Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et Informatique Industrielle

Réf. :

Présenté et soutenu par :
Ahmed Amin Gasmi _ Hani Yagoub

Le : avril 2025

Smart App for bulk cement loading, Using Python, MySQL and TIA portal

Jury :

M.	Athmane Zitouni	Pr	Universite de Biskra	Président
M.	Khaled Abada	MAA	Universite de Biskra	Examinateur
M.	Fateh Benchabene	Pr	Universite de Biskra	Encadreur

Année universitaire : 2024/2025

Université Mohamed Khider de Biskra
Faculté des Sciences et de la Technologie
Département de génie électrique

MÉMOIRE DE MASTER

Sciences et Technologies
Automatique
Automatique et Informatique Industrielle

Réf. : Entrez la référence du document

Smart App for bulk cement loading, Using Python, MySQL and TIA portal

Le : ………………………………………………….

Présenté par :

- Hani Yagoub
- Ahmed Amin Gasmi

Avis favorable de l'encadreur :

- M. Fateh Benchabene

Signature Avis favorable du Président du Jury

Cachet et signature

SUMMARY

This project was carried out at the Biskria Cement Plant with the aim of addressing the issue of mismatching the type of bulk cement loaded into trucks with customer orders. To tackle this problem, we developed a smart application that primarily relied on the Python programming language, along with the use of several auxiliary programming tools. A Programmable Logic Controller (PLC) from the S7-1200 series was employed using the TIA Portal development environment .Additionally, an artificial intelligence technique was implemented to recognize truck license plate numbers using Python libraries. Database creation and management were handled using the MySQL tool. The smart application will be able to verify product and customer order database accuracy and grant the necessary permissions to initiate the loading process. The newly integrated system was tested in real-world conditions and proved its effectiveness by delivering accurate and satisfactory results. This project allowed us to successfully achieve our objectives, particularly in improving the reliability and precision of the cement loading process. It also contributed to enhancing our technical competencies and expanding our understanding of the integration between industrial automation programming and artificial intelligence technologies.

نُفِّذ هذا المشروع في مصنع بسكريا للإسمنت بهدف معالجة مشكلة عدم توافق نوع الأسمنت السائب المُحمَّل في الشاحنات مع طلبات العملاء. ولمعالجة هذه المشكلة، طوَّرنا تطبيقًا ذكيًا يعتمد بشكل أساسي على لغة برمجة بايثون، إلى جانب استخدام العديد من أدوات البرمجة المساعدة.

استُخدمت وحدة تحكم منطقية قابلة للبرمجة (PLC) من سلسلة S7-1200 باستخدام بيئة تطوير بوابة TIA. بالإضافة إلى ذلك، طُبِّقت تقنية ذكاء اصطناعي للتعرف على أرقام لوحات ترخيص الشاحنات باستخدام مكتبات بايثون. وتم إنشاء وإدارة قواعد البيانات باستخدام أداة MySQL.

سيتمكن التطبيق الذكي من التحقق من دقة قاعدة بيانات المنتج وطلبات العملاء، ومنح الأذونات اللازمة لبدء عملية التحميل.

تم اختبار النظام المتكامل حديثًا في ظروف واقعية، وأثبت فعاليته من خلال تقديم نتائج دقيقة ومرضية. وقد مكّننا هذا المشروع من تحقيق أهدافنا بنجاح، لا سيما في تحسين موثوقية ودقة عملية تحميل الأسمنت. كما ساهمت في تعزيز كفاءاتنا التقنية وتوسيع فهمنا للتكامل بين برمجة الأتمتة الصناعية وتقنيات الذكاء الاصطناعي.

Dedication

Man existed on the face of the earth, and did not live in isolation from the rest of humanity. At all stages of life, there are people who deserve our thanks, and the people most deserving of thanks are the parents. Because of their virtue that reaches the sky; Their presence is a reason for salvation and success in this world and the hereafter. Thanks also to my brothers and friends, whom I bear witness to be the best companions in all matters. Thank you

Acknowledgment

Praise be to God always and forever. Praise be to God, who inspired us with patience, success, and payment to complete this simple and humble work. I would like to thank Mr. Fateh Benchabene for agreeing to be our supervisor during this work, for the trust he placed in us, and for his valuable advice. We extend our sincere thanks to the workers of the Biskria Cement Factory, who gave us the space and the opportunity to build this project. The project, We extend a very special thanks to the engineer, Mr. Ibrahim Yaqoub, for his trust, guidance, and tireless efforts. Despite his intense work, he contributed a lot, and without him, we would not have reached where we are. We are grateful to him and we also thank his staff, who also helped us during our training work. In conclusion, I extend my sincere thanks to my entire family, my parents, my brothers and all my close friends for accompanying, saying goodbye and encouraging them all the time.

TABLE OF FIGURES

Table of Abbreviations

AC	Alternating Current
AESOP	Adaptive Electronic System for Optimized Performance
APC	Advanced process control
AI	Analog input
AO	Analog output
BC	Biskria Cement
CEO	Chief executive officer
CETIM	Center for Studies and Technological Services of the Building Materials Industry
CPU	Central processing unit
CEM	CEMENT
CV	Computer Vision
DCS	Distributed Control System
DB	Data Block
DI	Digital input
DO	Digital output
DP	Decentralized Periphery
DC	Direct Current
DIV	Division
ERP	enterprise resource planning
ESD	Emergency Shut Down
GPS	Global Positioning System
GUI	Graphical User Interface
HMI	human machine interfaces

HTTP	HyperText Transfer Protocol
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IMC	Integrated Marketing Communications
INT	Integer
IO	Input/Output
IP	Internet Protocol
ISA	International Society of Automation
ISBN	International Standard Book Number
ISO	International Organization for Standardization
ISSN	International Standard Serial Number
MES	manufacturing execution systems
MID	Measuring Instruments Directive
MPI	Multipoint interface
NLTK	Natural Language Toolkit
OB	Organization Block
OCR	Optical Character Recognition
OPC UA	Open Platform Communications Unified Architecture
PA	Process Automation
PCS	Process control system
PI	PROFINET International
PID	Proportional – Integral – Derivative
PLC	Programmable Logic Controller
PNO	PROFIBUS Nutzerorganisation
PTP	Point-to-point communication
RFID	Radio-frequency identification

RLY	Relay
RS	Recommended Standard
RCV	Receive
SCADA	supervisory control and data acquisition
SCL	Structured Control Language
SIS	safety instrumented systems
S P A	Société Par Actions
SQL	Structured Query Language
STRG_VAL	STRING Value
TCP	Transmission Control Protocol
TIA	Totally Integrated Automation
TV	Television
UI	User Interface
WFCS	Workshop on Factory Communication Systems

TABLE OF CONTENTS

General problem

The main problem is the possibility of errors during the bulk cement loading process, as customers may be provided with a different type of cement than what was ordered. This is due to the lack of an accurate digital system that ensures the loading product type matches customer orders. Furthermore, it is difficult to verify the type of cement inside the truck, unlike cement packaged in bags, which can be easily identified by the color of the bag or the outer label.

General Introduction

One of the pillars of economic growth is modern industry, which is always changing due to the quick development of new technologies and the broad use of intelligent and artificial intelligence-powered solutions. the cement industry, a vital component of infrastructure and building projects, is one of the industries that has profited the most from this digital revolution. [I]

This industry depends on complex, highly regulated processes that demand precise monitoring and control to ensure operational efficiency, safety, and final product quality. Consequently, digital transformation, artificial intelligence and process automation have become essential components in achieving high-performance industrial operations. [I]

Smart industrial applications and intelligent automation technologies offer significant potential for enhancing process efficiency by enabling accurate data acquisition, improving data-driven decision-making, reducing human error and accelerating operational response times. In the cement manufacturing sector, these technologies spport improvements in raw material handling, production consistency, and the coordination of logistics and distribution processes. [II]

Despite the widespread use of traditional automation systems in many cement plants, continuous optimization using AI-based solutions remains essential to meet the growing demands for productivity, quality, and sustainability.

This thesis aims to enhance the truck inspection process in bulk cement loading operations by developing a smart application using OpenCV technology. This application enables truck license plates to be read and compared to a central database to ensure compliance with loading requirements, including the type of cement required. This solution contributes to increased accuracy and speed of operational processes through its integration with the Distributed Control System (DCS), enhancing control efficiency and streamlining workflow within the plant.

Chapter 1: This chapter provides a comprehensive overview of the cement production facility (Biskria Cement), including an overview of the production line and the overall process.

Chapter 2: This chapter explains the architecture of the distributed control system, covering both hardware and software aspects. It describes the main hardware components, such as programmable logic controllers (PLCs), sensors, actuators, I/O modules, and servers. From a software perspective, it explains the role of advanced tools such as PCS7, TA Portal, SQL, and Python in achieving robust process control and data management. Chapter Three: Application Development and Integration with the Factory Environment

Chapter 3: This chapter focuses on the design and implementation of the smart application, explaining how to use OpenCV to capture and process license plate images, how to interact with the TIA Portal for real-time control, and how to use MySQL to manage the database. It also discusses how to integrate the application with the factory system to ensure automatic truck verification and improve cement loading efficiency.

CHAPTER 1

AN OVERVIEW OF THE CEMENT MANUFACTURING PROCESS AT BISKRIA CEMENT FACTORY

I.1. Introduction:

Cement is an essential pillar in the construction sector, as it is the vital component of the construction of infrastructure and infrastructure thanks to its physical and chemical characteristics that give it durability and rigidity. Here, BISKRIA CEMENT stands out as one of the leading cement production and distribution companies, with its ability to meet the market's needs with high quality products and world standards.

This chapter aims to provide a comprehensive overview of BISKRIA CEMENT, from its definition and areas of activity, to reviewing its technologies used at various stages of production. We will also highlight basic industrial processes, from raw material preparation to processing and manufacturing processes, helping to understand the production cycle.

I.2. Presentation of SPA Biskria Cement:

I.2.1. Definition of Biskria Cement:

BISKRIA CEMENT is a Société Par Actions (S.P.A), a private company under Algerian law, established in January 2009. Its primary mission is the production and commercialization of cement. The company consists of a single entity located at its headquarters. All its shareholders are private Algerian economic operators[1].

Figure I.1 : SPA BISKRIA CEMENT

I.2.2. Headquarters :

- **Address**: DJAR BELAHRACHE, BRANIS, Wilaya of Biskra 07000, ALGERIA
- **Phone**: +213 (0) 33 62 74 62
- **Fax**: +213 (0) 33 62 73 92
- **Email**: biskria.spa@gmail.com
- **Website**: biskriaciment-dz.com **[3]**

Figure I.2 : Logo SPA BISKRIA CEMENT

I.2.3. Legal Framework :

- **Company Name**: BISKRIA CEMENT
- **Legal Form**: Société Par Actions (Joint-stock company)
- **Date of Establishment**: January 2009
- **Business Activity**: Cement production and commercialization
- **Share Capital**: 4,284,000,000 DA

I.2.4. Services :

SPA BISKRIA CEMENT operates in three main sectors:

- **Administrative Sector**: Includes administrative services responsible for managing the company.
- **Industrial Sector**: Comprises three production lines with a total capacity of **5 million tons per year**.
- **Commercial Sector**: Includes sales and shipping services. [1]

I.2.5 Organigramme de l'entreprise :

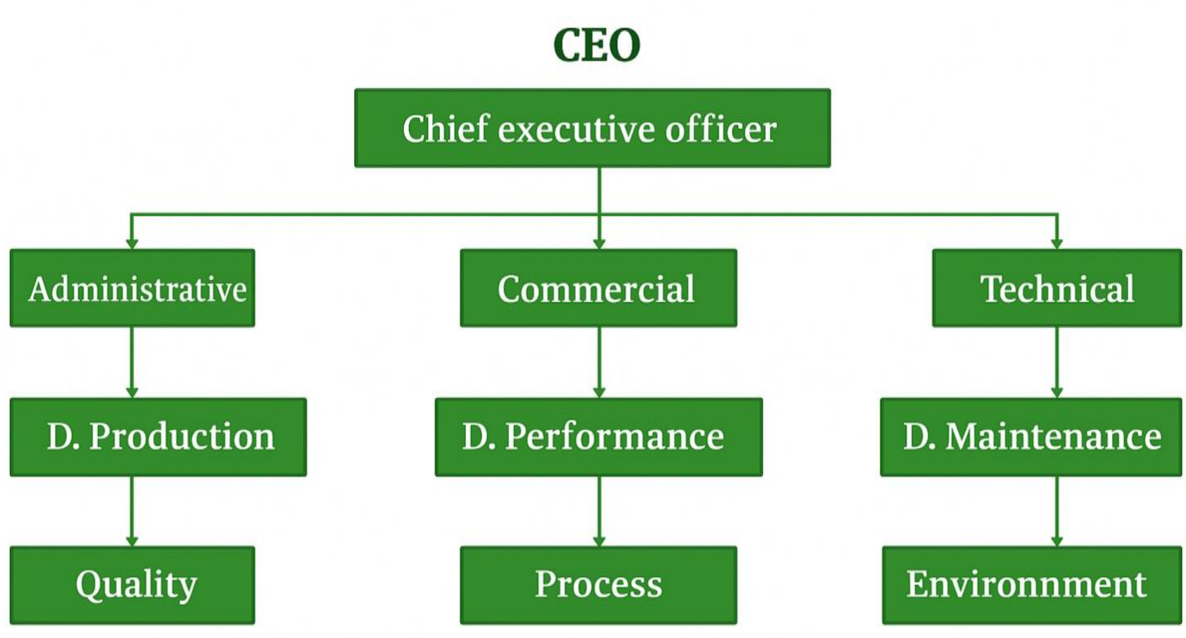


Figure I.3 : Organigramme de l'entreprise

I.2.6. Products :

SPA BISKRIA CEMENT produces high-quality cement, under a quality management system certified to **ISO 9001 standards**. The products are systematically tested by the company's laboratory and periodically evaluated by **CETIM** (Center for Studies and Technological Services of the Building Materials Industry).

The following figure presents the different types of cement produced by **BISKRIA CEMENT**:

Figure I.4: Types of cement produced by BISKRIA CEMENT.

I.3. Cement Manufacturing Process :

The raw mix is prepared from a combination of **limestone** and **clay**, adjusted and corrected with **sand** and **iron ore** [3].

The cement manufacturing process is divided into **five main stages:**

- **Extraction Zone**
- **Raw Materials Zone**
- **Clinker Firing Zone**
- **Cement Grinding Zone**
- **Packaging Zone**

I.3.1 Extraction Zone :

a. Raw Material Extraction :

Raw materials are extracted from quarries in the form of blocks of various sizes. The extraction process involves obtaining limestone from the Djebel M'hor open-pit quarry and clay from Djebel Etaref, located 15 km away.

These raw materials are extracted from rock formations using explosives, mechanical shovels, or bulldozers [4].

Once extracted, limestone and clay are transported by dump trucks to a storage area. Additionally, clay and iron ore are purchased and transported by trucks to a separate storage facility.

Figure I.5: Limestone Quarry

b. Crushing :

The crusher's role is to reduce the raw materials to a size suitable for the grinder. The raw materials are provided by the quarries in large chunks and must be crushed to the desired granulometry (0 – 25 mm) [4].

c. Raw Material Storage :

After the crushing operation, the materials are reduced to a granulometry of 0 to 25 mm. These materials are then conveyed to the primary stock (storage hall) using covered conveyor belts for pre-homogenization. The storage area includes:

- Limestone stored in a circular hall with two white and red polar structures, each with a capacity of 40,000 tons.
- Clay stored in a green polar structure with a capacity of 40,000 tons.

Figure I.6: Storage Polars

I.3.2 Raw Material Zone :

a. Pre-homogenization :

The raw material is spread in successive horizontal layers, which eventually form a pile with the desired overall composition. The layers are then taken vertically, allowing continuous extraction of material with a consistent composition. The material is retrieved from the pre-homogenization piles by bucket wheels, and pre-homogenization is achieved using a raking arm that rotates around an axis within a hall [4].

b. Dosing :

The dosing of the different components is automatically controlled at the entrance of the grinder.

The raw material and consisting of a mixture of correction raw materials in of the proportions which are defined according to the values of the raw chemical modules, in general the raw material consists of 70% limestone, 20% clay and 10% between sand and iron ore, four dosers are installed under the hoppers, limestone and the clay and iron ore and sand are respectively extracted by these measuring devices, each dispenser is automatically systematized, the raw materials extracted are loaded onto a single belt conveyor and transported to the raw mill.

Figure I.7 : Doser

c. Raw Grinding :

The grinding and drying of the raw mix is carried out at a rate of 500 tons per hour in a vertical roller mill with a central discharge, which includes one drying compartment and two grinding compartments. Drying is achieved using kiln gases at 320 – 330°C during normal conditions for a moisture content of 5.5%.

Figure I.8: Raw Grinding Mill

To promote subsequent chemical reactions, the raw materials must be dried and finely ground to 90 µm in vertical roller mills. These more recent mills are energy-efficient and allow for more effective drying [4].

d. Homogenization :

The powder obtained at the output of the grinder is called raw meal. This raw meal is sent to a storage silo known as the homogenization silo, with a unit capacity of 20,000 tons.

The most commonly used method for homogenization is the injection of air at the bottom of the silo, which helps to fluidize the raw meal [5].

Figure I.9 : Raw Meal Homogenization Silo

I.3.3. Firing Zone :

a. Preheater or Cyclones :

The plant has two 5-stage preheaters that supply the rotary kiln in parallel. During this stage, the mixture is introduced into the upper part of the preheating/precalcination tower. Inside the tower, five cyclones, funnel-shaped structures stacked one inside the other, receive the raw mix in turn. As the raw mix cascades down, it comes into contact with the gases from the kiln. The raw mix, which is further mixed, begins to gradually heat up as it approaches the kiln, thus avoiding thermal shock from the drastic temperature change (from 20°C to 1400°C). At this point, the raw mix starts a decarbonation process.

To assist in this process, an additional burner is added in the lowest cyclone to keep the gases at a higher temperature, between 800°C and 1000°C [7, 8, 9].

Figure I.10 : Preheating Tower

This technique is called "precalcination" because the material reaches the kiln entrance with a temperature close to the firing temperature.

b. Rotary Kiln :

The rotary kiln is the centerpiece of the cement plant. It consists of a cylindrical shell that is 74 meters long and 5.2 meters in diameter, protected by refractory bricks, and tilted at a 3% angle relative to the horizontal. The flame, the main element of the kiln, reaches a temperature of about 2000°C and is generated by the combustion of natural gas.

Figure I.11: Rotary Cement Kiln

The raw mix enters the upstream part of the kiln. This section helps to raise the temperature of the raw materials to about 800-900°C, thus completing the decarbonation of limestone. The material then moves to the clinkerization zone, where temperatures reach approximately 1450°C. At this temperature, reactions between lime and oxides (SiO_2, Al_2O_3, Fe_2O_3) form silicates, aluminates, and calcium alumino-ferrites, which constitute the clinker.

c. Cooler :

At the exit of the rotary kiln, the clinker must be cooled to preserve its chemical formula, by a cooler has grates it is equipped with a battery of powerful fans supplying cooling air.

The role of the cooler is to lower the temperature of clinker falling from the furnace to a temperature from about 1135 c up to 80-100 c, the clinker will progress inside thanks to the Repeated strokes of the grids on which it rests, through the grids of powerful fans will blow under the clinker to cool it andtransported with a clinker silo conveyor. [1].

Figure I.12: Cooler

d. Clinker Storage:

The clinker appears as small granules, roughly the size of a hazelnut. It is conveyed by belt conveyors to the clinker silo, where it is stored until it is ready to be ground into cement. The storage capacity of each silo is 40,000 tons [4].

Figure I.13: Clinker Silo

I.3.4 Cement Zone :

a. Clinker Grinding :

Clinker grinding is carried out continuously in grinders fed from the clinker stocks and other components and additives.

The grinding process aims to reduce the clinker aggregates into powder and, on the other hand, to add gypsum (which regulates the setting process), along with any other potential components (such as pozzolana, limestone, etc.). This allows for the production of different standardized cement types .

This type of grinder, which allows for the grinding of clinker with the addition of additives, is called a "cement grinder" [5].

Figure I.14: Cement Grinder

b. Storage :

At the output of the grinder, the cement is transported to silos (there are six silos) with a capacity of 20,000 tons each.

Figure I.15: storage silos

I.3.5 Packaging Zone:

Cement is transported to storage silos by air or mechanical transport. Each storage silo feeds filling stations for packed cargo in bags (. Bags usually contain 25 or 50 kg) or loading and delivery systems for bulk shipments. Shipping includes storage of cement, packaging for delivery in bags or large quantities, and loading on transport vehicles (trucks). This represents the factory interface with the customer.

The methods used to ship cement can be seen in the following images.

Figure I.16: Small bag Packaging

Figure I.17: Big bag Packaging

Figure I.18: Raw Cement Packaging (Clinker)

I.4 Conclusion:

In this chapter, we provided a comprehensive overview of the BISKRIA CEMENT plant, illustrating the different stages of cement production and the basic steps in its preparation. We also reviewed the production line, highlighting key processes that contribute to the efficient functioning of the manufacturing process.

Since production efficiency relies heavily on the precision of industrial process control to ensure production quality and operating efficiency, modern control systems, such as DCS (Distributed Control System), play a pivotal role in industrial process management. In the next chapter, we will talk about these systems, explaining their working mechanisms and their importance in improving and enhancing operational stability and ensuring production according to the required standards.

CHAPTER 2

DCS NETWORK ARCHITECTURE AND SOFTWARE TOOLS UTILIZED IN THE PROJECT

II.1. Introduction :

The cement industry relies on complex processes that require high precision and control to ensure operational efficiency and final product quality. Distributed control systems (DCS) are the cornerstone of industrial automation, providing an integrated operating environment that enables real-time monitoring and intelligent control at various stages of production.

DCS systems are capable of integrating hardware and software into a distributed network, allowing for efficient process coordination and data management. Programmable logic controllers (PLCs), and advanced industrial software tools like PCS7 and the TIA PORTAL are all part of these systems.

This chapter provides an overview of the (DCS) in the BISKRIA CEMENT FACTORY, explaining how they work, their main components, the associated industrial software, and the software used to develop our smart application.

II.2. DCS

II.2.1. Definition of Distributed Control System (DCS)

Distributed Control Systems (DCS) are advanced control systems used in industrial automation and process control. They play a crucial role in managing complex processes, ensuring safety, and optimizing efficiency.

DCSs are used in several industrial facilities like pharmaceutical manufacturing, oil refineries, chemical plants, and petrochemical plants.

DCS allows prompt monitoring and control of various plant processes by distributing control tasks across multiple controllers. DCS provides a more

flexible and scalable solution compared to traditional centralized control systems. This enables industries to adapt to changing requirements and maintain high levels of performance and reliability.[6]

II.2.2 Architecture of Distributed Control System

As the name suggests, DCS has three main qualities. The first one is the distribution of various control functions into relatively small sets of subsystems, which are of semiautonomous, and are interconnected through a high-speed communication bus. Some of these functions include data acquisition, data presentation, process control, process supervision, reporting information, storing and retrieval of information.

