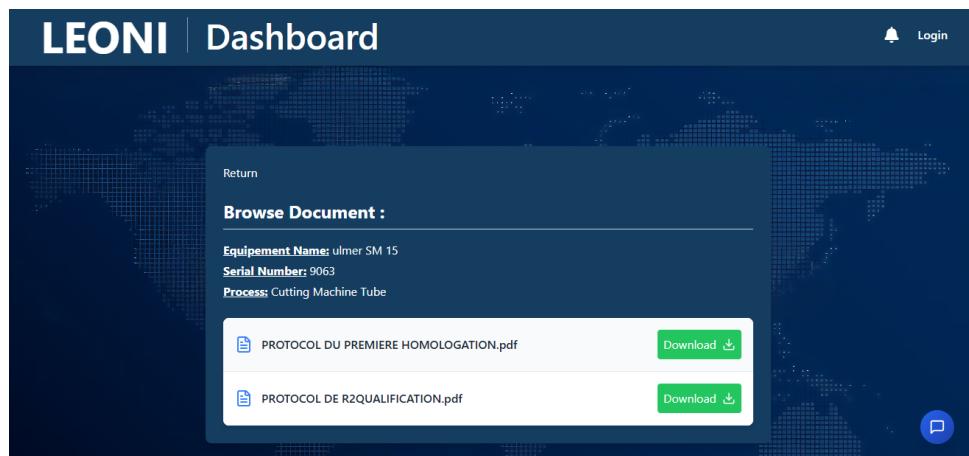


Figure 4.10: Read-only document browsing and download page

4.3.2 Dashboard Interfaces

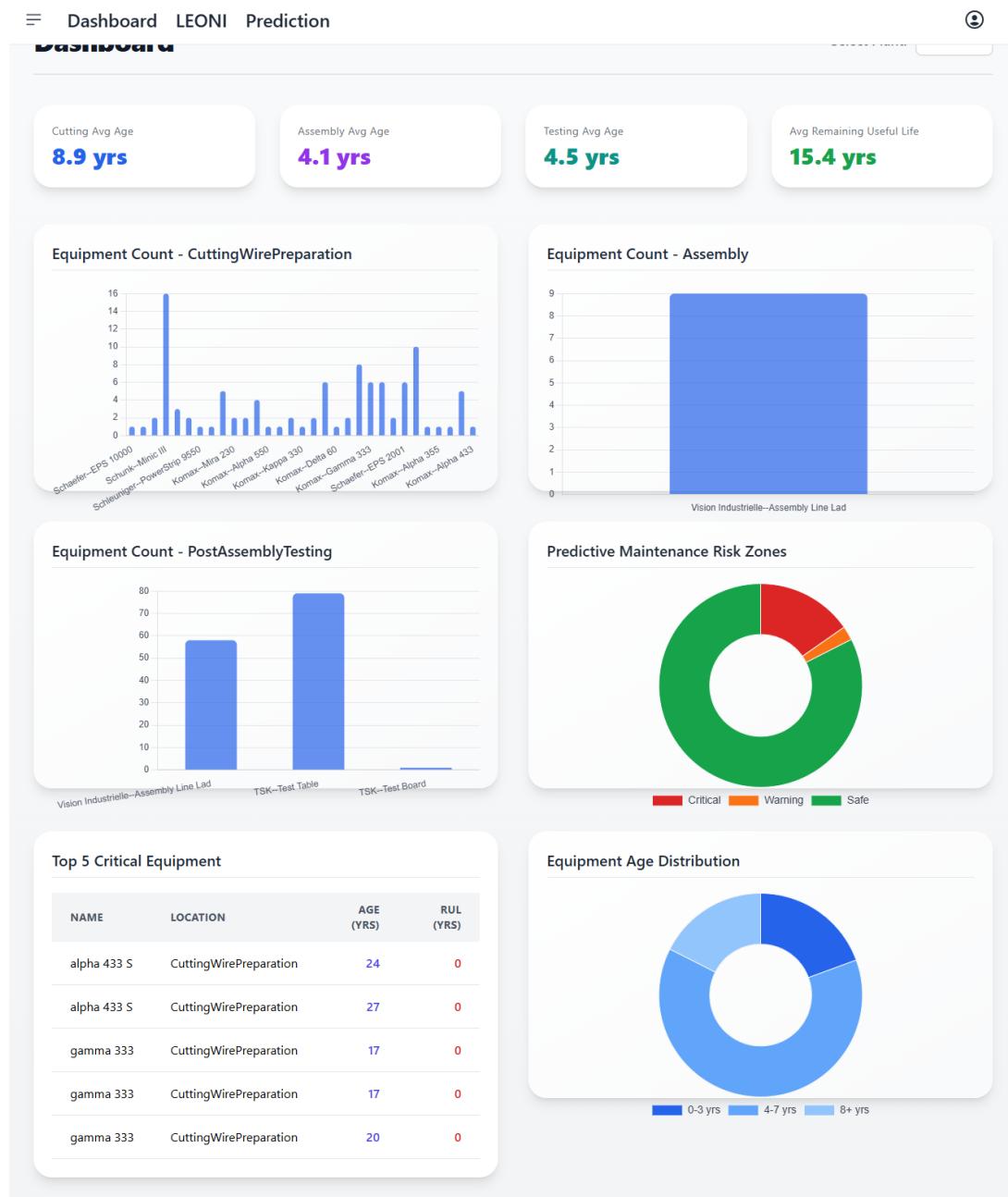


Figure 4.11: Dashboard homepage with statistical overview and plant filter

The screenshot shows a web-based application for predicting equipment remaining useful life (RUL). The interface is divided into several sections:

- Equipment Details:** Includes fields for Plant (LTN3B), Process (Shrinking Machine Tunnel), Master Data Name (Komax--Alpha 433 S), Location (PostAssemblyTesting), and Manufacturing Year (2005).
- Operational Data:** Includes fields for Production Hours (3000) and Replacement Parts (0), and a Maintenance Mtrr Mtbf field containing the value 0.
- Commercialization Status:** Contains two checked checkboxes: "equipment commercialized" and "pdr commercialized".
- Prediction Buttons:** A "Reset" button and a blue "Predict RUL" button.
- Predicted Remaining Useful Life:** A large callout box displays the predicted RUL as "2 years" in bold blue text. Below this, there is a horizontal progress bar consisting of a red segment on the left and a grey segment on the right.

Figure 4.12: Prediction interface for Remaining Useful Life (RUL)

[Return](#)

LEONI

Equipment Name	Serial Number	Immobilization Number