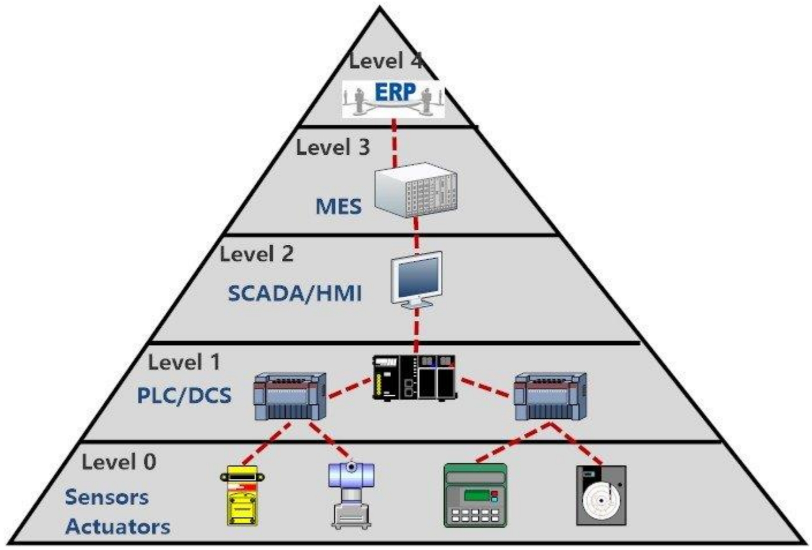


Figure II.1: ISA-95 Architecture for Industrial Automation Systems

The second attribute of DCS is the automation of manufacturing process by integrating advanced control strategies. And the third characteristic is the arranging the things as a system. DCS organizes the entire control structure as a single automation system where various subsystems are unified through a proper command structure and information flow.[7]

A Distributed Control System consists of several key components that work together to monitor and control an industrial process plant. These components include controllers, input/output (I/O) modules, human machine interfaces (HMI), and communication networks.[8]

As shown in Figure II.1. bellow, the information and communication side of existing industrial automation control systems are traditionally depicted as a "Pyramid" following the ISA-95 standard. In the ISA-95 pyramid, 5 layers are defined from top to bottom including enterprise resource planning (ERP), manufacturing execution systems (MES) (level 3), supervisory control and data acquisition (SCADA) (level 2), the controller (PLC) and Input/output units (level 1) And finally field level (sensors and actuators) (level 0). [9]

The structure of the distributed control system (DCS) is typically divided into multiple functional levels, each with specific roles in the system as a whole. Below is the distribution system structure of the BISKRIA CEMENT FACTORY:

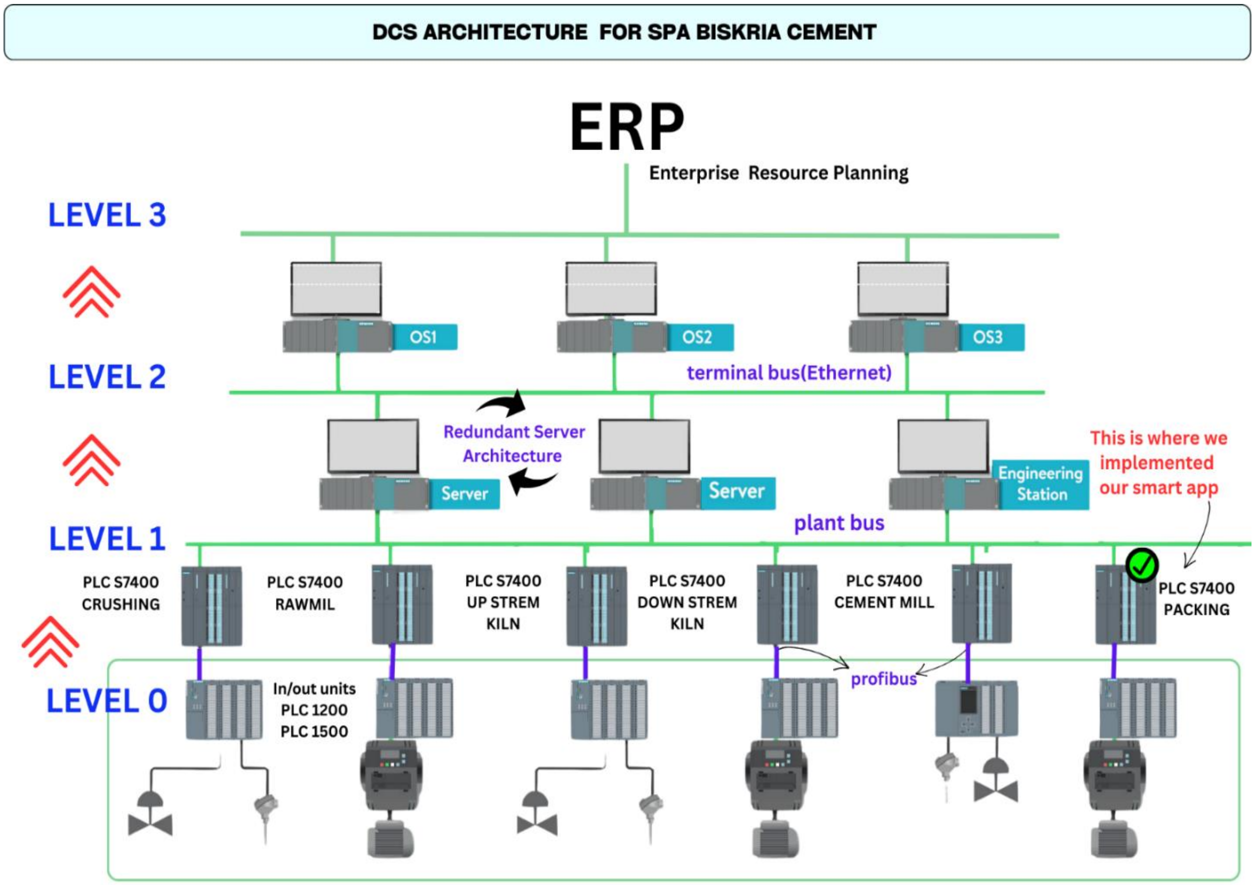


Figure II.2: DCS ARCHITECTURE FOR SPA BISKRIA CEMENT

II.2.3. A description of a DCS system's hardware and software:

A DCS is a combination of the concept of a simple local control loop and computer networks. Distributed control systems have enormously improved the management of industrial procedures with more flexibility and safety. DCS allow centralized driving and surveillance and control.

A DCS refers to a process control system, in which the regulatory elements are not centralized (such as the brain) but distributed with each subsystem under the control of one or more regulators. The entire elements of the system can be network connected to perform communication, driving and surveillance functions.[10]

The DCS at the BISKRIA CEMENT FACTORY is made up of a number of industrial computers that serve as an interface for controlling and monitoring different production stages. These computers are connected to controllers that are based on microprocessors to process production-related data. The system and field equipment, such as sensors and triggers dispersed across the production units, are interfaced by input/output units. Additionally, industrial contacts that facilitate communication with other subsystems based on programmable automation have been included, along with a variety of field and information communication networks that guarantee the simplification of data transmission between units. The flexible and sophisticated structure of this system at the Biskara facility enables precise data collection and processing, supporting ongoing monitoring and enhancement of the production process's quality.

Includes typical architecture for any DCS system:

- A set of computers used as a driving and supervision interface.
- Input/output cards used as an interface with the industrial process.
- Communication interfaces used to communicate with other PLC-based subsystems.
- Communication networks (field, control and information).

The DCS can be connected in parallel with other systems (ESD: Emergency Shut Down "Emergency stop"; F&G: Fire & Gas "Fire protection") or with PLC by the different communication protocols.[11]

The PLC used in the distributed control system (DCS) is a Robust type, ensuring stable and continuous process control. This means it is highly robust and can withstand harsh industrial conditions such as vibration, high temperatures, and electrical disturbances. These devices are already in use at the biskria cement plant due to their reliability and suitability for the local industrial environment.

II.2.4. The advantages of DCS systems

The major advantages of functional hardware distribution are flexibility in system design, ease of expansion, reliability, and ease of maintenance. A big advantage compared to a single-computer system is that the user can start out at a low level of investment. Another obvious advantage of this type of distributed architecture is that complete loss of the data highway will not cause complete loss of system capability. Often local units can continue operation with no significant loss of function over moderate or extended periods of time. Moreover, the DCS network allows different modes of control implementation such as manual/auto/supervisory/computer operation for each local control loop. In the manual mode, the operator manipulates the final control element directly. In the auto mode, the final control element is manipulated automatically through a low-level controller usually a PID. The set point for this control loop is entered by the operator. In the supervisory mode, an advanced digital controller is placed on the top of the low-level controller.

The advanced controller sets the set point for the low-level controller. The set point for the advanced controller can be set either by the operator or a steady state optimization. In the computer mode.

One of the main goals of using DCS system is allowing the implementation of digital control algorithms. The benefit of digital control application can include:

- Digital systems are more precise.
- Digital systems are more flexible. This means that control algorithms can be changed and control configuration can be modified without having rewiring the system.
- Digital system cost less to install and maintain.
- Digital data in electronic files are easier to deal with. Operating results can be printed out, displayed on color terminals, stored in highly compressed form.[12]

II.3. PCS7

II.3.1. What is the PCS7

SIMATIC PCS7 is a comprehensive process control system offered by Siemens, designed for various industrial applications. It provides powerful engineering, efficient remote-control capabilities, and seamless integration with Siemens' Totally Integrated Automation (TIA) ecosystem. PCS7 enables optimization of operations from ERP and MES levels down to the field level, offering flexibility, scalability, and security. Siemens offers comprehensive training courses on the standard and expansion functions of the PCS7 system to provide a thorough understanding of this automation technology.[13]

Figure II.3: LOGO OF PCS7

II.3.2. The main features of Simatic PCS7:

1. **Engineering**:

The system offers powerful engineering tools for configuring, programming, and maintaining process automation systems. It includes libraries of preconfigured objects and function blocks, as well as a graphical interface for easy configuration and visualization.

2. **Distributed Control System (DCS):**

SIMATIC PCS7 functions as a DCS, allowing for centralized control and monitoring of various field devices, such as sensors, actuators, and controllers. It supports communication protocols like Profibus and ProfNet, enabling seamless integration with a wide range of automation components.

3. **Process Visualization**:

The system provides a user-friendly interface for operators to monitor and control the industrial processes. It offers graphical representations of the process, including real-time data, trends, and alarms, allowing operators to make informed decisions and respond quickly to any abnormalities.

4. **Process Control:**

SIMATIC PCS7 offers advanced control strategies to optimize process performance and ensure safe and efficient operations. It supports various control functions, such as regulatory control, sequential control, and advanced process control (APC), allowing for precise control and automation of complex processes.

5. **Redundancy and Fault Tolerance**:

To ensure high availability and reliability, SIMATIC PCS7 offers built-in redundancy features. It allows for redundant controllers, redundant communication networks, and redundant I/O modules, ensuring continuous operation even in the event of a failure.

6. **Integration with Other Systems:**

SIMATIC PCS7 can be easily integrated with other systems and applications, such as enterprise resource planning (ERP) systems, manufacturing execution systems (MES), and asset management systems. This enables seamless data exchange and provides a comprehensive view of the entire production process.

7. **Safety and Security:**

The system includes comprehensive safety features to protect personnel, equipment, and the environment. It supports safety instrumented systems (SIS) and complies with relevant safety standards. Additionally, SIMATIC PCS7 incorporates robust security measures to protect against unauthorized access and cyber threats.[14]

II.3.3. industrial applications for pcs7

Siemens' SIMATIC PCS 7 is a cutting-edge industrial automation tool, used in a wide range of vital applications such as emission control, operational quality and operational safety. It has proven successful in many industries, including the cement industry; Biskara Cement Plant relies on this system to enhance performance efficiency and ensure product quality and continuity of operations.

The following images represent some of the uses of this system in the Biskria factory.

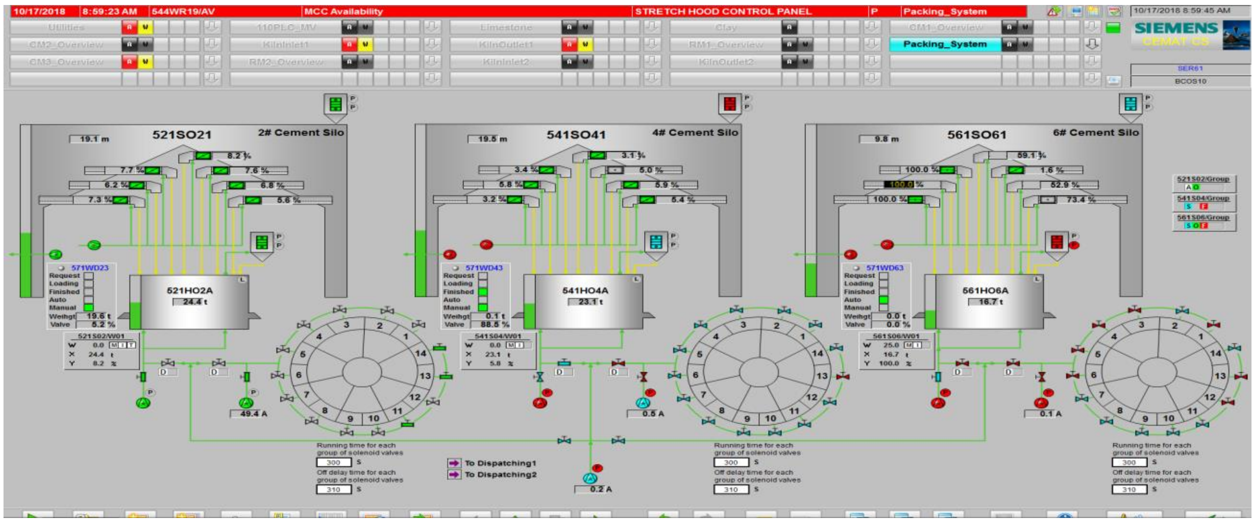


Figure II.4: SILO PLAN FROM BELOW

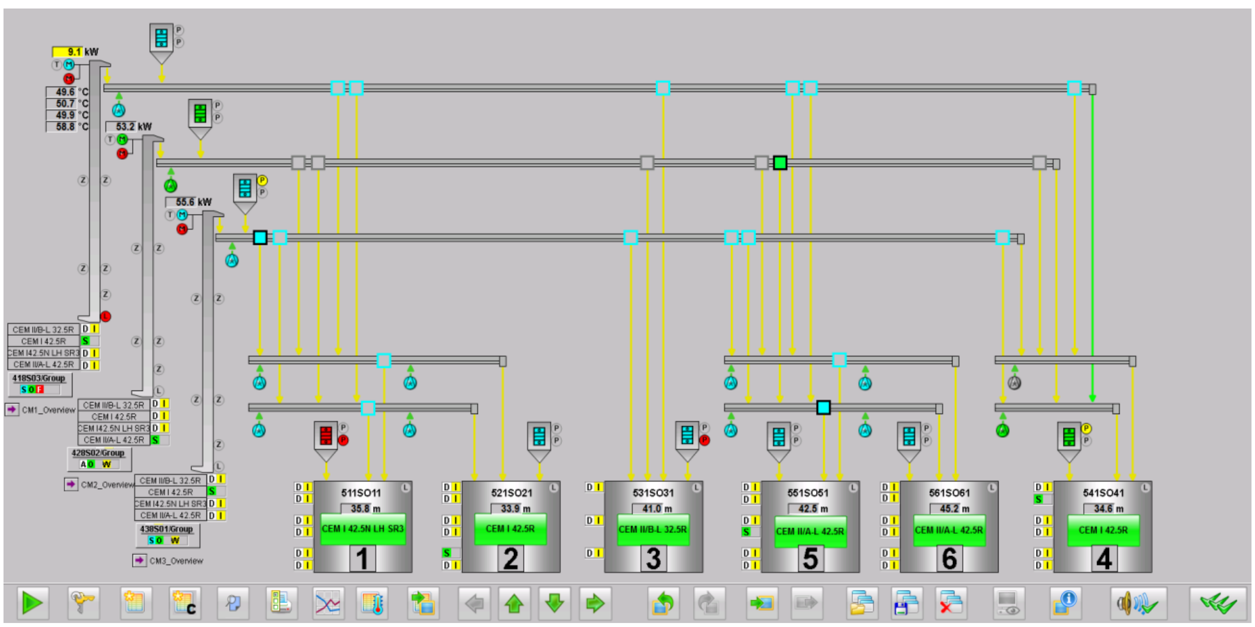


Figure II.5: SILO PLAN FROM ABOVE

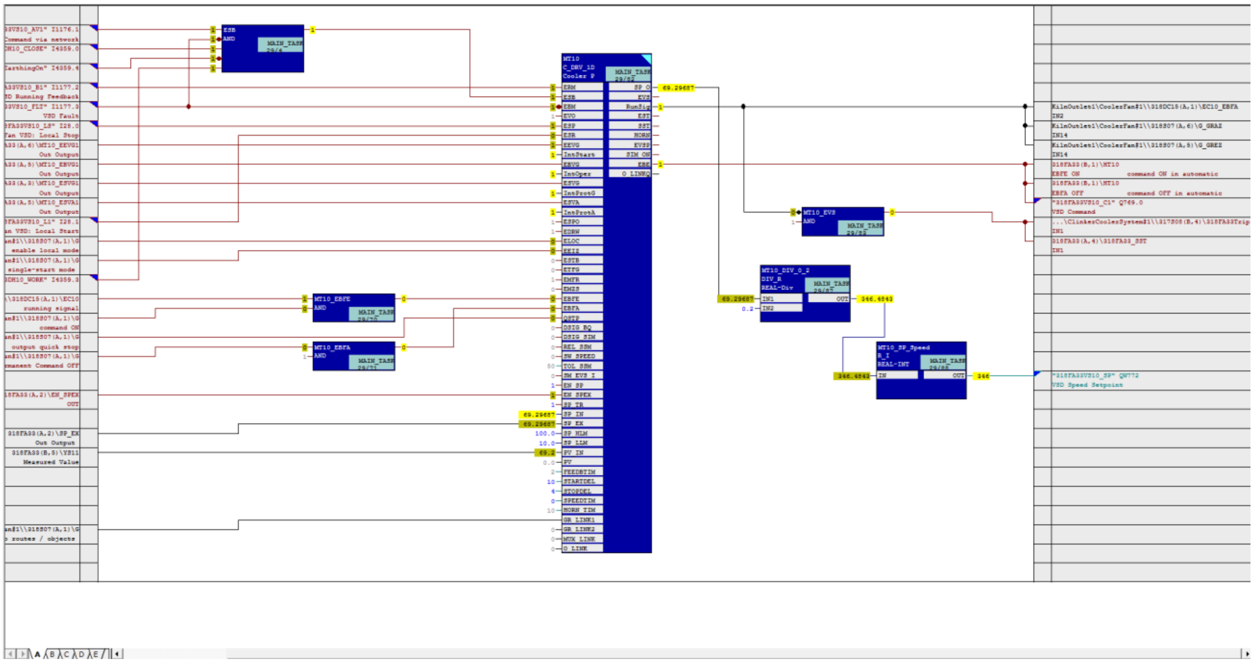


Figure II.6: CDRIVE

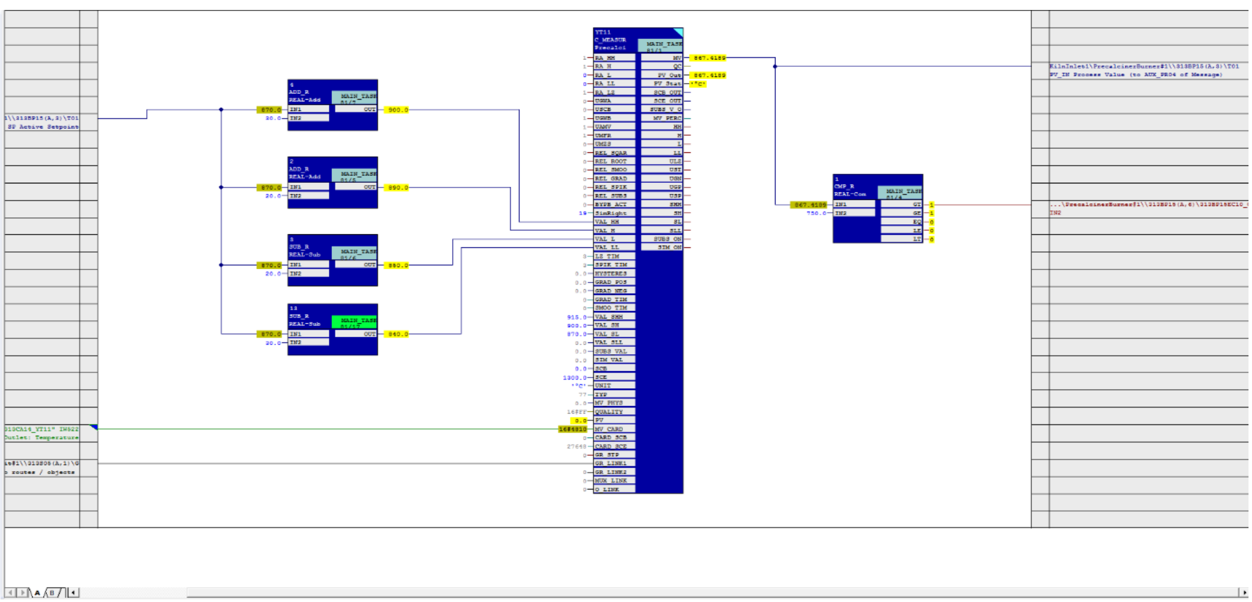


Figure II.7: CMEASURE

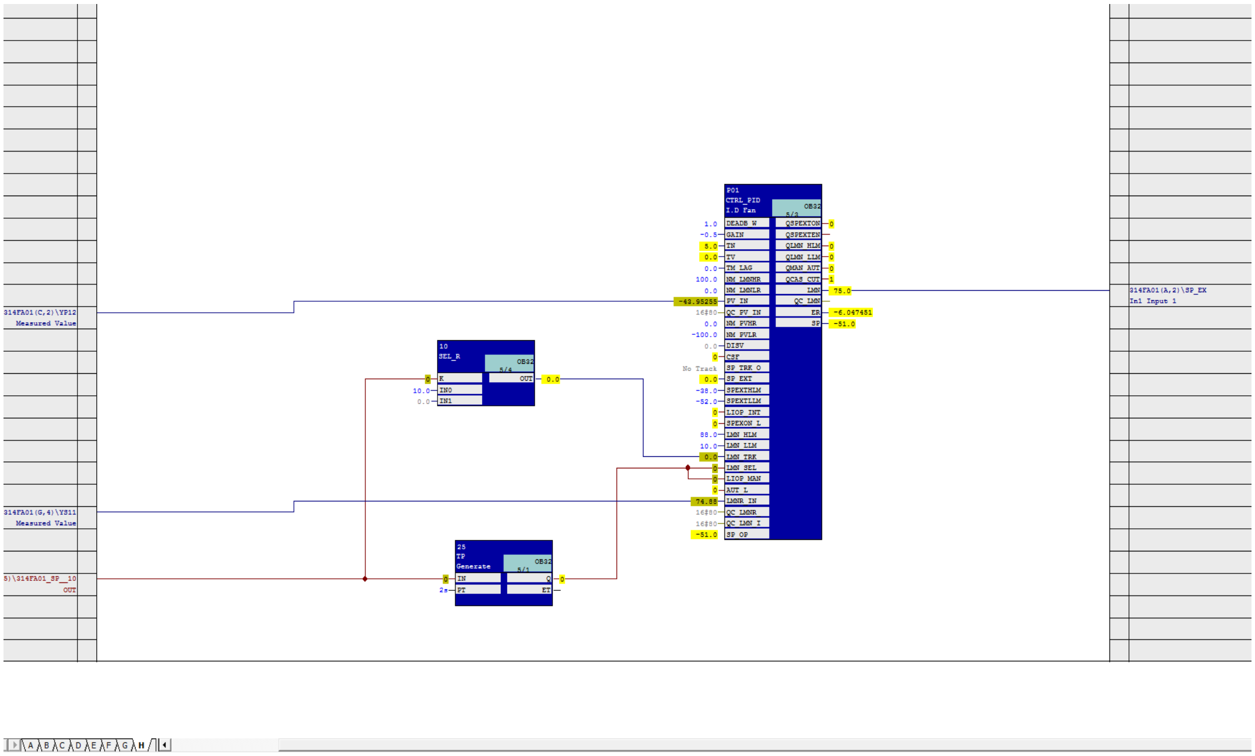


Figure II.8: PID

II.4. TIA PORTAL

II.4.1. Overview of Siemens TIA Portal

The TIA Portal is an integrated engineering platform from Siemens that provides efficient software development, testing and commissioning for automation systems. It enables the management of PLCs, HMIs, drives and peripherals within a single framework. TIA Portal improves software quality and reduces engineering costs through agile development methods and scalable solutions. New subscription models offer flexibility for individual needs and improve efficiency, adaptability and security in automation. [15]