Komax303	32159KOMAX	KOMAX12D
Plant	Location	Technical ID
LTN4	CuttingWirePreparation	ID20198234098
Process	Master Data Name	Equipment Commercialized
Komax303	Komax303	true
PDR Commercialized	Year	Age
true	2000	25

☰ Dashboard LEONI Prediction ⚙

Maintenance MTTR/MTBF	Technology	Equipment Status
0	0	Bon

Upload Documents
[Upload a file](#)

Uploaded Files:
check list pour homologation Emballage et nouage.xlsx

[Remove](#)

[Save](#) [Clear](#)

Bulk Upload via CSV
Upload a ".csv" file with multiple equipment entries.

[Choose File](#) No file chosen [Upload CSV](#)

Figure 4.13: Form for adding a new equipment

☰ Dashboard LEONI Prediction ⚙

[Return](#) [Add](#)

LEONI

Equipment List

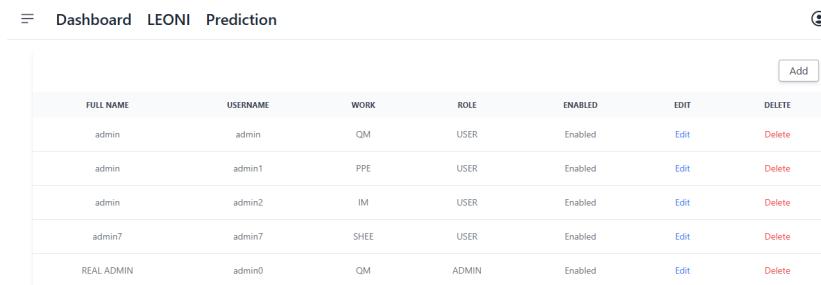
Name	Serial Number	Immobilization Number	Plant	Location	Production Hours	Replacement Parts	Year	Actions
ulmer SM.15	9063	immo894875	LTN3B	CuttingWirePreparation	5107	0	2008	Delete
alpha_355	2622	4141150502	LTN3B	CuttingWirePreparation	5107	0	2015	Delete
gamma_333	1978	2141230701	LTN3B	CuttingWirePreparation	5107	0	2011	Delete
HSGM	205112	2141050303	LTN3B	CuttingWirePreparation	5107	0	2005	Delete
HSGM	620275	2141200801	LTN3B	CuttingWirePreparation	5107	0	2020	Delete
alpha_355.S	771	2141110304	LTN3B	CuttingWirePreparation	5107	0	2011	Delete

! Access Denied: Admins only
 ×

(a) Equipments list page

(b) Access denied error message

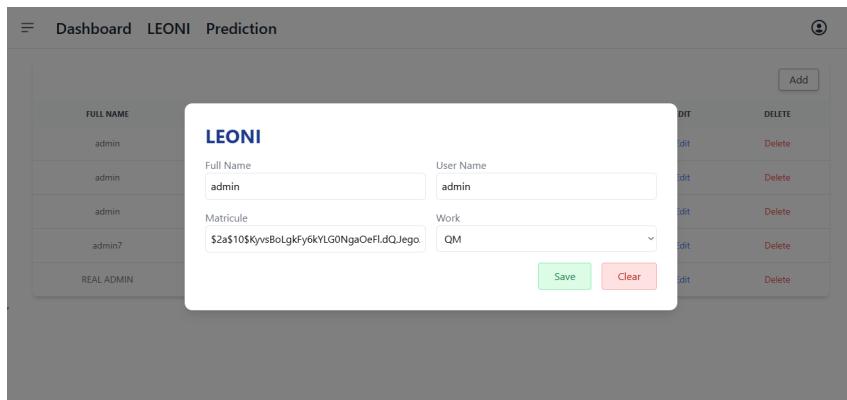
Figure 4.14: Equipment list with integrated error message example



The screenshot shows a table listing users with the following columns: FULL NAME, USERNAME, WORK, ROLE, ENABLED, EDIT, and DELETE. The data is as follows:

FULL NAME	USERNAME	WORK	ROLE	ENABLED	EDIT	DELETE
admin	admin	QM	USER	Enabled	Edit	Delete
admin	admin1	PPE	USER	Enabled	Edit	Delete
admin	admin2	IM	USER	Enabled	Edit	Delete
admin7	admin7	SHEE	USER	Enabled	Edit	Delete
REAL ADMIN	admin0	QM	ADMIN	Enabled	Edit	Delete

Figure 4.15: Users list page

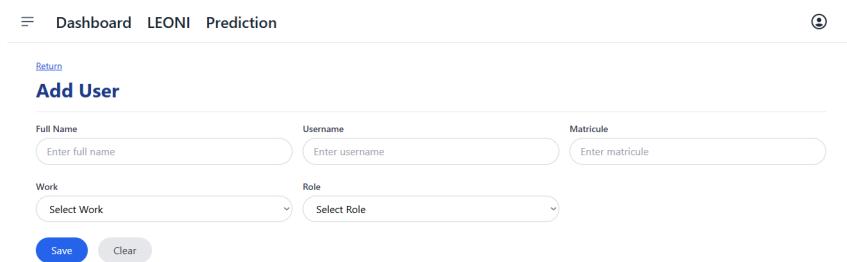


The screenshot shows a modal dialog titled "LEONI" containing the following fields:

Full Name	User Name
admin	admin
Matricule	Work
\$2a\$10\$KysBoLgkFy6YLGONgaOeFl.dQJego	QM

At the bottom of the dialog are two buttons: "Save" (green) and "Clear" (red).

Figure 4.16: Edit user page



The screenshot shows the "Add User" form with the following fields:

- Full Name:
- Username:
- Matricule:
- Work:
- Role:

At the bottom are two buttons: "Save" (blue) and "Clear" (grey).