Figure II.9: TIA PORTAL LOGO

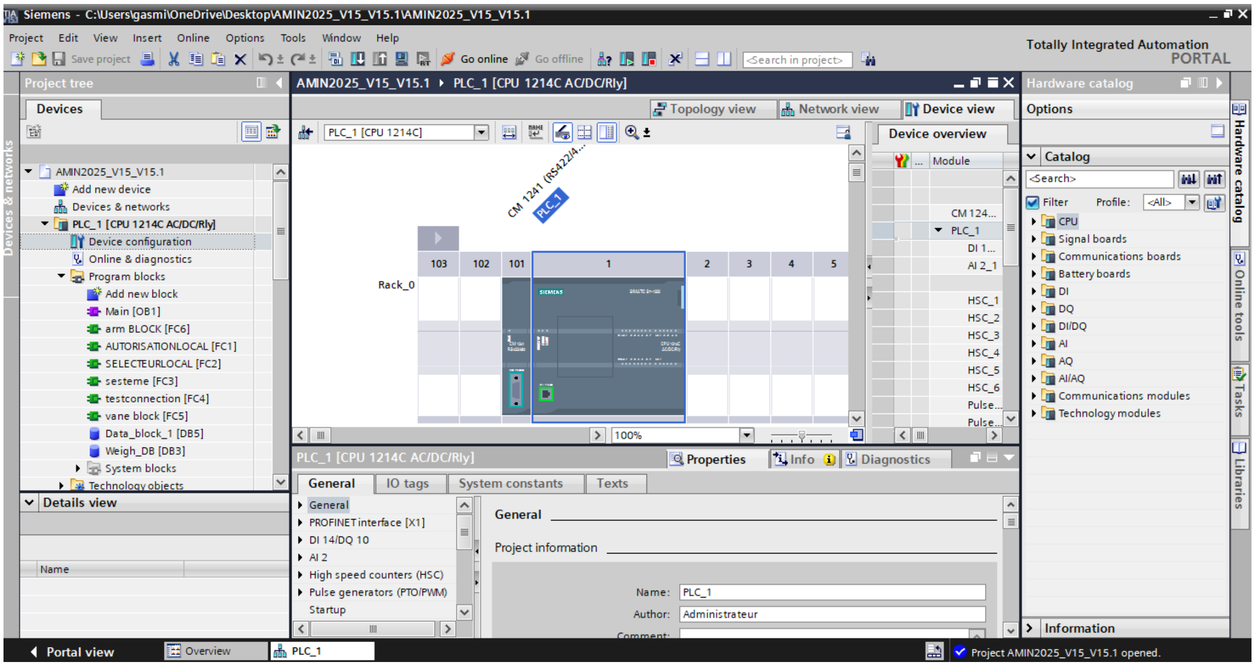


Figure II.10: Project view in TIA PORTAL V15.1

At the Biskria cement plant, the TIA Portal is a key component in driving industrial automation. It offers robust support for various programming languages—including Ladder Logic and Structured Control Language (SCL)—to meet diverse operator and engineering requirements. This platform ensures smooth, real-time communication between the plant’s automation components, significantly boosting operational efficiency and system reliability. Its rich library of pre-configured functions and templates allows engineers to swiftly deploy and customize control solutions tailored to the specific challenges of cement production. Furthermore, the open architecture of the TIA Portal enables easy integration with third-party devices, thereby broadening its application potential and reinforcing its role as an indispensable tool in the plant’s automation strategy.

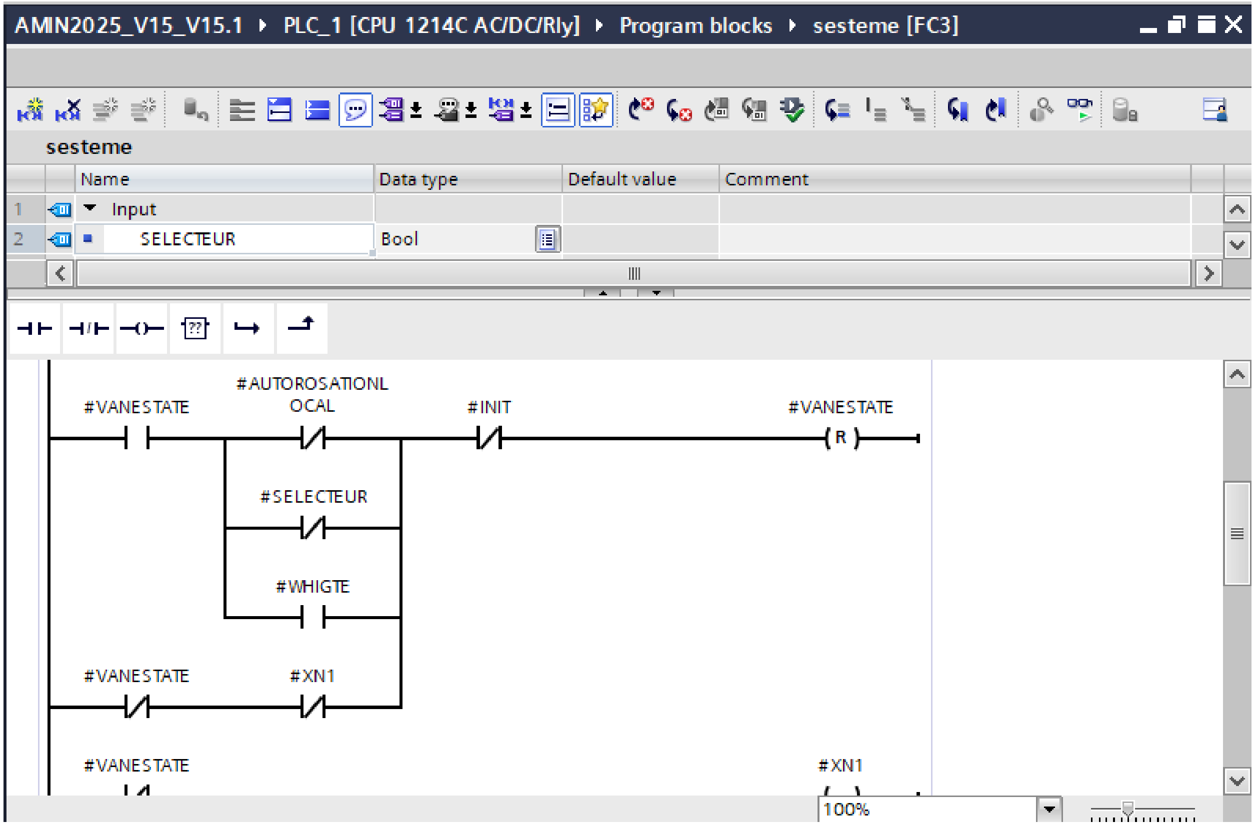


Figure II.11: Language ladder in the project

II.4.2. Benefits of Using TIA Portal

Proponents highlight numerous benefits of the TIA Portal approach compared to a collection of separate engineering tools:

- Reduced engineering time and costs through an integrated toolchain and automation.
- Easier network configuration through automatic topology detection.
- Faster commissioning and plant modifications enabled by bulk download of hardware configurations.
- Improved productivity through unified data models and consistent interfaces.
- Enhanced visibility into plant status with integrated diagnostic tools.
- Lower training costs due to a common development environment.
- Decreased risk of errors or data inconsistency across tools.
- Centralized data management for version control, backup and collaboration.
- Reusable standardized libraries encourage programming best practices.

- Scalability to large automation projects with thousands of hardware devices.
- Tighter integration between SIMATIC components without external interfaces by following rigorous testing methodologies, Siemens aims to deliver high reliability and quality standards for TIA Portal. Frequent updates incorporate customer feedback into ongoing improvement.[16]

II.4.3. Applications of TIA Portal in Various Industries

TIA Portal, a comprehensive automation software platform by Siemens, offers versatile applications across various industries. Its integration capabilities facilitate seamless operation in manufacturing, energy, and transportation sectors. By providing a unified engineering environment, TIA Portal enhances productivity and reduces time-to-market for industrial projects.

In manufacturing, TIA Portal enables efficient control and monitoring of production lines. Its advanced simulation tools allow engineers to test and optimize processes before implementation, minimizing downtime. The software also supports Industry 4.0 initiatives, promoting smart manufacturing practices.

In the energy sector, TIA Portal aids in the management of power distribution networks and renewable energy systems. Its robust data analytics capabilities provide insights for improving energy efficiency and reliability. Additionally, TIA Portal's interoperability with other Siemens products ensures comprehensive solutions for diverse industrial needs.[17]

II.4.4. Recent modifications

Version 20: It has advanced tools such as new PLCs support, cloud integration and artificial intelligence, leading to performance and collaboration.

These features make TIA Portal a powerful tool in industrial accounting, combining integration, efficiency and security in one platform.[18]

II.5. Python

II.5.1. What is Python?

The Python programming language was created in 1989 by **Guido van Rossum** in the Netherlands. The name Python comes from a tribute to the TV series Monty Python's Flying Circus Which G. van Rossum is a fan of. The first version public of this language was published in 1991.

The latest version of Python is version 3. Specifically, version 3.11 was released in October 2022. The Python version 2 is outdated and no longer maintained, avoid using it.

The Python Software Foundation 1 is the association that organizes the development of Python and animates the community of developers and users.[19]

Figure II.12: LOGO OF PYTHON

II.5.2. Python Advantages

Python has many features including:

- **Simple** Python is not difficult to code and simple to peruse when contrasted with different dialects like Java, C, C++, and so on Python sentence structure can be examined by anybody during a brief timeframe. Python code resembles English that permits the student to zero in on the outcome.
- **Expressive** Python can execute a convoluted capacity with a couple of lines of code contrasted with different dialects.
- **Free and Open-Source** Python is open source and uninhibitedly accessible. The general population can help and add to the improvement of the language. The Python source code can be downloaded, changed, utilized, and be dispersed.

- **Significant level language** Python is an undeniable level language. There is no requirement for recollecting the engineering and memory the board which makes python entirely great.
- **Deciphered Language** Python is a deciphered language. The code doesn't require aggregation, they are executed line by line and not all at a time which makes troubleshooting the code more straightforward than the wide range of various dialects. Along these lines
- **Object-Oriented** Python upholds an article arranged approach which assists the software engineer with composing reusable code and help in fostering the application with lesser code.
- **Extensible and installed** the extensible property of Python permits code to be composed and ordered in different dialects like C or C++. This code then, at that point, can be utilized further in Python when required. The inserted property of Python permits is permitting the utilization of Python in another programming language.
- **Enormous Standard Library** Python alongside the enormous standard library accommodates a huge scope of modules and capacities. So the software engineer doesn't have to compose the code, they can simply import it **Powerfully Typed** Python is supposed to be a powerfully composed language since it doesn't have to indicate the information sort of the variable while proclaiming it. The sort of significant worth is chosen during the run time.
- **Web improvement** Python is a go-to language for web improvement. Django, Pyramid, Flask, Bottle are a portion of the structures presented by Python. Python web structures are well known for their security, adaptability, and adaptability. Demands, Beautiful Soup, Paramiko, Feedparser, Twisted Python, and so on, are libraries that additionally remembered for the Python's Package Index.
- **Game turn of events** Python has many in-constructed libraries that are good for fostering a game. PyGame, PyKyra are structures for game turn of events and PySoy is a 3D cloud game motor for Python3.
- **Man-made brainpower and Machine Learning** Man-made brainpower and Machine Learning is one of the rising themes and will be gone on from now on. Python is famous and positive to be utilized in Artificial insight and AI because of its personality of being steady, secure, adaptable, and of its different

devices. Some of Python libraries and structures utilized in Artificial Intelligence are SciPy, Pandas, Seaborn, Keras, TensorFlow, Scikit-learn, NLTK, Pytorch, Accord.NET, and so forth. [20]

II.5.3. Integration and Compatibility

Python plays well with others. Its ability to integrate with other languages and technologies is a significant advantage. Python can invoke C/C++ libraries, communicate with Java code, and seamlessly interact with databases or web services. This interoperability makes Python a universal tool in a developer's arsenal, enabling complex, multi-technology systems to be created easily.[21]

Python serves as a bridge between databases and industrial control systems because of its adaptability, where it Take data from the PLC using Modbus or Snap7.

II.6 MySQL

II.6.1. What is MySQL?

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The MySQL website (http://www.mysql.com/) provides the latest information about MySQL software.[25]

Figure II.16: MYSQL

II.6.2. The main features of MySQL are:

One of the features of My SQL:

- **Open Source & Free of Cost:** It is Open Source and available at free of cost.
- **Portability:** It can be installed and run on any types of Hardware and OS like Linux or Mac etc.
- **Security:** It creates secured database protected with password.

- **Connectivity** It may connect various types of Network client using different protocols and Programming Languages.
- **Query Language** It uses SQL (Structured Query Language) for handling database.[26]

II.6.3. Industrial data management using MySQL

MySQL plays an important role in industrial data management thanks to its scalability, reliability and ability to process complex queries efficiently. Here are MySQL's most important apps and features in industrial environments:

II.6.3.1. MySQL applications in the industry

- **Real-time data processing:** MySQL can be integrated with IoT devices and sensors in manufacturing plants to collect and process real-time data for predictive maintenance and surveillance.
- **Data repositories:** Used to store large amounts of organized data from industrial processes, allowing for advanced analysis and reporting.[27]

II.7. Protocol:

II.7.1. Definition of the protocol

The author's definition of a protocol is a method for digital data communications between two or more devices in different locations, or on a network.[28]

Globally, there are a variety of industrial protocols that are employed in control systems such as PLCs and DCS, as well as between connected devices within the network, such as controllers, sensors, and triggers. Some of these protocols are for data transfer only, usually used in sensor networks and field areas, while others are used for both data transfer and implementation of control orders, allowing for effective integration of the various components of the industrial system and enhancing its stability and performance.

II.7.2. Communication Protocols and Network Integration

Modern automation systems mainly rely on industrial communication protocols and data networks to ensure reliable and rapid information exchange, achieving scalable operations within various industrial environments. Protocols such as Ethernet/IP, Profibus and Modbus are among the most commonly used standards in this area, as they provide efficient communication and easy integration between system components. Flexible and scalable network designs also enable future expansion requirements to be met without affecting performance stability or system reliability.[29]

II.7.3. Network Architecture

Industrial networks typically adopt a hierarchical structure aligned with the OSI model, integrating:

- **Field-level devices**: Sensors, actuators, and I/O modules using protocols like IO-Link or ASi-5.
- **Control layer**: PLCs and SCADA systems communicating via Ethernet/IP or PROFINET for real-time control.
- **Enterprise layer**: Cloud integration and data analytics using OPC UA or HTTP/HTTPS.

II.7.4. Types of industrial networks

- ❖ **Ethernet/IP:** Utilizes standard Ethernet technology for high-speed communication and seamless integration with devices.
- ❖ **PROFINET**: A powerful, open Industrial Ethernet standard providing real-time communication and flexibility.
- ❖ **Modbus TCP/IP:** A widely used protocol for connecting industrial devices over Ethernet.
- ❖ **Device Net:** An industrial network protocol designed for connecting simple devices and sensors in a network. [30]

II.8. Conclusion

This chapter reviewed the detailed structure of distributed control systems (DCS), explaining its primary role in industrial automation, operating management, and achieving smooth integration between different factory components. The main elements of these systems, including control units (PLCS), and advanced software tools, are also highlighted, as well as data analysis role in promoting performance and improving decision -making.

Although DCS provides a stable and advanced operating environment, modern developments in artificial intelligence and data processing provides additional opportunities to improve performance and enhance the level of automation. Accordingly, the next chapter focuses on the applied aspect of this study, where a smart application that benefits from the technologies of computer vision and databases will be developed to improve truck verification processes, which contributes to raising the efficiency of the system and reducing operational errors. In this chapter, we will discuss the steps of development, tools used, and the mechanism of integration of application with the current system, which constitutes an important step towards developing a more intelligent production environment and in response to the requirements of modern industry and more improvement and innovation in industrial automation.

CHAPTER 3

DEVELOPMENT OF A SMART APPLICATION FOR A BULK CEMENT PACKAGING SYSTEM

III.1. Introduction

This chapter presents the practical aspect of the project, which aims to develop a smart application to control the bulk cement filling method, utilizing High-level programming tools in both IT and OT technologies. The application addresses a real-world issue observed at the Biskria Cement Plant—namely, the lack of a reliable method to verify that the customer has been supplied with the same type of bulk cement they ordered, according to the purchase database and the invoice they paid.

Unlike packaged cement which can be easily identified by the color of bag used, Verifying the conformity of bulk cement with the customer's order is challenging prior to the departure of the customer's truck.

The smart system was designed to enhance the accuracy and efficiency of the filling process by integrating TIA Portal for programmable logic controller (PLC) management, OpenCV for image processing and visual recognition, and MySQL for real-time data storage and traceability. The application was developed using the Python programming language, serving as the intermediary between the PLC and the database. It handles data acquisition, implements control logic, and provides a user-friendly interactive interface. This system contributes to improving the bulk cement filling process, reducing human errors, and ensuring complete consistency in the bulk cement filling process at the plant.

III.2. System Overview

The **Smart Cement Loading Application** is the outcome of a graduation project aimed at automating the process of loading raw cement into transport trucks at the loading packing of the Biskria cement plant. The application was developed to achieve effective integration between modern software technologies (**Python, MySQL, OpenCV**) and industrial control systems (**TIA Portal, PLC**), with a strong emphasis on enhancing operational accuracy, reducing human intervention, and ensuring full traceability of each loading operation.

The application's workflow can be summarized in the following steps:

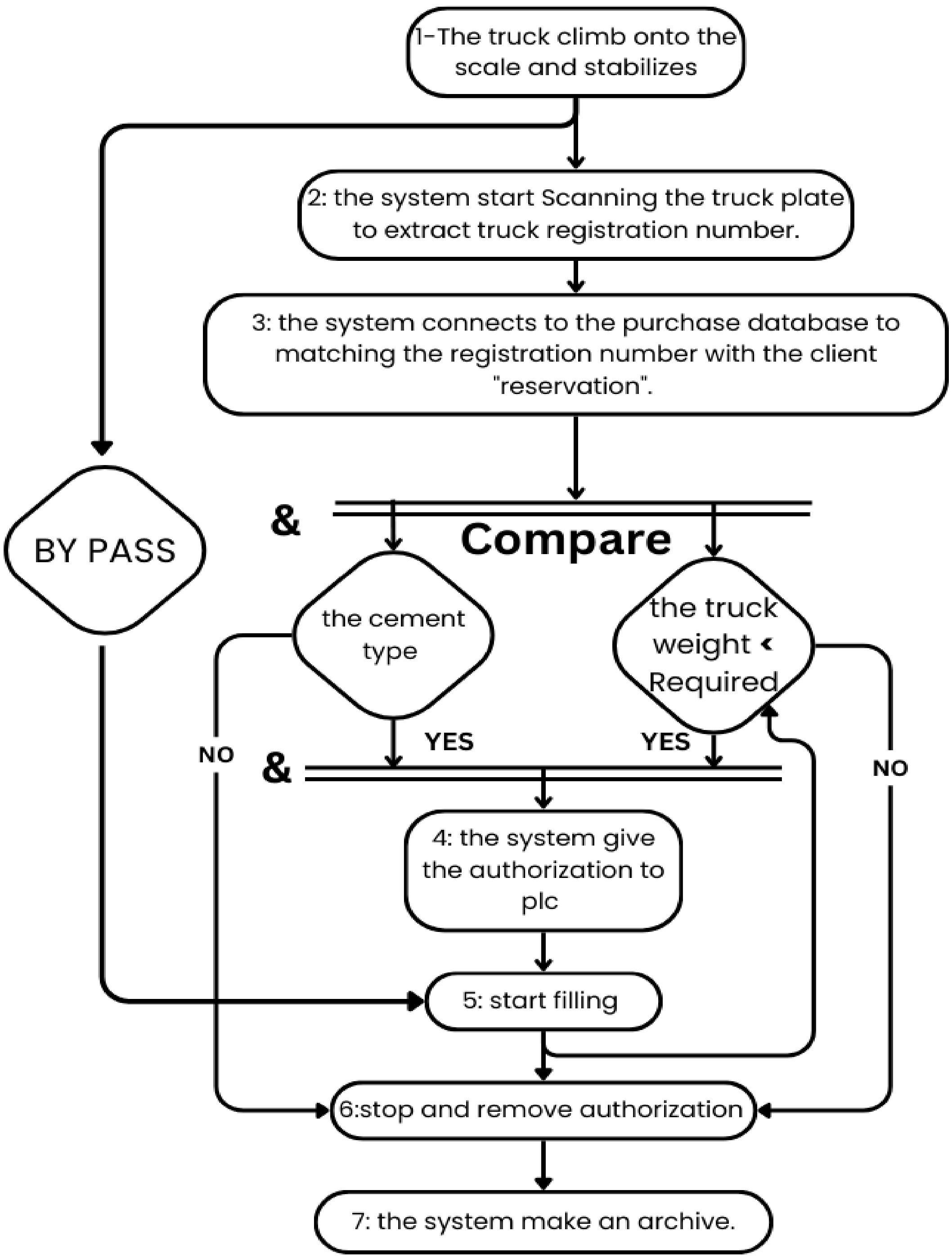


Figure III.1: Cement filling application flowchart

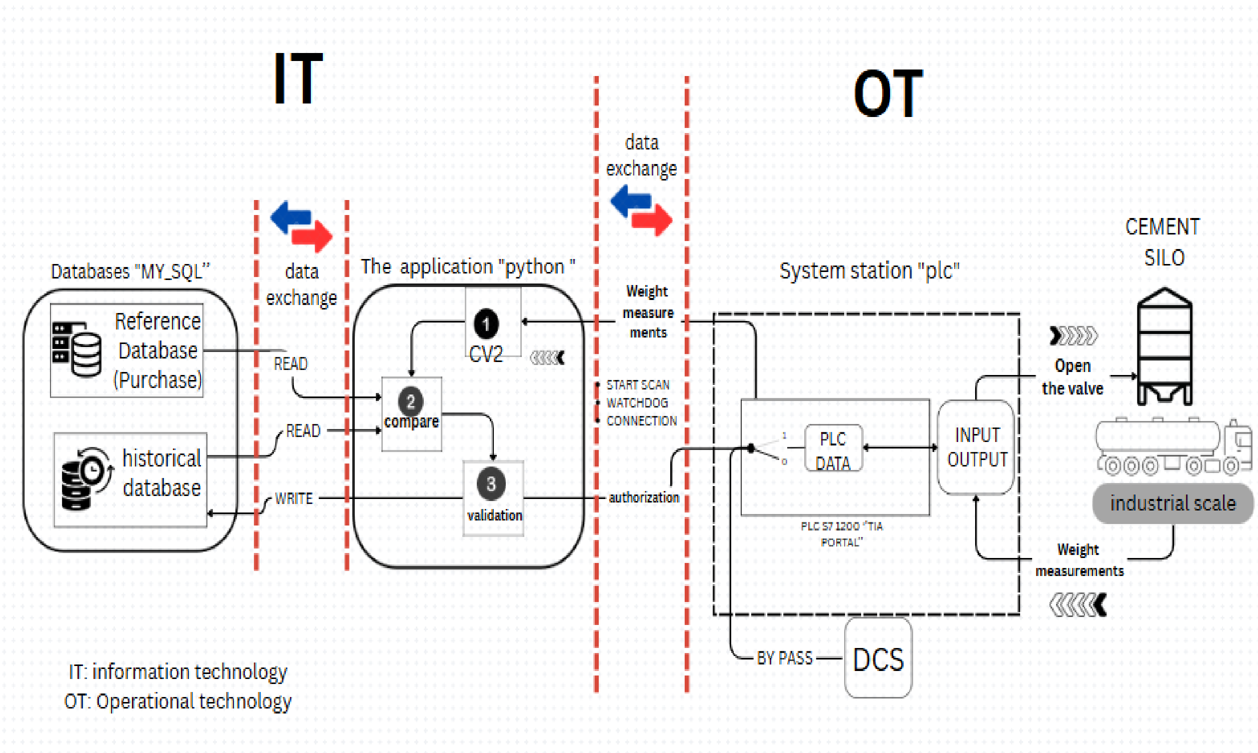


Figure III.2: Smart App Workflow Plan

III.3. Development Environment

This section presents the software tools and hardware components used in developing and implementing the intelligent cement filling application. The development environment was selected based on criteria such as compatibility, performance, and the ability to integrate software and industrial control systems.

III.3.1. Software used :

The development environment for the intelligent raw cement filling application relied on the integration of three main software tools:

- **Python** served as the core programming language for building the application's logic, designing the graphical user interface, handling data flow, and establishing communication with industrial devices.

- **MySQL** was used as the relational database management system to store process-related data such as cement type, filling time, and batch information. This allowed for real-time tracking and structured archiving of operational data.

- **TIA Portal (v15.1)** was employed to program and configure the Siemens S7-1200 PLC, which interacts with the physical components of the system. It played a crucial role in defining control logic and managing signals exchanged between the application and the industrial hardware.

Together, these tools formed a cohesive and flexible development stack that enabled tight integration between digital control, database management, and real-world industrial operations.

III.2.2 Hardware Components :

Device / Sensor	Function
Local PC	It hosts a smart app that manages communication with the database and programmable logic controller (PLC), and provides a graphical interface that helps control and monitor the filling process.
Camera	Takes photos to find the truck's numbering board using OpenCV.
Weight Indicator Scale	Measures the quantity of raw cement loaded into the truck with high accuracy.
Load Cell (Weight Sensor)	It is a sensor used to measure weight or force, and converts it into an electrical signal that can be read via a PLC or other measurement systems.
Siemens S7-1200 PLC	Main controller responsible for executing filling commands

Figure III.3: table of Hardware Components

Figure III.5: The PLC S7-1200

Figure III.4 :
Siemens SIMATIC S7-1200
CM 1241 RS 422 / 485

Figure III.6: weight indicator Scale & Load Cell

S7-1200 : A modular micro-PLC used for small-scale performance requirements and network connectivity options include the multipoint interface (MPI), PROFINET, PROFIBUS, or Industrial Ethernet.

III.4. Database Design:

The database system for the smart cement loading application was designed using **MySQL**, with the aim of organizing and storing all data related to loading sessions in a structured manner that facilitates future tracking and processing.

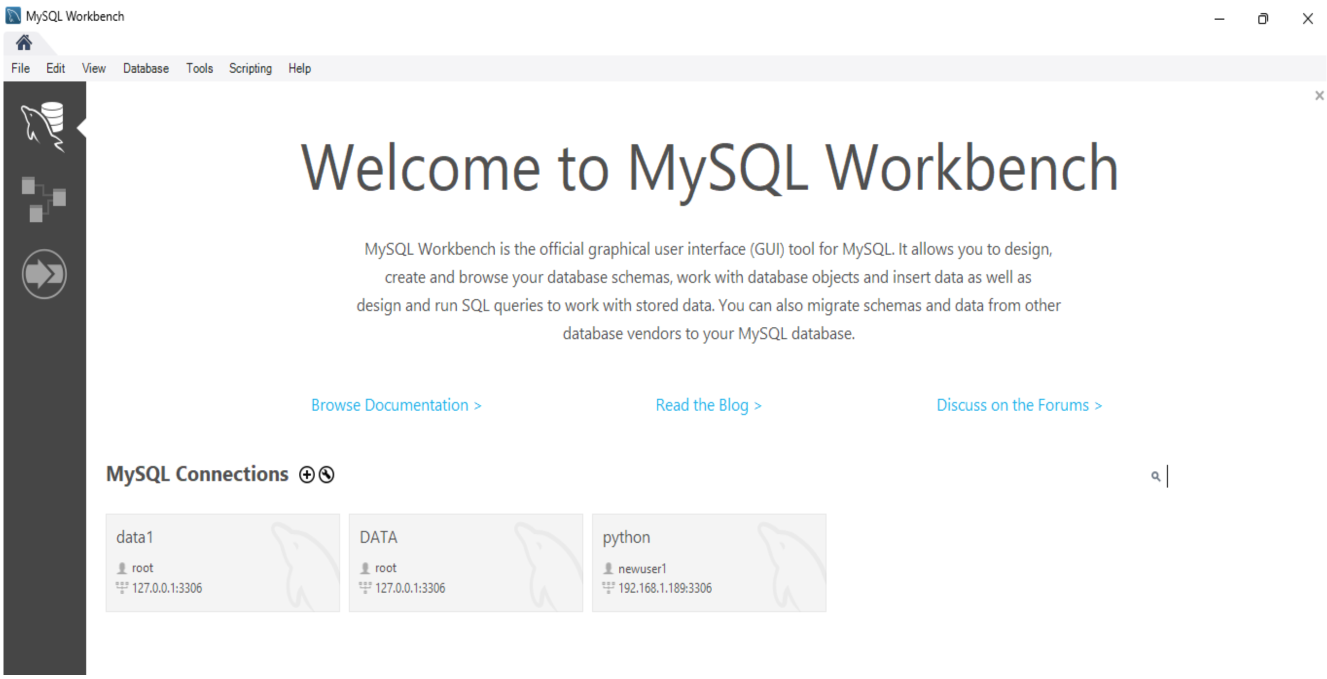


Figure III.7 : My SQL Workbench

This project relies on two separate databases, ensuring a clear separation between commercial processing and industrial operations:

- **The Reference database:**

is dedicated to the commercial aspect. It serves as the primary reference containing all customer information, cement types, and order details. The smart application connects to this database to retrieve the necessary information before initiating any loading process, ensuring that all operations are aligned with the factory's official commercial system.

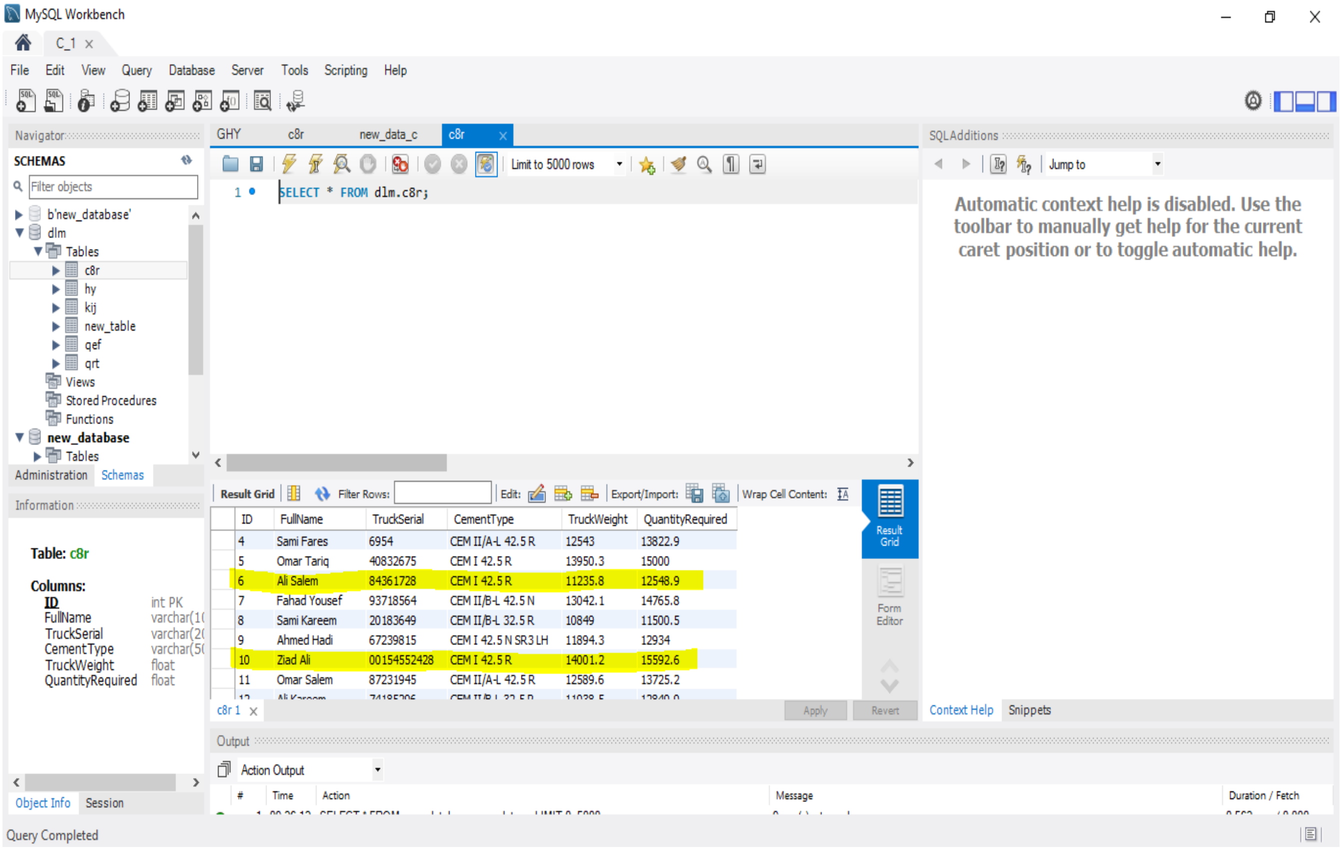


Figure III.8: Reference Database Table Overview

The Historical database :

is used to record and track actual loading session data. It stores detailed information for each loading operation executed via the application, including the truck number, cement type, target and actual weight, as well as timestamps of the operation. This allows for precise tracking and subsequent performance analysis.

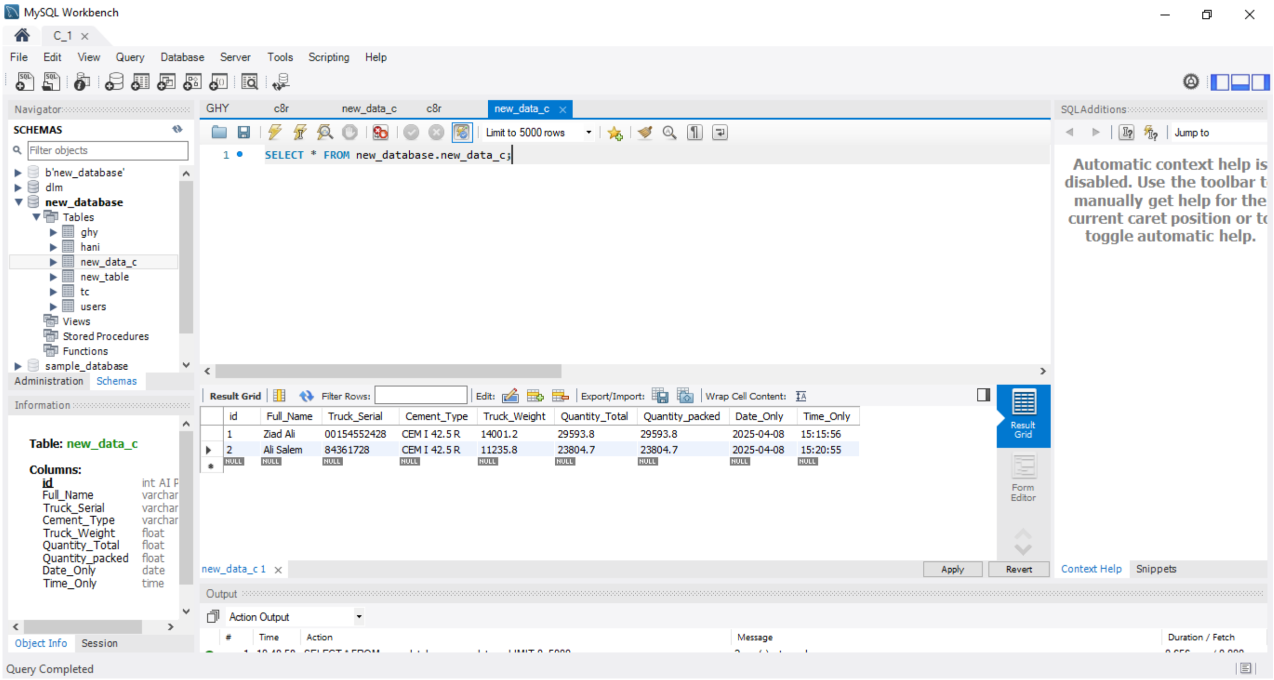


Figure III.9: Historical Database Table Overview

This dual-database structure provides greater flexibility in maintenance and enhances data security and integrity in the exchange between commercial and operational systems.

III.4.1. Steps to Create the Database in MySQL Workbench:

1. **Open MySQL Workbench** and connect to the MySQL Server.
2. From the top menu, click on **"File"** and select **"New Model"** to create a new project.
3. In the **Schemas** section, right-click and choose **"Create Schema"**.
4. Enter the schema name (packing database) and click **"Apply"** to confirm creation.
5. Once the schema is created, expand it and go to **"Tables"** to begin adding the required tables.

6. For each table, define the name, columns, data types, and set the appropriate primary and foreign keys.
7. Click **"Apply"** to generate the SQL statements and create the tables within the schema.

III.5. Smart Application Development with Python:

The Python programming language was used as a central component in developing the smart bulk cement packaging application, acting as an intermediary between the PLC, camera, and MySQL database. This application facilitates data management, decision-making, and direct user interaction through a simplified graphical interface.

III.5.1. Libraries Used in the Project

Several powerful Python libraries were used, the most important of which are:

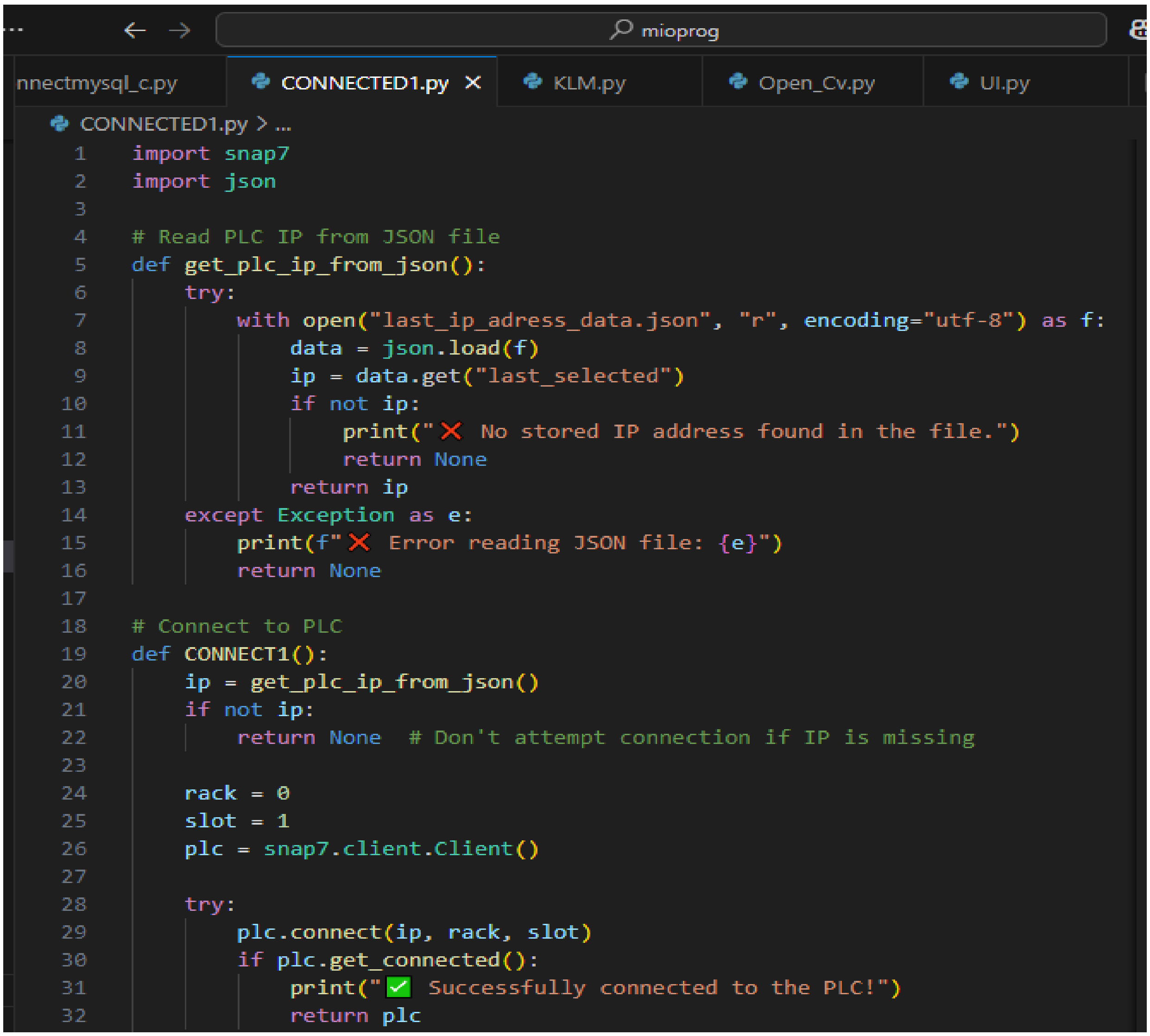
- **cv2 (OpenCV):** For image processing and truck license plate recognition.
- **MySQL. Connector**: For connecting to a MySQL database and reading/storing data.
- **snap7**: For communicating with the PLC via the S7 protocol.
- **Tkinte**r: For creating the application's graphical interface.

III.5.2. Python functions

Python was used to develop the smart application for the bulk cement packaging system, due to its high flexibility and ease of use of various software libraries. Each function within the application was designed in separate code to ensure smooth operation and good organization of the program components.
The application handles several key tasks, most notably :

III.5.2.1. Programmable Logic Controller (PLC) Connection:

The snap7 library was used to create a direct connection to the (PLC). This connection enables data exchange with the data block (DB) in the TIA Portal environment.




```python
import snap7
import json

# Read PLC IP from JSON file
def get_plc_ip_from_json():
    try:
        with open("last_ip_adress_data.json", "r", encoding="utf-8") as f:
            data = json.load(f)
            ip = data.get("last_selected")
            if not ip:
                print("❌ No stored IP address found in the file.")
                return None
            return ip
    except Exception as e:
        print(f"❌ Error reading JSON file: {e}")
        return None

# Connect to PLC
def CONNECT1():
    ip = get_plc_ip_from_json()
    if not ip:
        return None  # Don't attempt connection if IP is missing

    rack = 0
    slot = 1
    plc = snap7.client.Client()

    try:
        plc.connect(ip, rack, slot)
        if plc.get_connected():
            print("✅ Successfully connected to the PLC!")
            return plc
```

Figure III.10: Part of the code for connection to the (PLC)

III.5.2.2. Weight Monitoring:

The application **receives real-time weight values** from the industrial scale via the PLC and displays them instantly on the graphical interface, helping to monitor the progress of the packaging process and precisely adjust it to the required weight.