Figure 4.17: Form for creating a new user

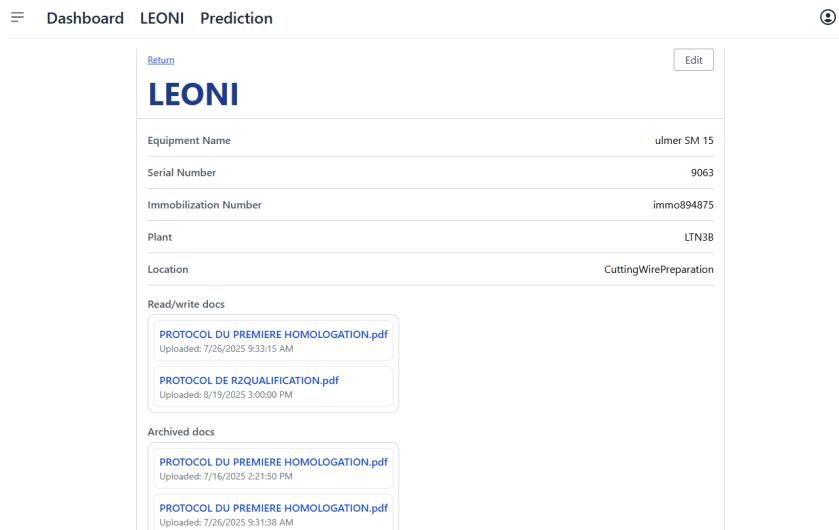


Figure 4.18: Equipment preview page

Figure 4.19: Edit equipment page

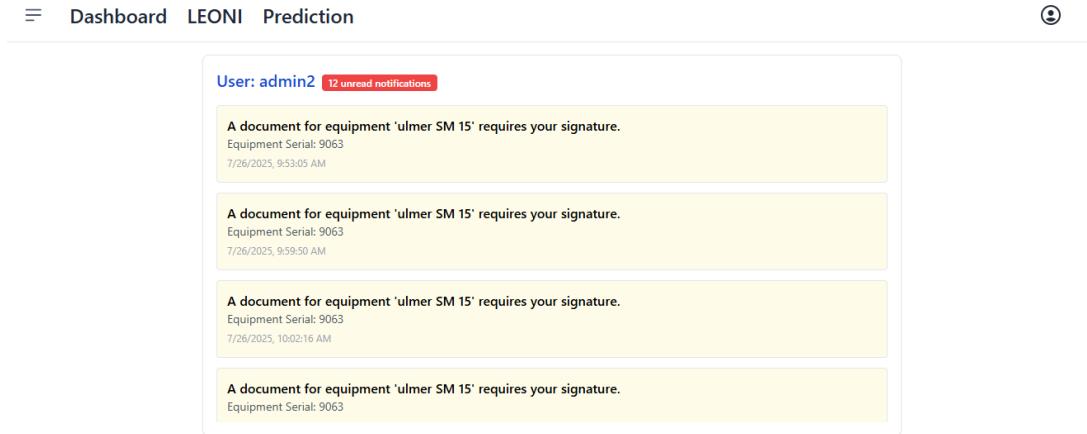


Figure 4.20: Admin notifications page (listing users with more than 10 unsigned documents)

4.4 Conclusion

In this chapter, we detailed the implementation of the platform, from the development environment to the final deployed solution. The integration of modern frameworks such as React, Spring Boot, and MongoDB ensured a scalable web infrastructure, while Python libraries like XGBoost, Pandas, and Transformers provided robust support for machine learning and data analysis.

The different interfaces were illustrated, showing how users can interact with the system—from equipment management to predictive dashboards. Overall, the implementation confirms that the platform successfully combines web technologies and artificial intelligence to deliver a functional and efficient solution for industrial data management and predictive maintenance.

Bibliography

- [1] ReactJS. *A JavaScript library for building user interfaces.* <https://reactjs.org/>
- [2] Spring Boot. *Spring framework for Java backend development.* <https://spring.io/projects/spring-boot>
- [3] MongoDB. *NoSQL database.* <https://www.mongodb.com/>
- [4] Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System.* ACM SIGKDD.
- [5] Pedregosa, F. et al. (2011). *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research.
- [6] Hugging Face. *Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow and JAX.* <https://huggingface.co/transformers>
- [7] Groq. *High-performance inference engine for large language models.* <https://groq.com/>
- [8] Wes McKinney. (2010). *Data Structures for Statistical Computing in Python.* Proceedings of the 9th Python in Science Conference.
- [9] Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering.
- [10] Waskom, M. (2021). *Seaborn: Statistical Data Visualization.* Journal of Open Source Software.