Figure III.11: Weight reading

III.5.2.3. Truck license plate reading:

Using **the OpenCV (cv2)** library, the application captures a photo of the truck's license plate using a camera connected to the computer. The image is then digitally processed to extract the registration number using **optical character recognition** (**OCR**) techniques. This is then compared to reference data stored in the database to ensure the truck's identity and the type of cement required match.

```python
import cv2
import json
from capture import capture_image
from detect import detect_numbers

LAST_SELECTED_FILE = "last_data_port_selected.json"

def load_last_selected_port():
    try:
        with open(LAST_SELECTED_FILE, "r", encoding="utf-8") as f:
            data = json.load(f)
            return int(data.get("last_selected", 0))
    except (FileNotFoundError, json.JSONDecodeError, ValueError):
        return 0

def show_camera_window():
    camera_port = load_last_selected_port()
    print(f"📷 Using camera port: {camera_port}")

    cap = cv2.VideoCapture(camera_port)
    ret, frame = cap.read()

    if not ret:
        print("❌ Error: Failed to capture image from camera.")
        cap.release()
        return None


    cap.release()

    return frame

```

Figure III.12: Part of the code for OPEN CV (CV2)

III.5.2.4 Database connection:

The application relies on **the MySQL. Connector** library to create a direct connection to the MySQL database, allowing for verification of customer and truck data, as well as automatically recording each filling operation for tracking, statistics, and performance improvement purposes.

mioprog

connectmysql_c.py × | CONNECTED1.py | KLM.py | Open_Cv.py

connectmysql_c.py > connect_to_database

```python
import pymysql
import tkinter as tk
from tkinter import messagebox, ttk
import json
import os

LAST_CONNECTION_FILE = "last_info_C.json"
CONNECTIONS_FILE = "connections_C.json"

def connect_to_database(host, user, password, database):
    try:
        conn = pymysql.connect(
            host=host,
            user=user,
            password=password,
            database=database
        )
        print("✅ Connection successful!")
        return conn
    except pymysql.MySQLError as err:
        print(f"❌ Connection Error: {err}")
        return None


```

Figure III.13: The programming code for MySQL

III.5.2.5 Communication monitoring via Watchdog:

To ensure continuous communication between the application and the PLC, a **watchdog system** has been integrated into the program, sending and receiving a time pulse (bit) periodically. If the pulse is not received within the specified period, the connection is considered lost.

```python
import time
from snap7.util import set_bool
from CONNECTED import connect1  # 📡 استيراد دالة الاتصال

def send_watchdog_signal(plc):
    """كتابة قيمة 1 ثم 0 في كلب الحراسة Watchdog_Bit"""
    try:
        DB_bytearray = plc.db_read(1, 0, 1)  # قراءة البيانات من DB1
        set_bool(DB_bytearray, 0, 0, True)  # ضبط Watchdog_Bit إلى 1
        plc.db_write(1, 0, DB_bytearray)  # كتابة التحديث
        print("🔄 تم تحديث كلب الحراسة إلى 1")

        time.sleep(1)  # انتظار مؤقت قبل التبديل إلى 0

        DB_bytearray = plc.db_read(1, 0, 1)  # إعادة القراءة
        set_bool(DB_bytearray, 0, 0, False)  # ضبط Watchdog_Bit إلى 0
        plc.db_write(1, 0, DB_bytearray)  # كتابة التحديث
        print("🔄 تم تحديث كلب الحراسة إلى 0")

    except Exception as e:
        print(f"⚠ LOST WATCH_DOG: {e}")
```

Figure III.14: Part of the code for watchdog

III.5.2.6 Graphical User Interface

The application's graphical interface is designed using the Tkinter Library, the standard library in Python for the development of graphic user interfaces (GUI). This interface allows the user to easily interact with the packing system, such as display the connection status, monitor the momentary weight, and enter control orders.

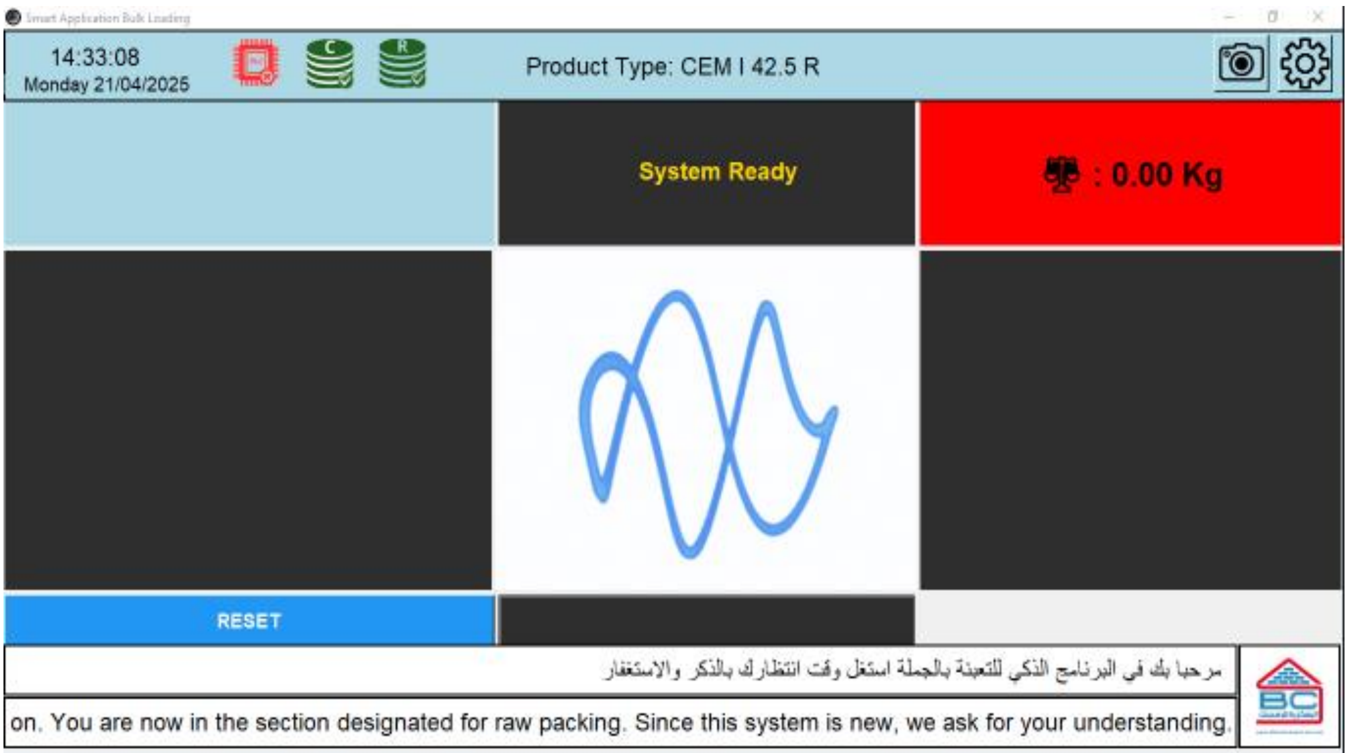


Figure III.15 : User Interface

III.6. Programming the PLC Using TIA Portal

III.6.1 Hardware Configuration

Once the project is created on the Tia Portal, we can configure the workstation. The first step is to identify the existing hardware. To do this, go to "Project View" and click "Add Device" in the Project Browser. A list of components that can be added (PLC, HMI, computer system) will appear. We will start by selecting the central processing unit (CPU) and then add additional modules (power supply, digital or analog I/O module, communications module, etc.). The figure below illustrates the hardware configuration and setup.

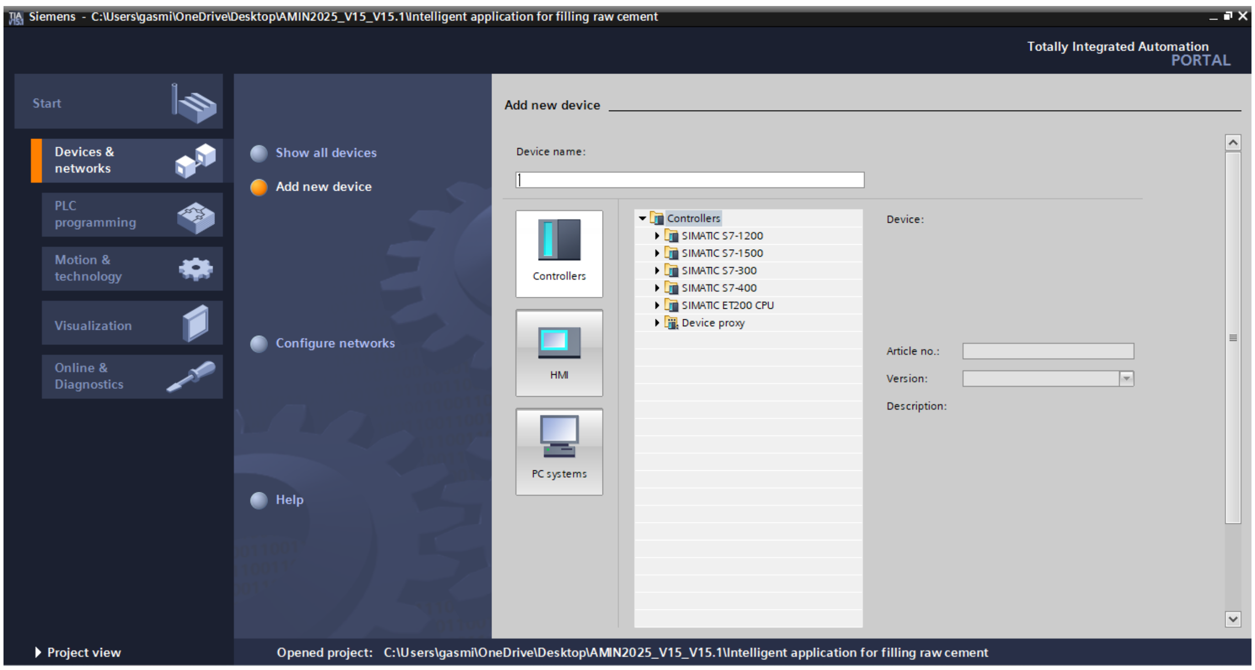


Figure III.16: hardware configuration and setup

Now we add the programmable logic controller (PLC) and the Communications module using the catalog. To add a display or other PLC, refer to the "Add Device" command in the Project Browser. When you select an item for inclusion in the project, a description appears in the Information tab. The figure below shows The Device Configuration.

- ✓ **Location 1: CPU 1214 AC/DC/RLY**
- ✓ **Location 101 : Communications module CM 1241 (RS422/485)**

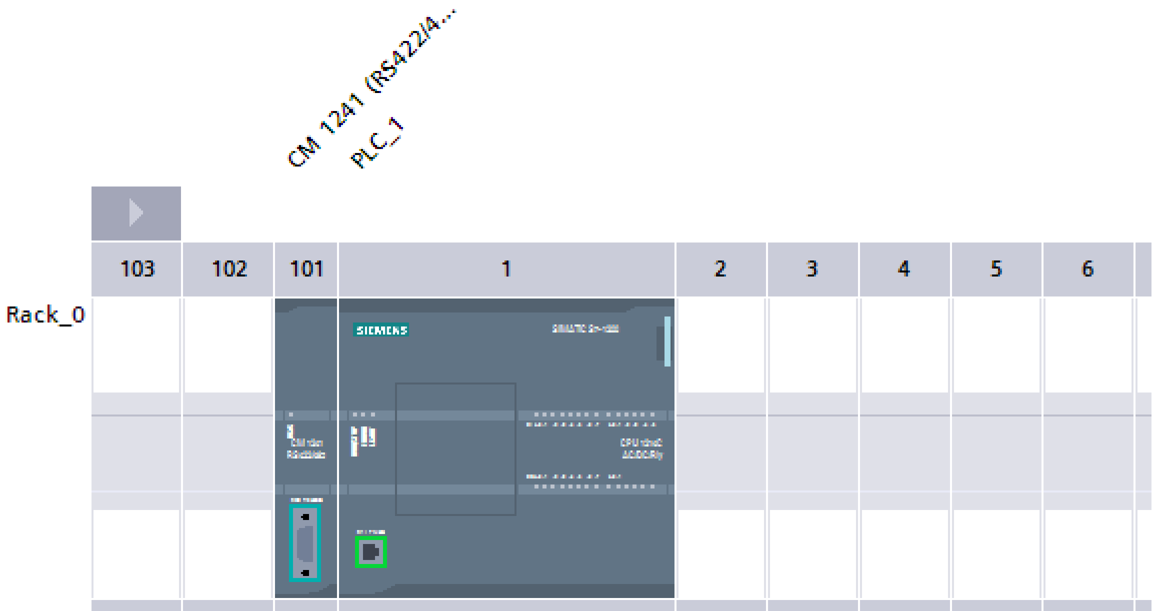


Figure III.17: Device Configuration

III.6.2. PLC Variables

In any automation program, it is essential to define a comprehensive list of variables that will be utilized during development. For this purpose, a **PLC variable table** is created to declare and manage system variables. Each variable includes a **symbolic name**—a descriptive identifier assigned by the programmer—and an **absolute address** that corresponds to its physical memory location (e.g., I0.1, Q4.0, M10.3). These symbolic names greatly enhance the readability, maintainability, and traceability of the program.

Additionally, the variable table defines the **data type** (such as BOOL or INT), and may include **comments** to further clarify the role of each variable. TIA Portal offers the flexibility to display symbolic names, absolute addresses, or both simultaneously, facilitating clearer program structure and easier debugging. The following figures illustrate the variable table used in our cement loading application.

1	Name	Path	Data Type	Logical Add	Comment
2	RAISE	Default tag	Bool	%I0.1	Command to raise the arm.
3	DOWN	Default tag	Bool	%I0.2	Command to lower the arm.
4	Open the valve	Default tag	Bool	%I0.3	Command to open the filling valve.
5	close the valve	Default tag	Bool	%I0.4	Command to close the filling valve.
6	RESPONSE VALVE	Default tag	Bool	%I0.5	Feedback signal confirming valve response.
7	VALVE	Default tag	Bool	%Q0.3	Current status of the valve (open/closed).
8	ARM RAISE	Default tag	Bool	%Q0.1	Output signal to actuate the arm raise.
9	ARM DOWN	Default tag	Bool	%Q0.2	Output signal to actuate the arm lower.
10	POWER CHECK	Default tag	Bool	%Q0.7	Power presence verification signal.
11	FirstScan	Default tag	Bool	%M1.0	Active during the first PLC scan cycle (used for initialization).
12	Clock_5Hz	Default tag	Bool	%M1000.1	5Hz clock pulse used for timing or watchdog mechanism.
13	SELECTEUR	Default tag	Bool	%M30.0	Selector for operation mode
14	VALVE_STATE	Default tag	Bool	%M1.4	Monitoring the current state of the valve.
15	authorization local	Default tag	Bool	%M2.0	Authorization signal (e.g., after data validation).
16	XN1	Default tag	Bool	%M2.1	
17	INIT	Default tag	Bool	%M2.2	System initialization signal.
18	FLT	Default tag	Bool	%M2.4	Fault indication signal.
19	BY_PASS	Default tag	Bool	%M11.0	Bypass mode for manual override or maintenance.
20	POWER ALARM	Default tag	Bool	%M2.5	Alarm signal for power supply failure or issue.

Figure III.18: The PLC Tag Table

III.6.3. Program Creation:

In programming the intelligent cement loading project, we used OB, FC, and Data Block (DB) elements to structure the application within TIA Portal. The OB1 (Organization Block) was adopted as the main entry point for executing the program, as it is cyclically processed by the PLC's CPU and is used to call subordinate functions.

The FC (Function) blocks—displayed in green within the TIA Portal environment—were utilized to divide the program into specific and well-defined tasks, such as reading the weight from the scale, controlling the motor and valve, and processing start and stop commands.

To enable consistent data exchange between these functions and with the external application, we also created and used Data Block (DB). These blocks serve as structured memory areas to store and share relevant operational variables such as control bits, sensor readings, and setpoints.

This structured organization improved the maintainability of the program, clarified the control logic, and facilitated the reuse of the same functions in different parts of the project.

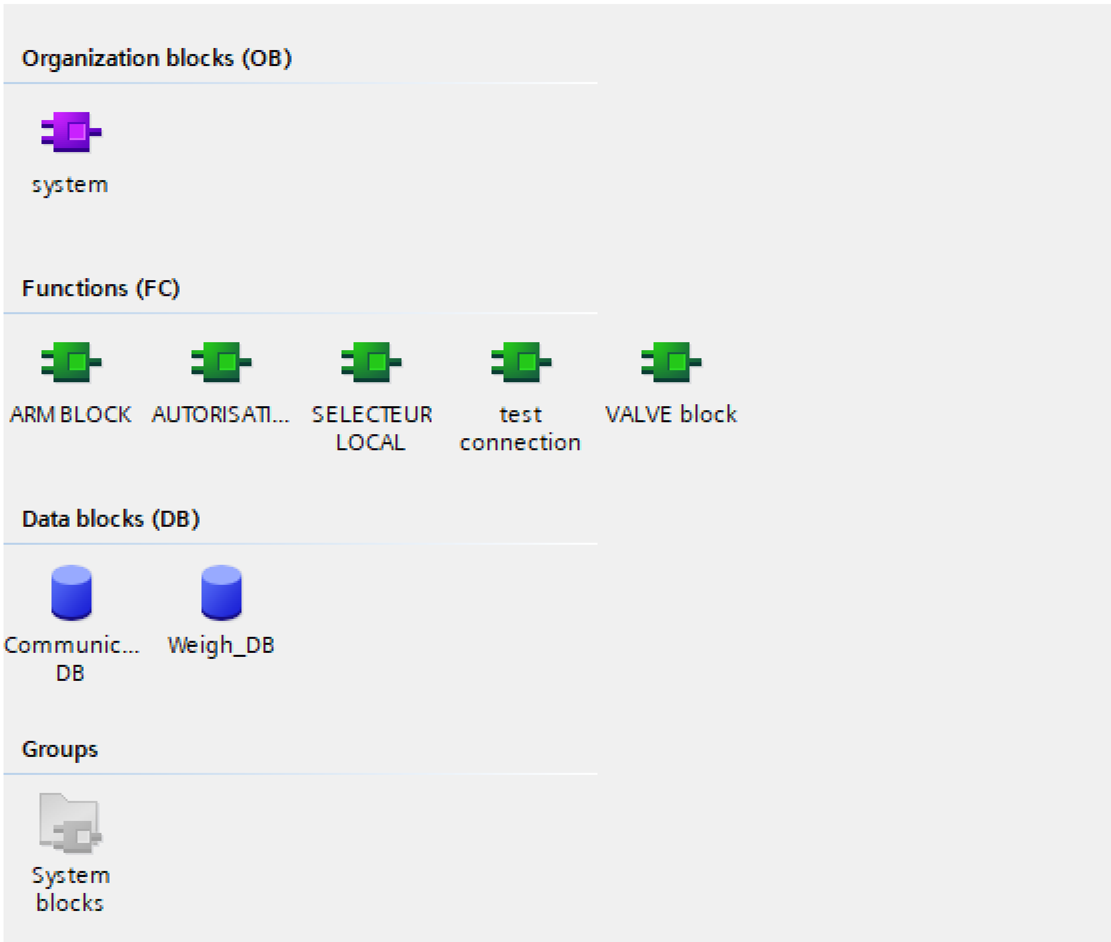


Figure III.19 : Program Blocks

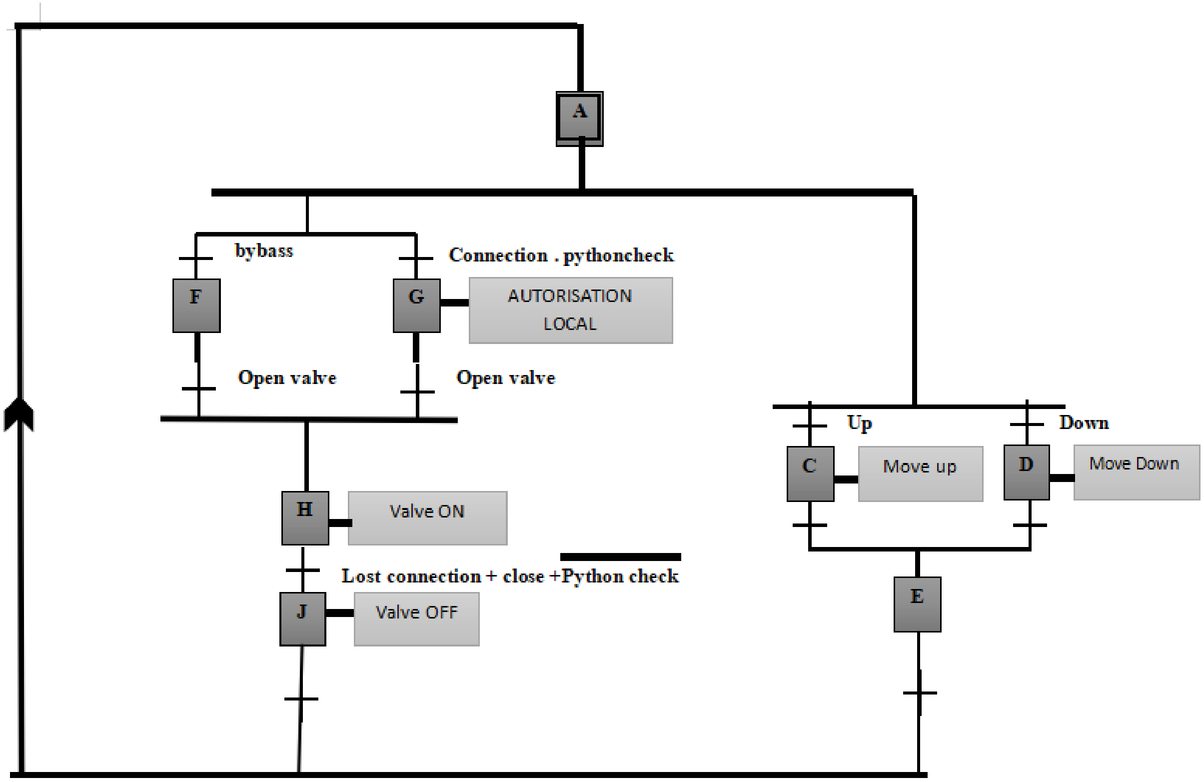


Figure III.20: General grafcet chart

III.6.4 Data Blocks (DBs)

In the context of this project, we created two main data blocks (DBs) within the TIA Gateway environment to manage the exchange of data and operational signals between the programmable logic controller (PLC) and the Python smart application:

III.6.4.1. Communication Data Block:

This data block serves as an interface for real-time data exchange between the PLC and the smart application. It contains variables representing control commands, system status, and alarm indicators, enabling the Python application to read the system state and send operational commands to the PLC. This structure ensures synchronization between the industrial control logic and the software interface.

III.6.4.2 Weigh Data Block:

The Weigh DB is a dedicated data block to receive and store real-time values read from the industrial weighing system used for truck measurements. This block acts as an interface between the PLC (Programmable Logic Controller) and the measuring devices, enabling the transfer of raw weight data to the smart application.

III.6.3 Function Blocks (FC):

As part of the "Smart Cement Loading Application" project, a set of function blocks were developed, each responsible for executing a specific task. This modular organization enhances the clarity of process sequences and simplifies future updates and maintenance of the program.

These blocks were written using a standardized programming approach, enabling their reuse in similar projects. Moreover, variables can be easily linked through the main organizational block OB1, ensuring efficient integration within the system's architecture.

III.6.3.1 ARM (FC):

Responsible for controlling the movement of the filling arm using a bidirectional motor.

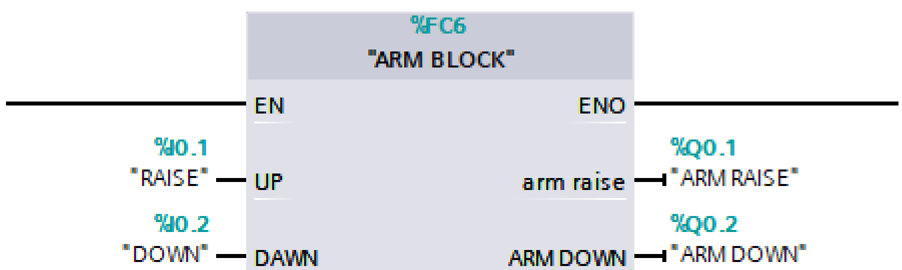


Figure III.21: ARM BLOCK (FC) Variable Declaration

III.6.3.2 VALVE (FC):

Manages the opening and closing of the filling valve according to specific operational conditions.

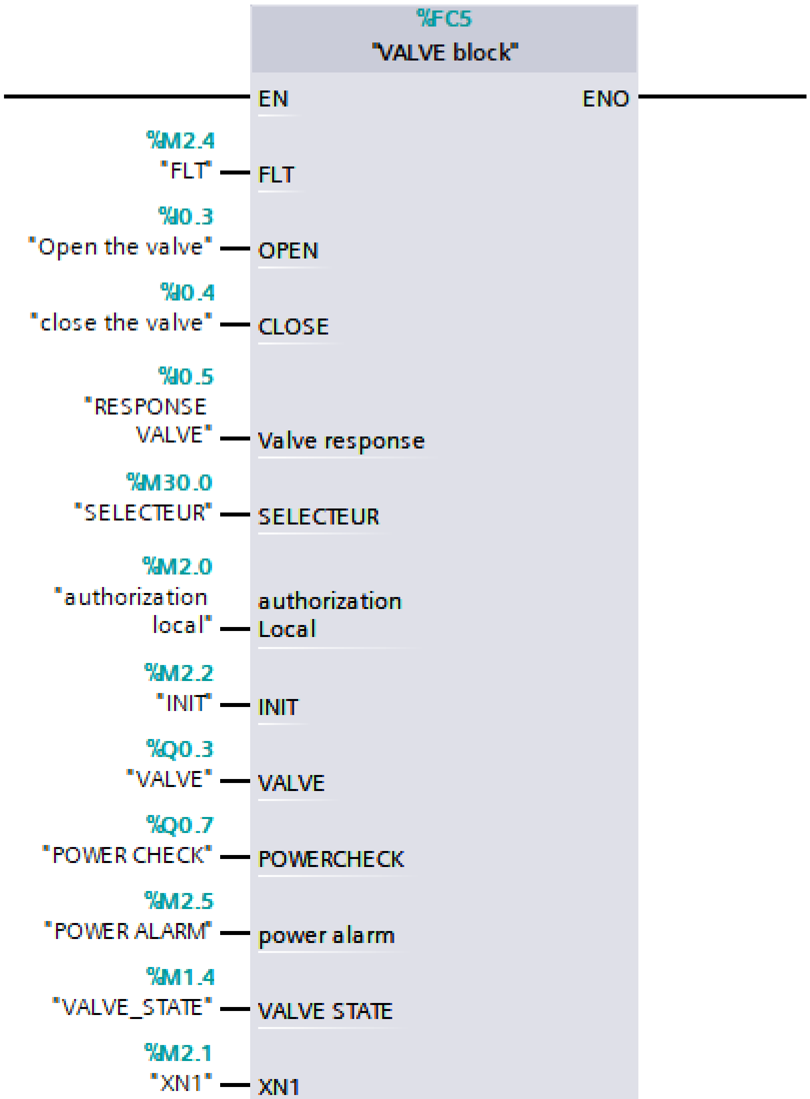


Figure III.22: VALVE BLOCK (FC)

III.6.3.3 AUTORISATION (FC):

Ensures all required conditions are met before initiating the filling process (such as truck validation, stable weight, etc.).

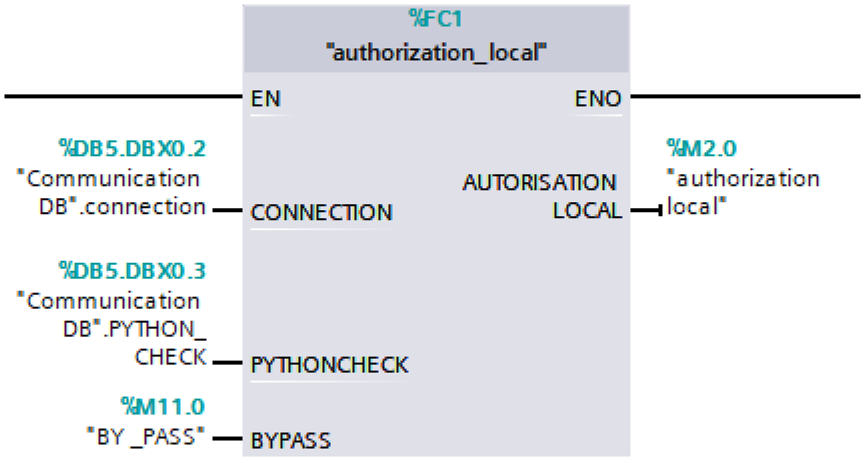


Figure III.23: AUTORIZATION LOCAL BLOCK (FC)

III.6.3.4 Test Connection (FC):

Used to verify the connectivity with external applications ‘’Python’’ ,ensuring reliable communication between systems.

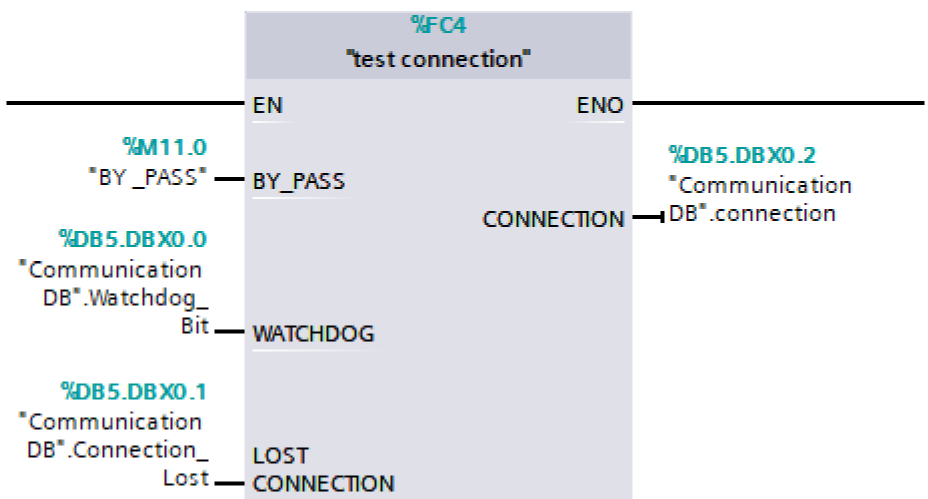


Figure III.24: TEST CONNECTION BLOCK (FC)

III.6.4 Organization Block (OB) :

- ✓ **System (OB1) :**

OB1 is the main block in the programmable logic controller (PLC) program, executed periodically by the CPU. Its primary function is to call the function blocks (FCs) used in the system, including those for controlling the arm, valve, authorization verification, and others.

This block also includes the blocks associated with the industrial scale, which are illustrated in the following images.

The PORT_CFG_BLOCK is used to store port settings, such as:

- Network connection settings (e.g., Profinet or Profibus).
- Serial port settings (e.g., RS232 or RS485).
- Or even I/O port settings on peripheral devices or I/O modules.

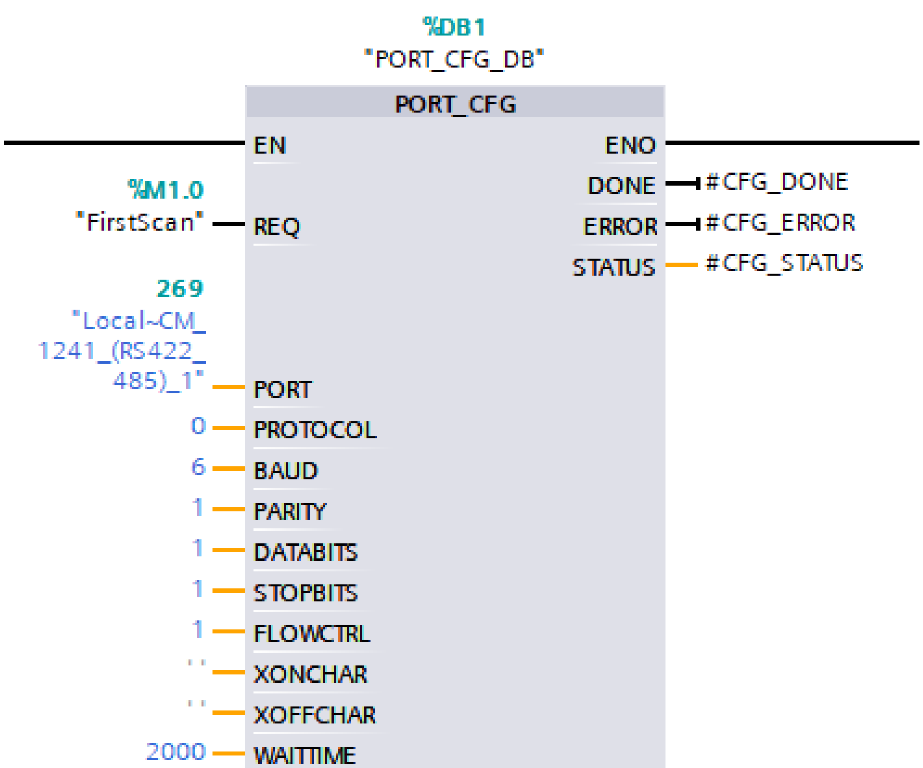


Figure III.25 : PORT_CFG_BLOCK

The RCV_PTP_DB block is used to store data received over point-to-point communication, and is often used with serial modules such as the CM 1241. RCV stands for "Receive" and PTP stands for "Point-to-Point."

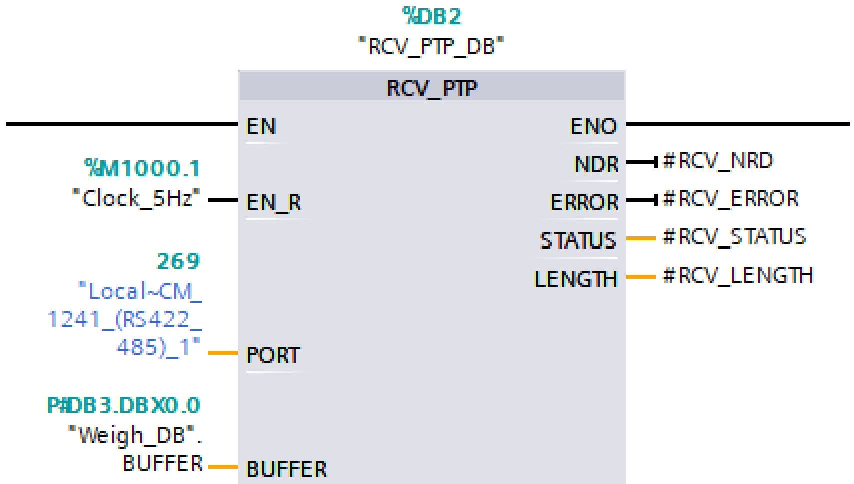


Figure III.26: RCV_PTP BLOCK

The MID block is a Standard Function Block in TIA Portal that is used to read a portion of a string.

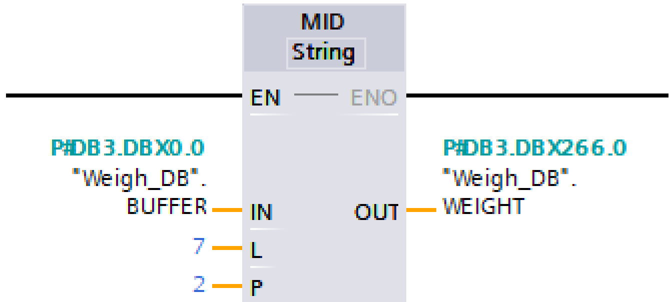


Figure III.27: MID BLOCK (FC)

The STRG_VAL BLOCK is a String block that is typically used to read, interpret, or convert input from a string type to a numeric or logical type.

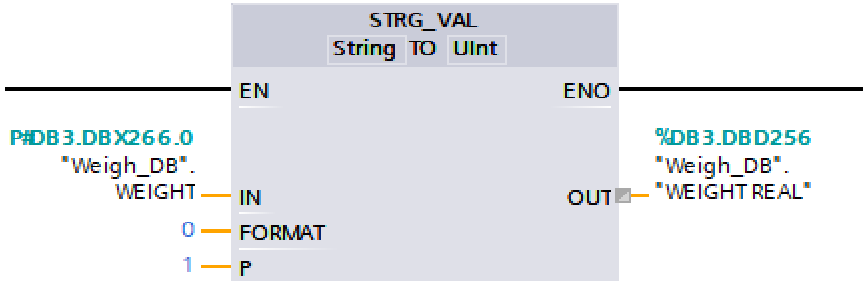


Figure III.28: STRG_VAL BLOCK

The DIV block stands for Division, and is used in the TIA Portal environment within Math Instructions.

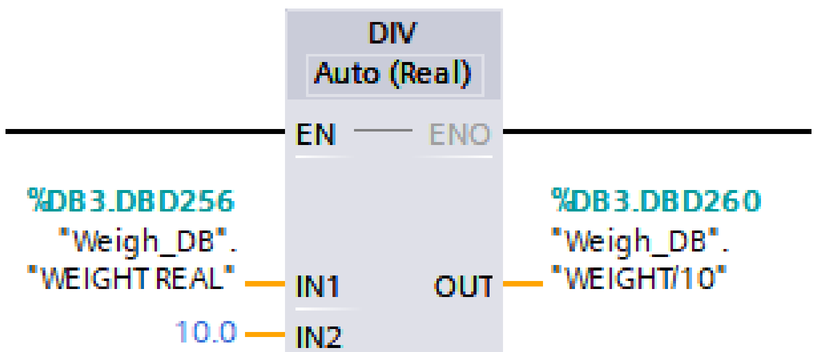


Figure III.29: DIV BLOCK

III.7. Communication Methods

This section presents the communication methods adopted in the smart cement packaging application, which ensure instant and reliable data exchange between the various components of the system.

III.7.1 Communication between the PLC and the Python application:

Data is exchanged between the programmable logic controller (PLC) and the smart application using data block(**Communication DB**) structured within the TIA Portal environment. **The Snap7 library** in the Python application facilitates direct communication with the PLC via **the S7 protocol,** enabling reading and writing of values within this block. This method enables real-time data synchronization between the industrial system and the smart application, enhancing reliability and accuracy during packaging operations.

III.7.2 Communication between Python and a MySQL database:

The application relies on the **MySQL. Connector library** in Python to connect to the database. This connection is used to retrieve truck information, validate data, and store session data for tracking and performance analysis.

III.7.3 Communication between Python and the camera:

The application uses the **cv2 library** (**OpenCV in Python**) to access a camera—connected to the computer via USB or Wi-Fi module—for capturing and processing the truck's license plate in real time.

III.7.4 Communication between PLC and weighing scale indicator:

Communication between the PLC and the weighing scale is via **the Profibus DP protocol** using a **CM 1241 module** connected to an RS-485 system. Using this protocol, the PLC receives weight data from the weighing scale.

III.8 Electrical cabinet

At the Biskria cement plant, specifically in the raw cement filling area, large silos are used to store cement ready for loading. Each silo is equipped with its own system, consisting of an electric motor that controls a mechanical arm which adjusts up and down according to the height of the truck, as well as an electric valve that regulates the flow of cement. Both components are controlled from the control station via the PLC unit.

Figure III.30 : Electrical cabinet before modification

Figure III.31 : Electrical cabinet after modification

The diagram below demonstrates the wiring and connection layout of the station's electrical cabinet.

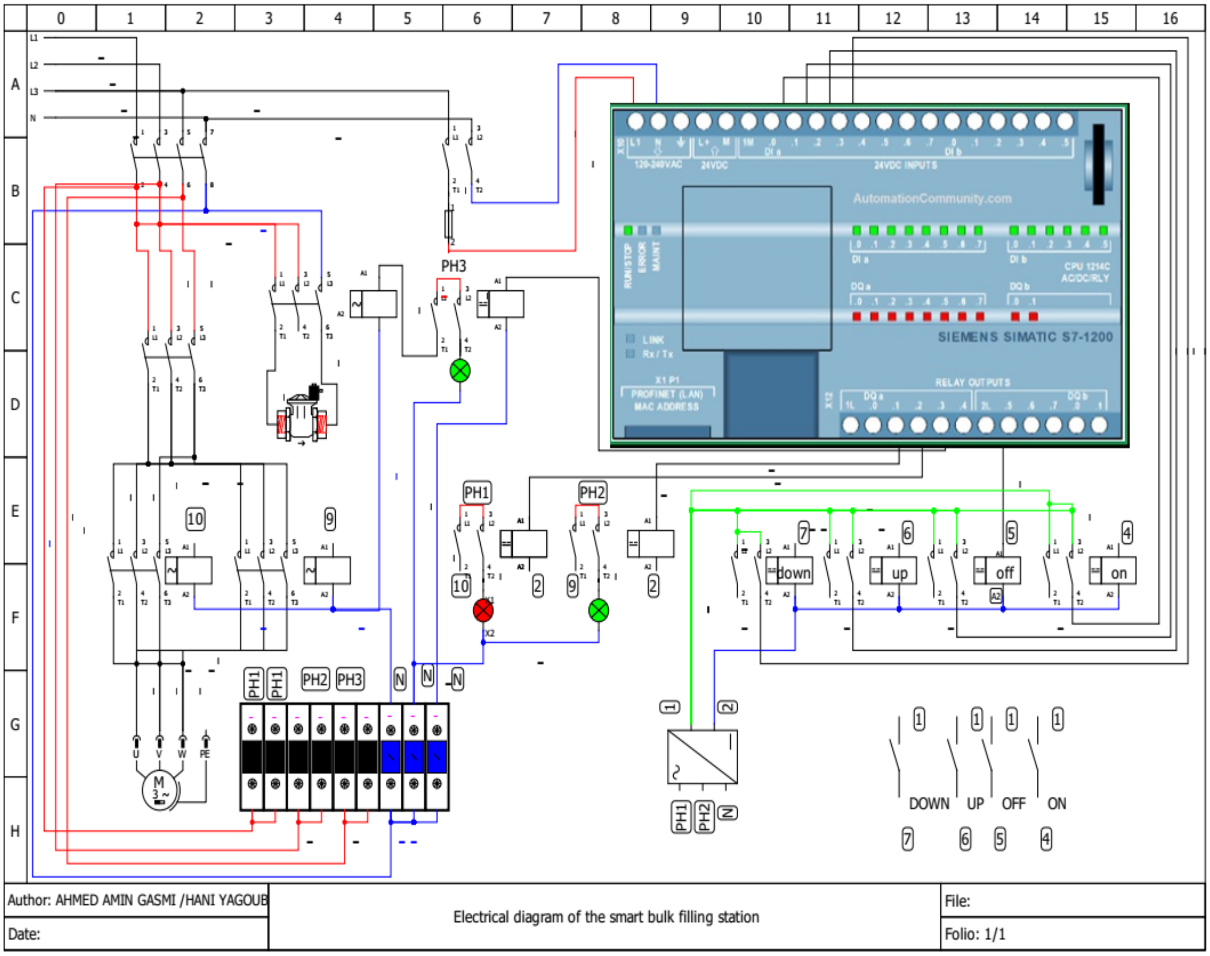


Figure III.32 : Electrical diagram of the smart bulk filling station

III.9 System Workflow

The system operates through a series of well-defined steps that ensure accurate identification, control, and data tracking throughout the filling process. These steps include:

III.9.1 Initialization:

At the beginning of the process, the truck designated for loading drives onto the industrial scale located at the loading station, in preparation for initiating the weight measurement procedures and carrying out the loading operation.

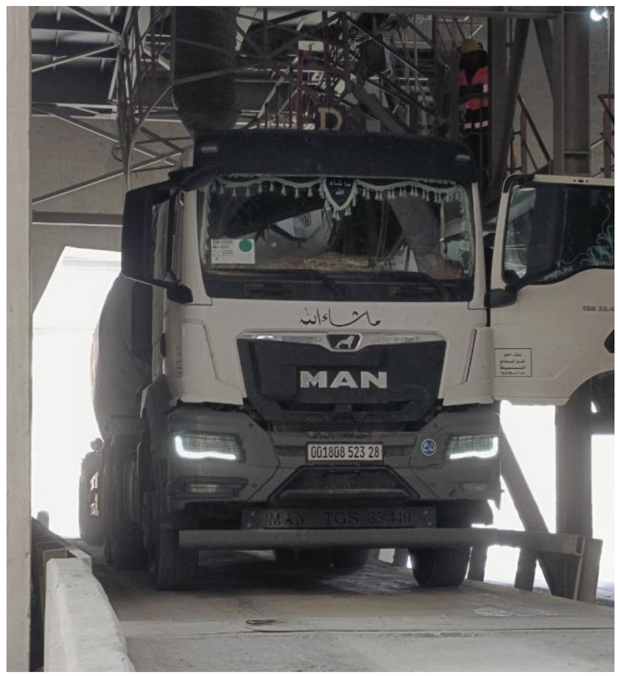


Figure III.33: Truck climbs onto scale

After the truck ascends onto the weighbridge, its weight is measured automatically with high precision. When the measured weight reaches a predefined threshold in the program executed on the PLC — representing the minimum weight of an empty truck (10 tons) — and the truck remains steadily positioned on the weighbridge for no less than 30 seconds without any significant weight fluctuation, the PLC changes the state of the output designated for the (Truck Check) from 0 to 1. This indicates the confirmed presence and stability of the truck on the weighbridge

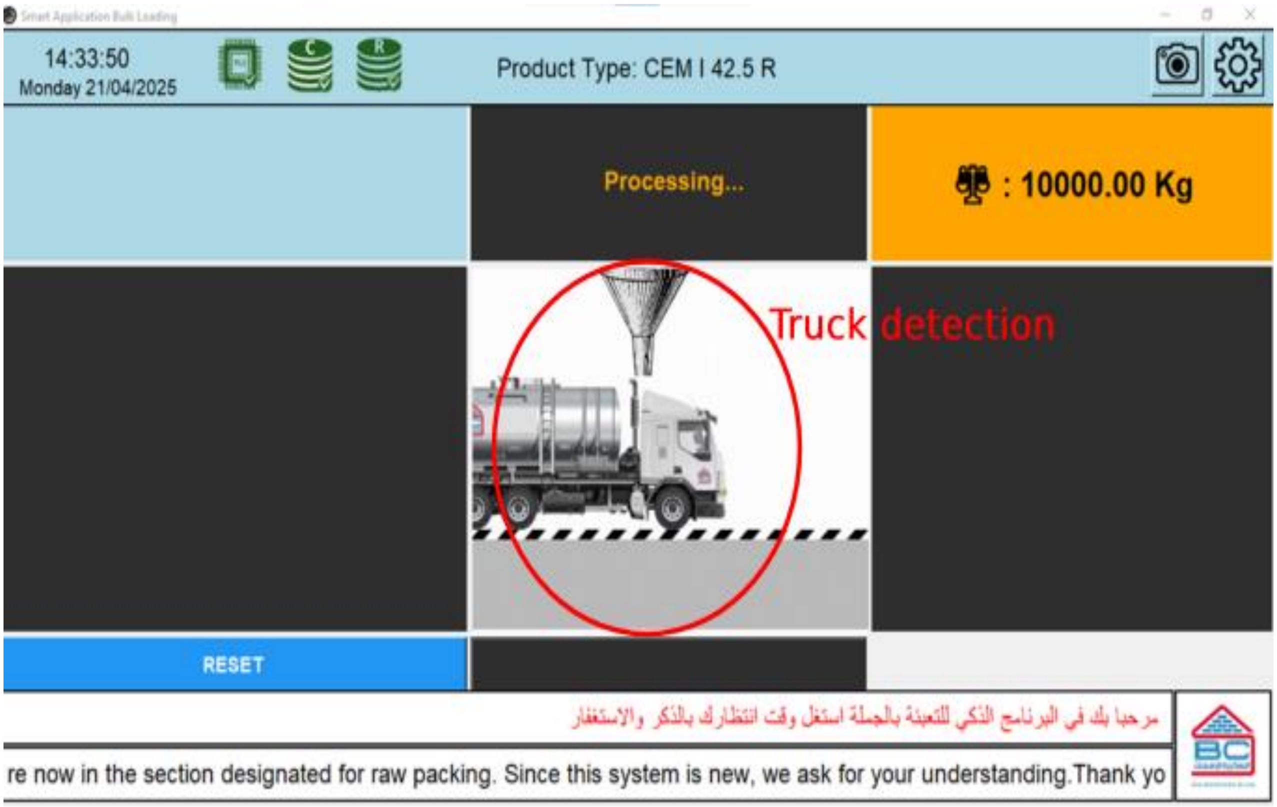


Figure III.34: truck detection

The intelligent application is programmed to continuously monitor the value of the Truck Check output from the Data Block within the PLC.

Upon detecting a change in this output from 0 to 1—indicating that the truck is stably positioned on the weighbridge—the application automatically authorizes the system to begin operation without manual intervention, thereby initiating the loading process.

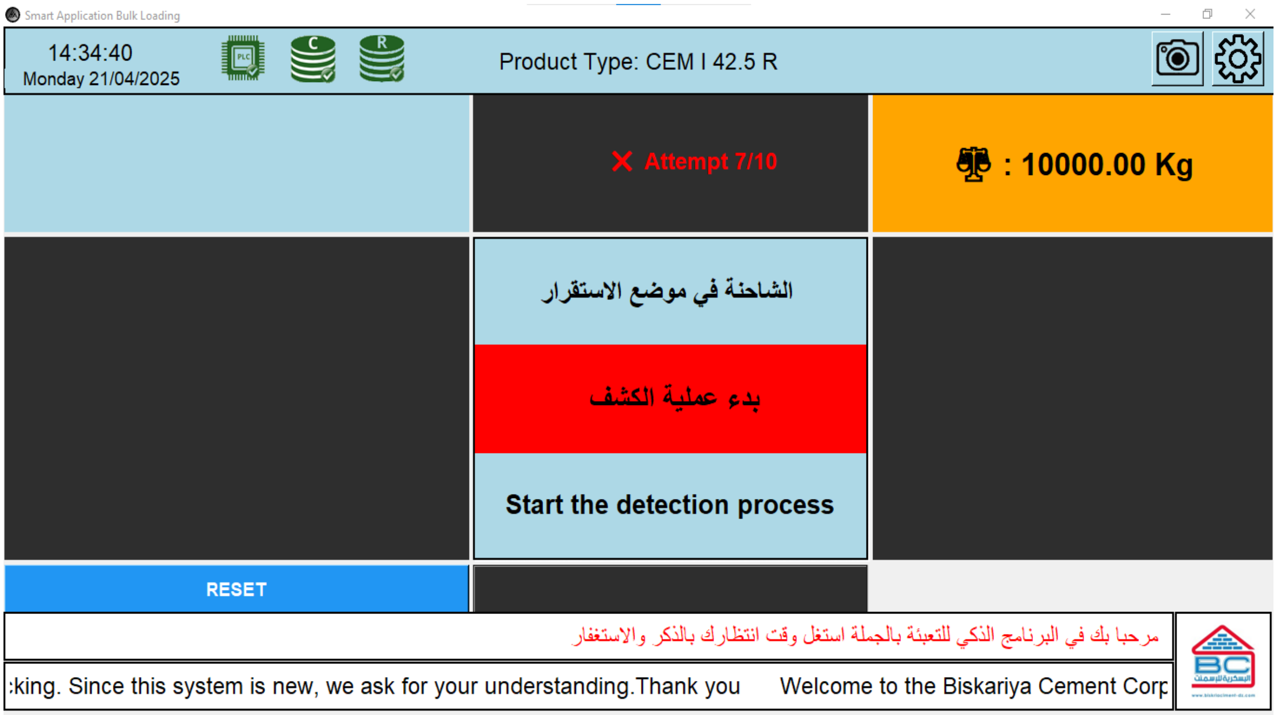


Figure III.35: System startup

III.9.2 Verification:

Before starting the upload process, the system automatically activates the camera to capture an image of the front section of the truck. The **OpenCV** framework is then utilized to apply Optical Character Recognition (OCR) technology, enabling the automatic extraction of the truck's license plate number.

Figure III.36: Image capture and processing

After capturing the image using the camera, the application processes the image with the **OpenCV** library to apply **Optical Character Recognition** (OCR) techniques for extracting the license plate number. The extracted number is then compared with the reference database, which contains the license plate numbers of registered trucks. If a match is found, the system proceeds to the next step, where it automatically imports the truck's information based on the entry where the matching number was found.

Subsequently, the cement type requested by the truck is compared with the cement type preconfigured in the system settings for the station. It is important to note that the cement type is set in advance according to the station where the system is installed, ensuring that the process aligns with the specified operational standards.

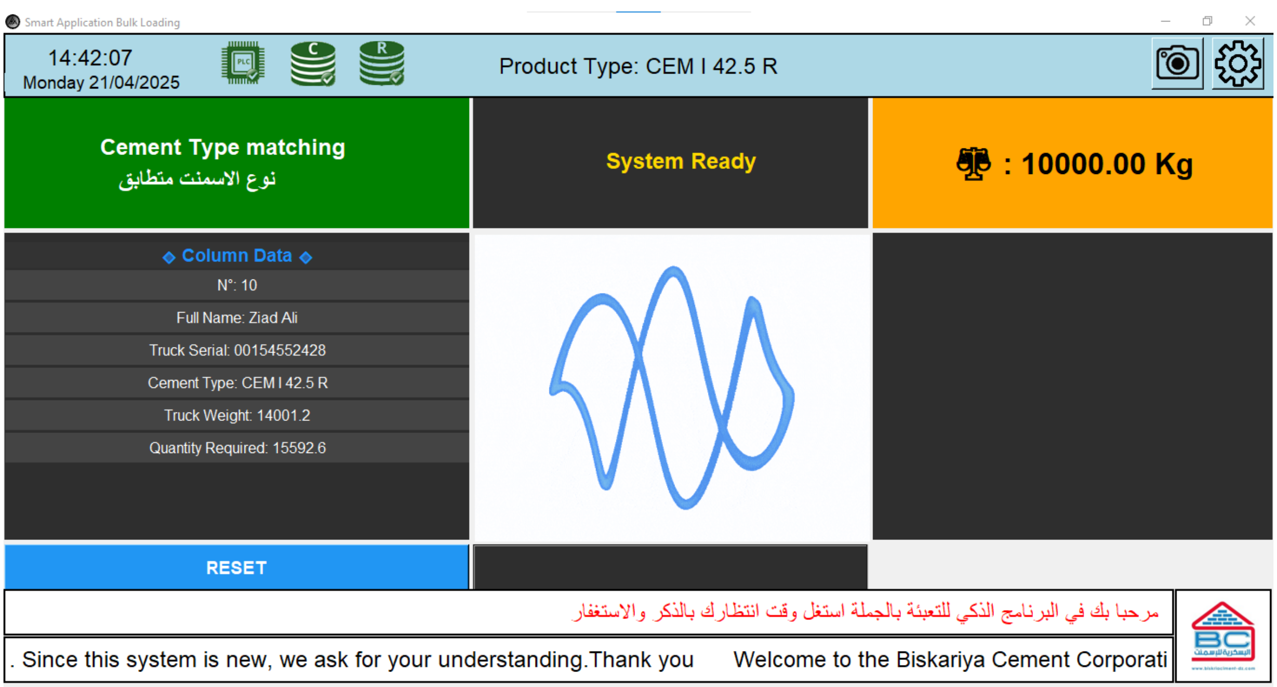


Figure III.37: Matching information after comparison

III.9.3 Filling Process:

After successful verification of the license plate number and data comparison, the application sends a start command to the Programmable Logic Controller (PLC) by modifying the **Python Check** bit in the **Communication (DB)** using

Python. The bit value is changed from 0 to 1, indicating the automatic initiation of the loading process.

Communication DB (snapshot created: 4/14/2025 10:54:10 AM)

	Name	Data type	Offset	Start value	Monitor value	Retain	Accessible
1	▼ Static					☐	☐
2	Watchdog_Bit	Bool	0.0	false	FALSE	☐	☑
3	Connection_Lost	Bool	0.1	true	TRUE	☐	☑
4	connection	Bool	0.2	FALSE	TRUE	☐	☑
5	PYTHON_CHECK	Bool	0.3	FALSE	TRUE	☐	☑
6	VANE STATU	Bool	0.4	false	FALSE	☐	☑
7	TRUCK CHECK	Bool	0.5	false	TRUE	☐	☑
8	WH	Real	2.0	0.0	0.0	☐	☑

Figure III.38: the Python Check bit

After sending the start command, the PLC activates the valve responsible for the cement flow, then start the loading process into the truck.

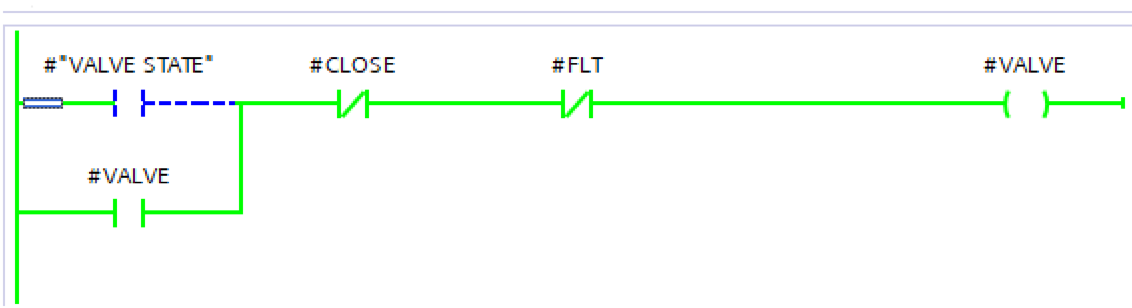


Figure III.39: Valve network (valve open)

After the loading process begins, the system uses the **Actual Weight bit** from the **Weigh (DB)** to monitor the filled weight. This weight is compared with the required cement quantity for loading, which is retrieved from the system's reference database to ensure that the filled weight aligns with the specified amount.

1	▼ Static					☐	☐
2	BUFFER	String	0.0	''	''	☐	☑
3	WEIGHT REAL	Real	256.0	0.0	0.0	☐	☑
4	WEIGHT/10	Real	260.0	0.0	0.0	☐	☑
5	BUFFER2	Int	264.0	0	0	☐	☑
6	WEIGHT	String	266.0	''	''	☐	☑

Figure III.40: Real Weight bit

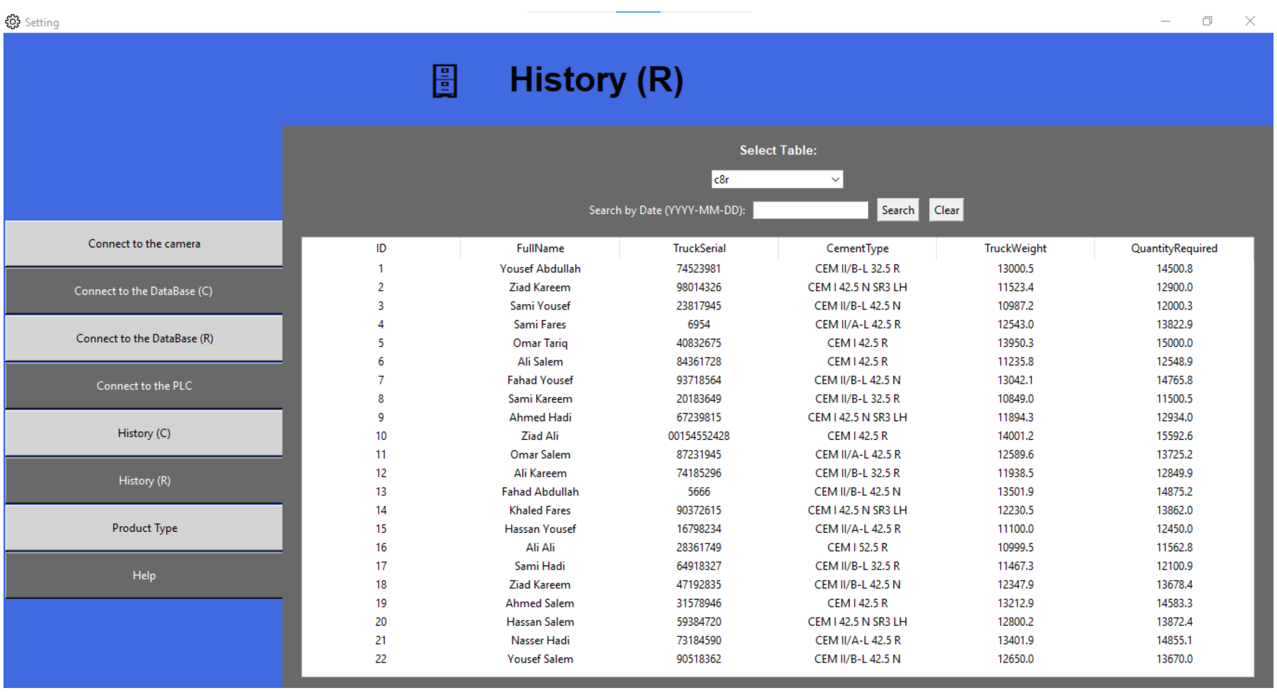
After reaching the desired weight, the system halts the loading process by changing the value of the Python Check bit in the data block to 0. Additionally, the process can also be manually stopped via the user interface, allowing the operator to intervene if necessary.

■	connection	Bool	0.2	FALSE	TRUE
■	PYTHON_CHECK	Bool	0.3	FALSE	FALSE
■	VANE STATU	Bool	0.4	false	FALSE

Figure III.41: Python check bit

III.9.4 Data storage:

All data related to each session, such as cement type, truck ID, filling time, and weight, is automatically stored in the historical database. This is intended to enable future retrieval of this data and provide the capability for process tracking and analysis, ensuring accuracy and efficiency in managing the loading operations.



ID	FullName	TruckSerial	CementType	TruckWeight	QuantityRequired
1	Yousef Abdullah	74523981	CEM II/B-L 32.5 R	13000.5	14500.8
2	Ziad Kareem	98014326	CEM I 42.5 N SR3 LH	11523.4	12900.0
3	Sami Yousef	23817945	CEM II/B-L 42.5 N	10987.2	12000.3
4	Sami Fares	6954	CEM II/A-L 42.5 R	12543.0	13822.9
5	Omar Tariq	40832675	CEM I 42.5 R	13950.3	15000.0
6	Ali Salem	84361728	CEM I 42.5 R	11235.8	12548.9
7	Fahad Yousef	93718564	CEM II/B-L 42.5 N	13042.1	14765.8
8	Sami Kareem	20183649	CEM II/B-L 32.5 R	10849.0	11500.5
9	Ahmed Hadi	67239815	CEM I 42.5 N SR3 LH	11894.3	12934.0
10	Ziad Ali	00154552428	CEM I 42.5 R	14001.2	15592.6
11	Omar Salem	87231945	CEM II/A-L 42.5 R	12589.6	13725.2
12	Ali Kareem	74185296	CEM II/B-L 32.5 R	11938.5	12849.9
13	Fahad Abdullah	5666	CEM II/B-L 42.5 N	13501.9	14875.2
14	Khaled Fares	90372615	CEM I 42.5 N SR3 LH	12230.5	13862.0
15	Hassan Yousef	16798234	CEM II/A-L 42.5 R	11100.0	12450.0
16	Ali Ali	28361749	CEM I 52.5 R	10999.5	11562.8
17	Sami Hadi	64918327	CEM II/B-L 32.5 R	11467.3	12100.9
18	Ziad Kareem	47192835	CEM II/B-L 42.5 N	12347.9	13678.4
19	Ahmed Salem	31578946	CEM I 42.5 R	13212.9	14583.3
20	Hassan Salem	59384720	CEM I 42.5 N SR3 LH	12800.2	13872.4
21	Nasser Hadi	73184590	CEM II/A-L 42.5 R	13401.9	14855.1
22	Yousef Salem	90518362	CEM II/B-L 42.5 N	12650.0	13670.0

Figure III.42 : Reference Database

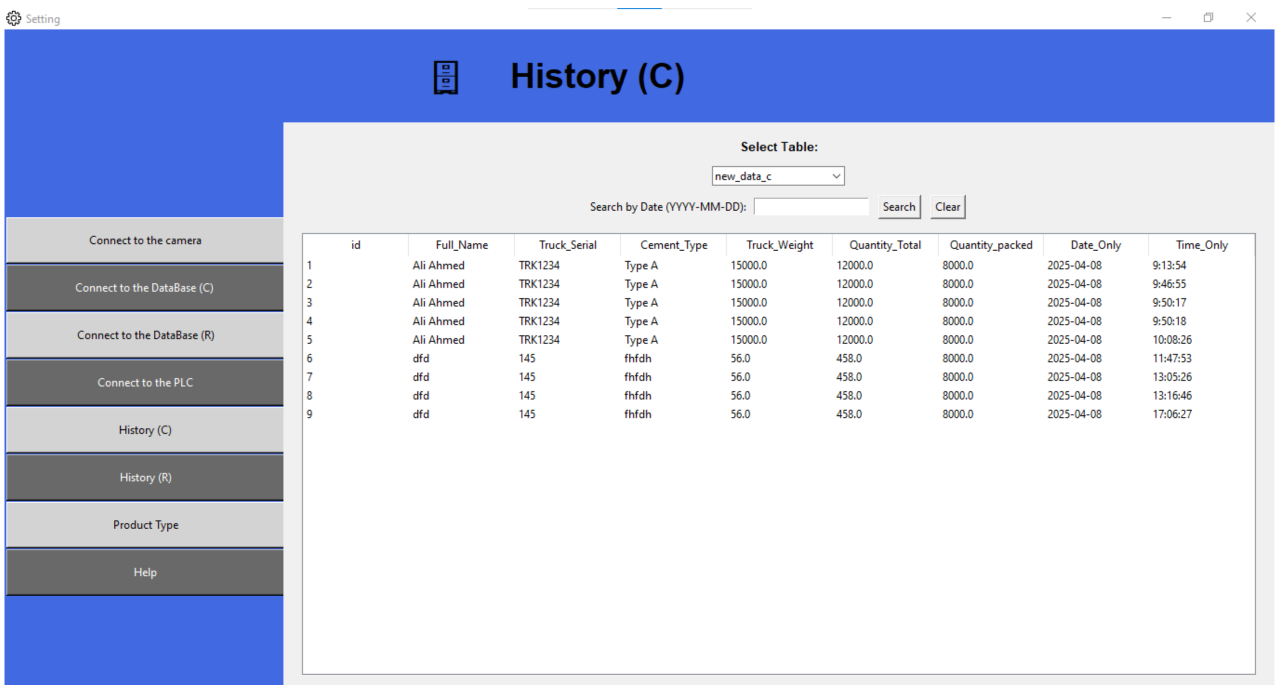


Figure III.43: Historical Database

Once the loading process is complete, the application automatically returns to idle mode until another truck arrives or it is manually activated by the operator.

The station can also be operated in manual mode in emergency situations, such as a malfunction of the intelligent application or for maintenance purposes. In these cases, the operation of the station is controlled through the DCS system by changing the **BY_PASS** bit state in the PLC program, allowing the operator to safely and effectively operate the system manually.

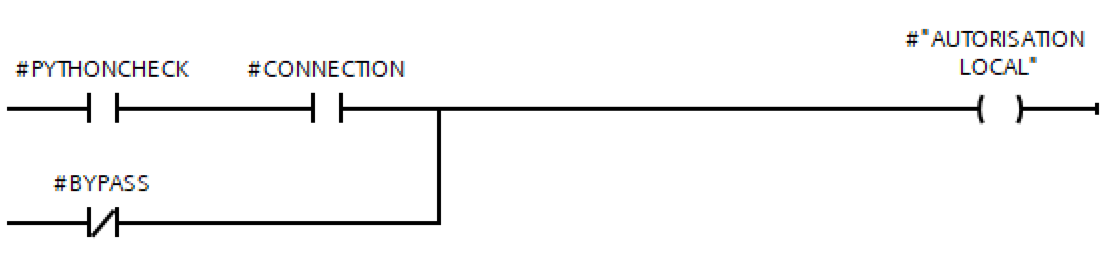


Figure III.44: Network of the system with **BY_PASS**

III.10 User Interface (UI):

The **Tkinter** library in Python was used to develop the cement filling application's user interface with the goal of making it simple for managers and workers to use. Along with basic control and navigation buttons, it shows important data including the current weight, vehicle license plate number, and type of cement.

This interface helps monitor the filling process smoothly and shows alerts in case of errors or connection loss, making the system practical and efficient in an industrial environment.

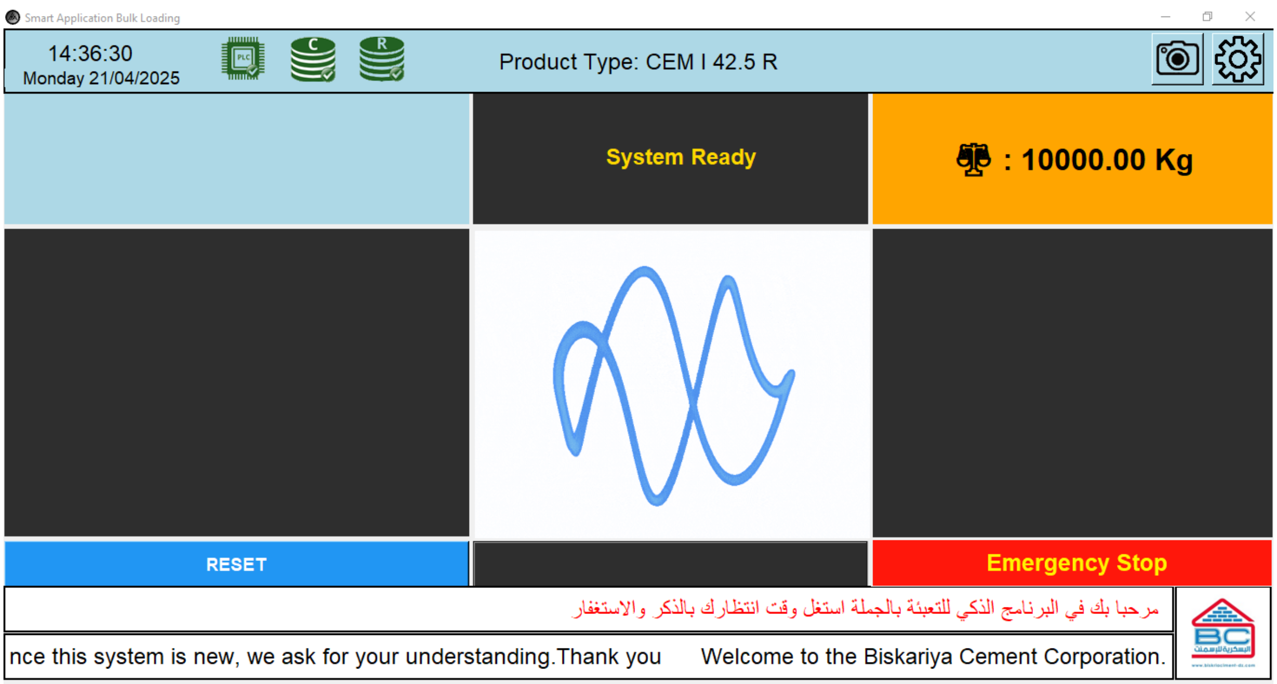


Figure III.45: Main User Interface

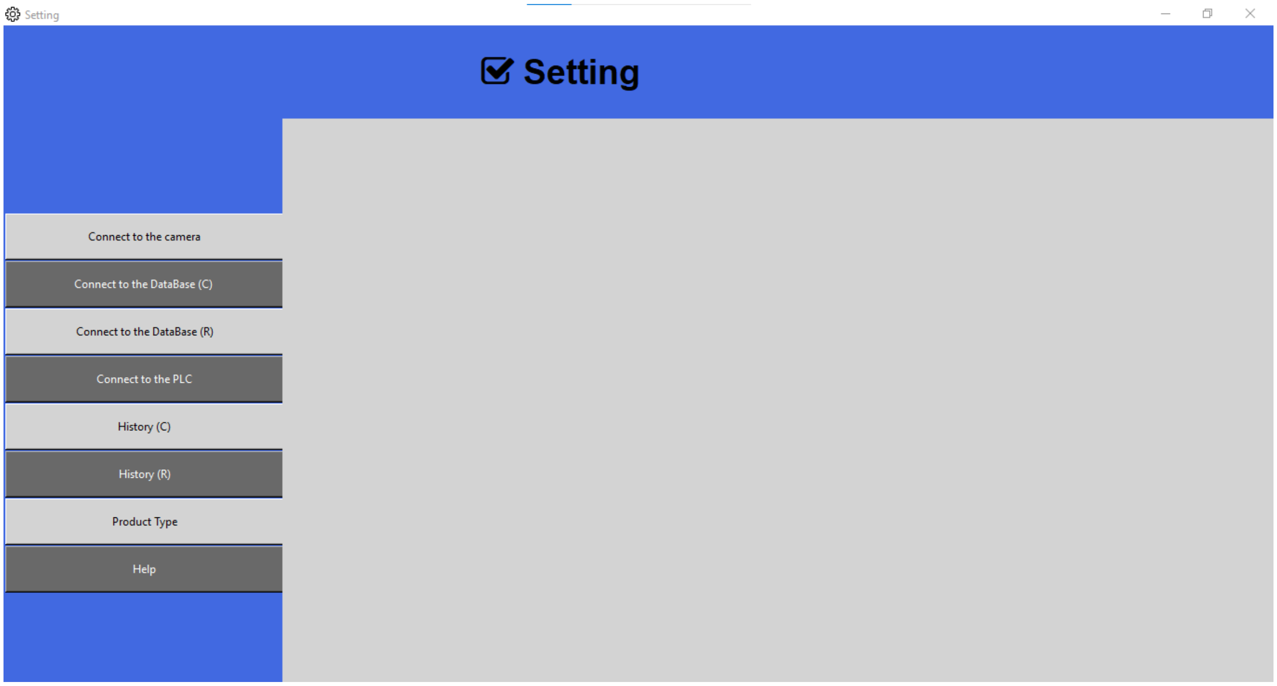


Figure III.46: Settings interface

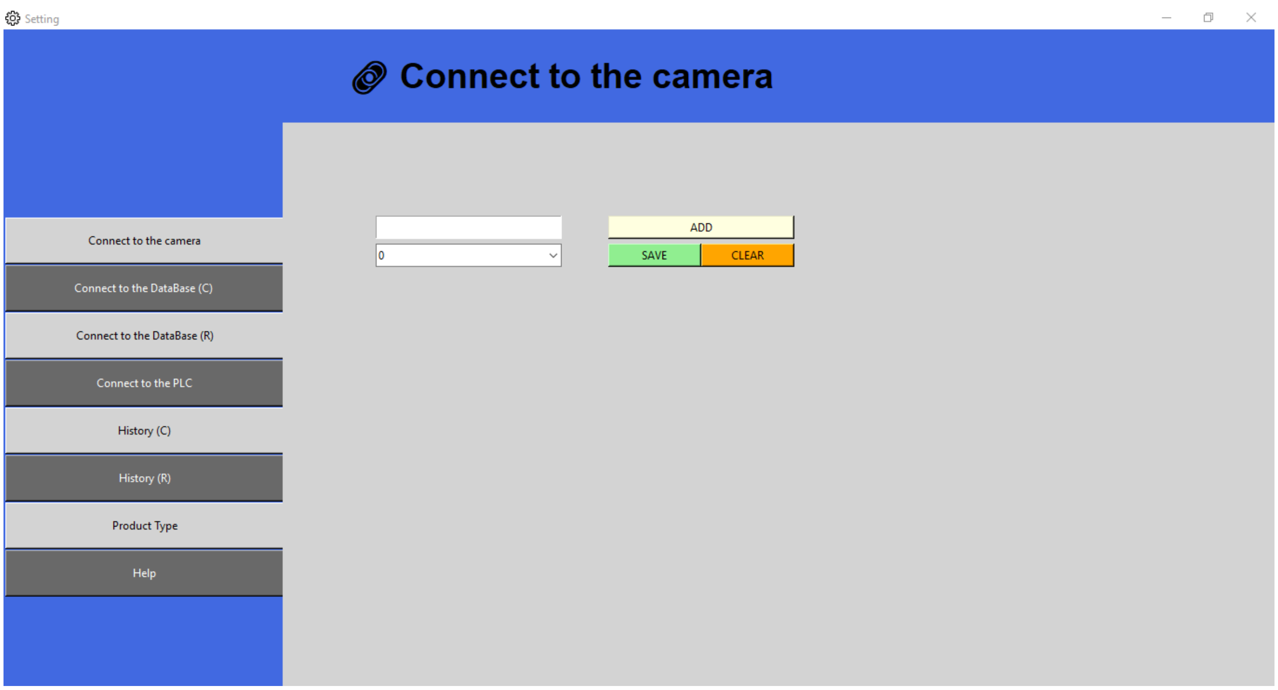


Figure III.47: Camera interface

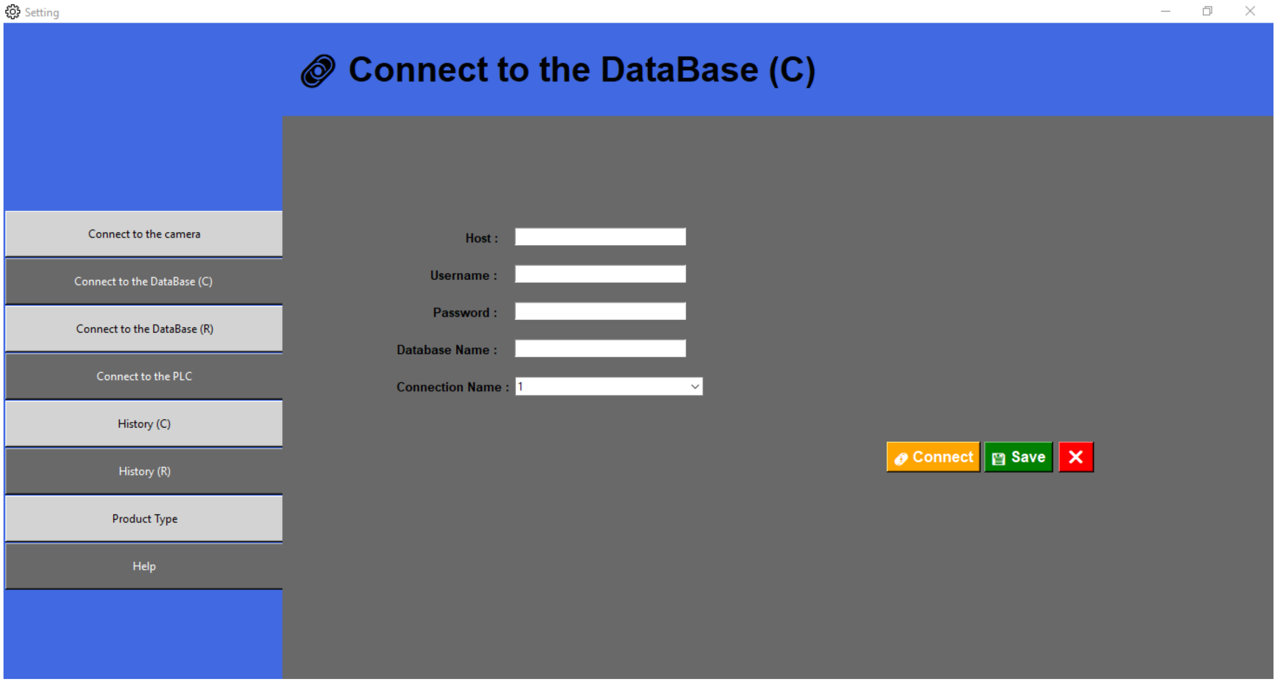


Figure III.48: User interface for connecting to the historical database

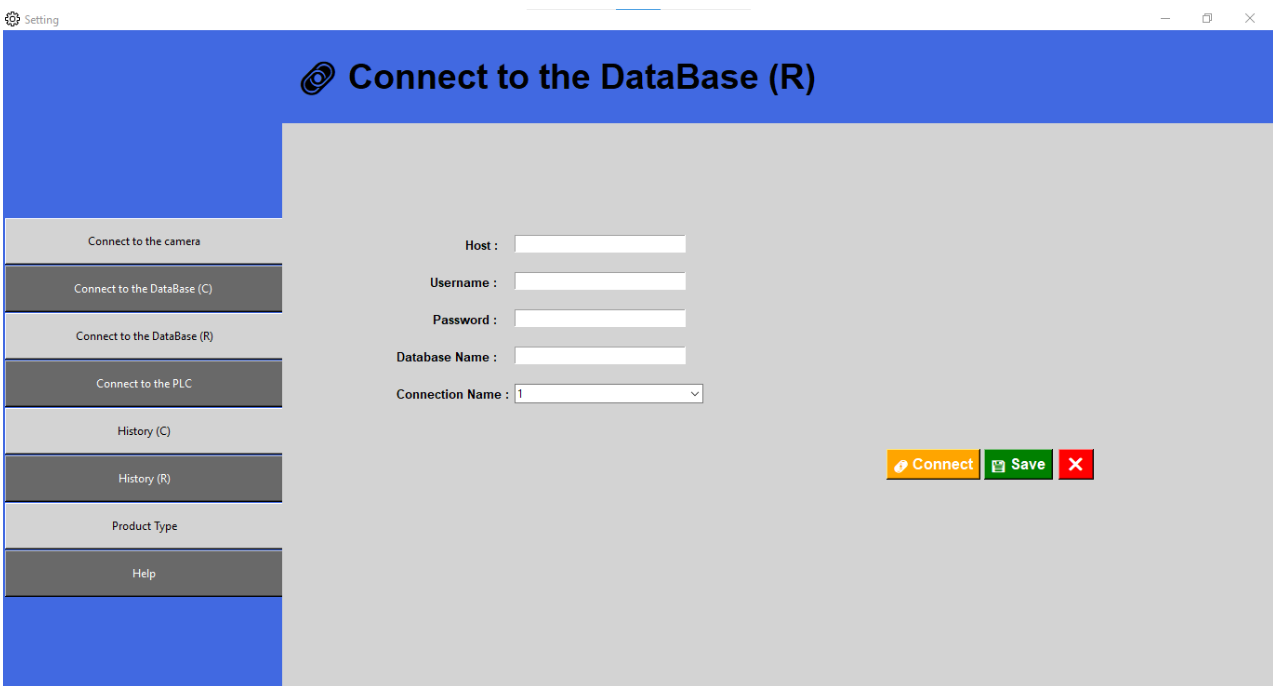


Figure III.49: User interface for connecting to the reference database

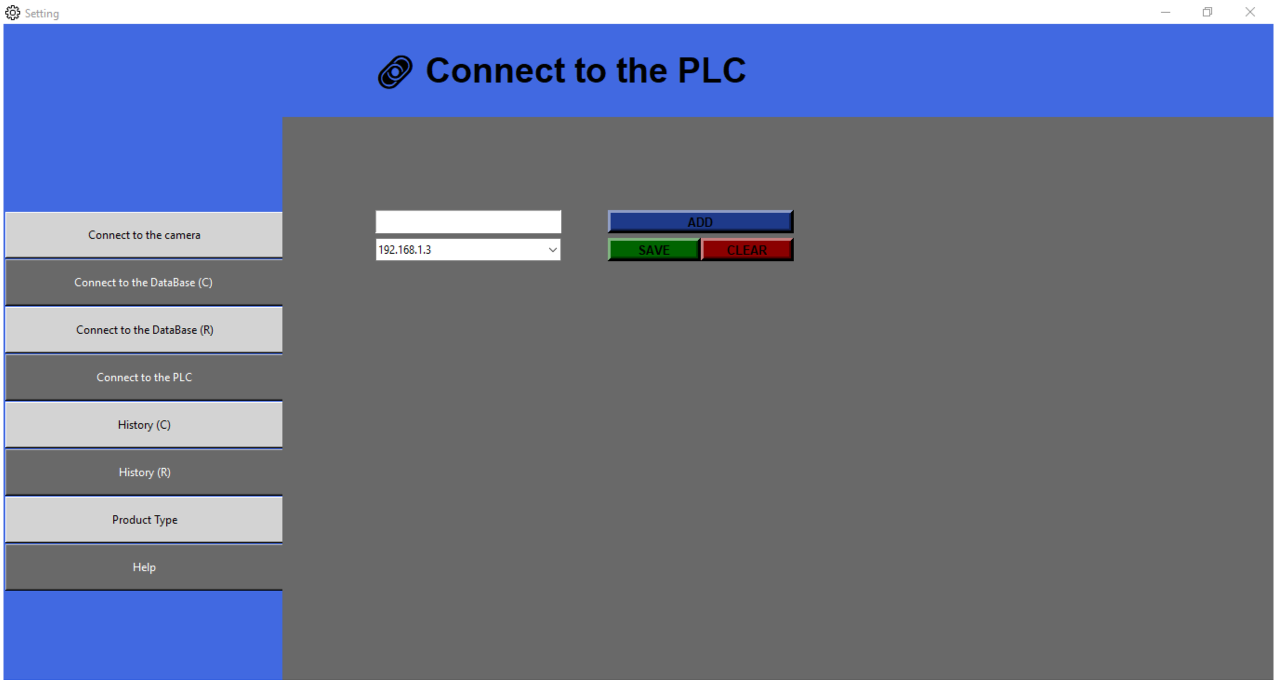


Figure III.50: User interface for connecting to plc

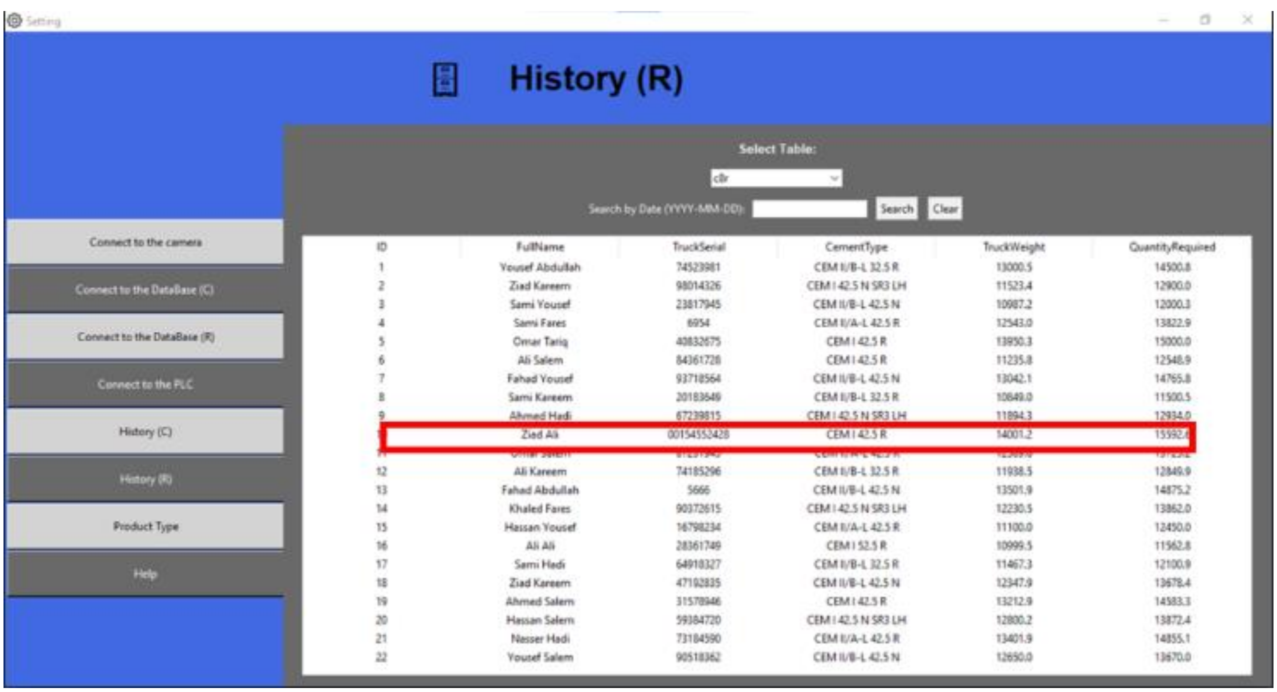


Figure III.51: Packing operations for today in Reference Database

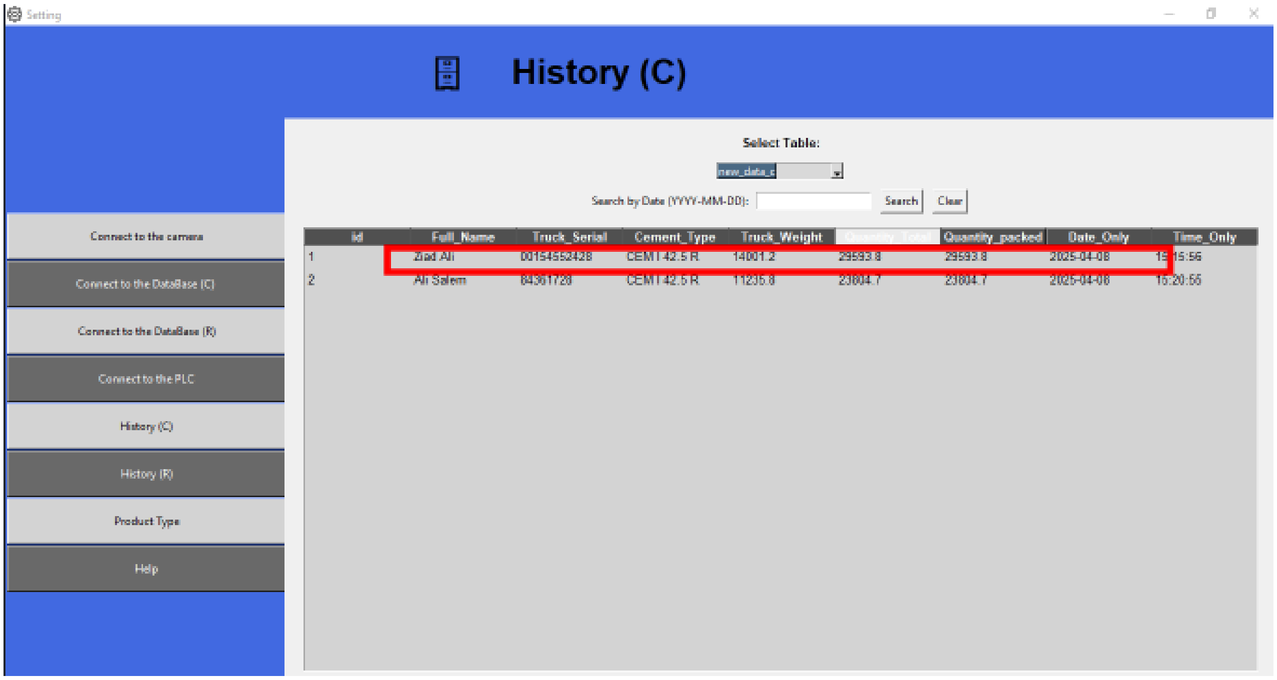


Figure III.52: Packing operations for today in Historical Database

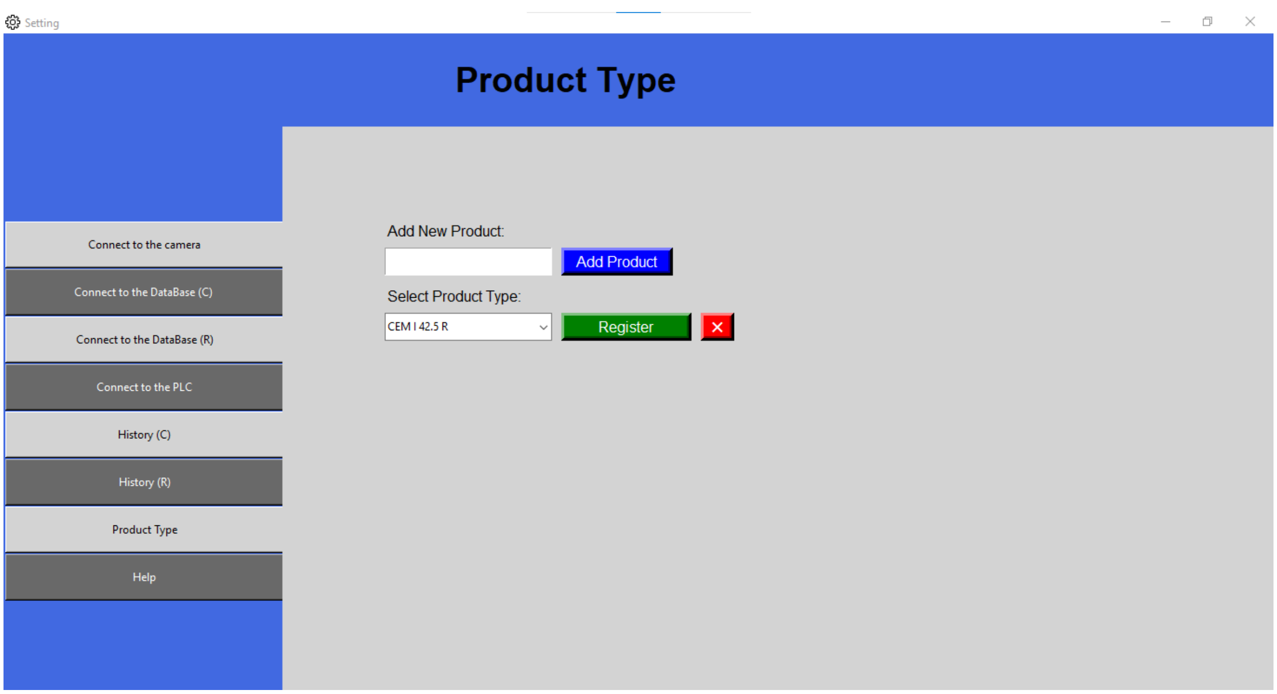


Figure III.53: Set cement type for the station

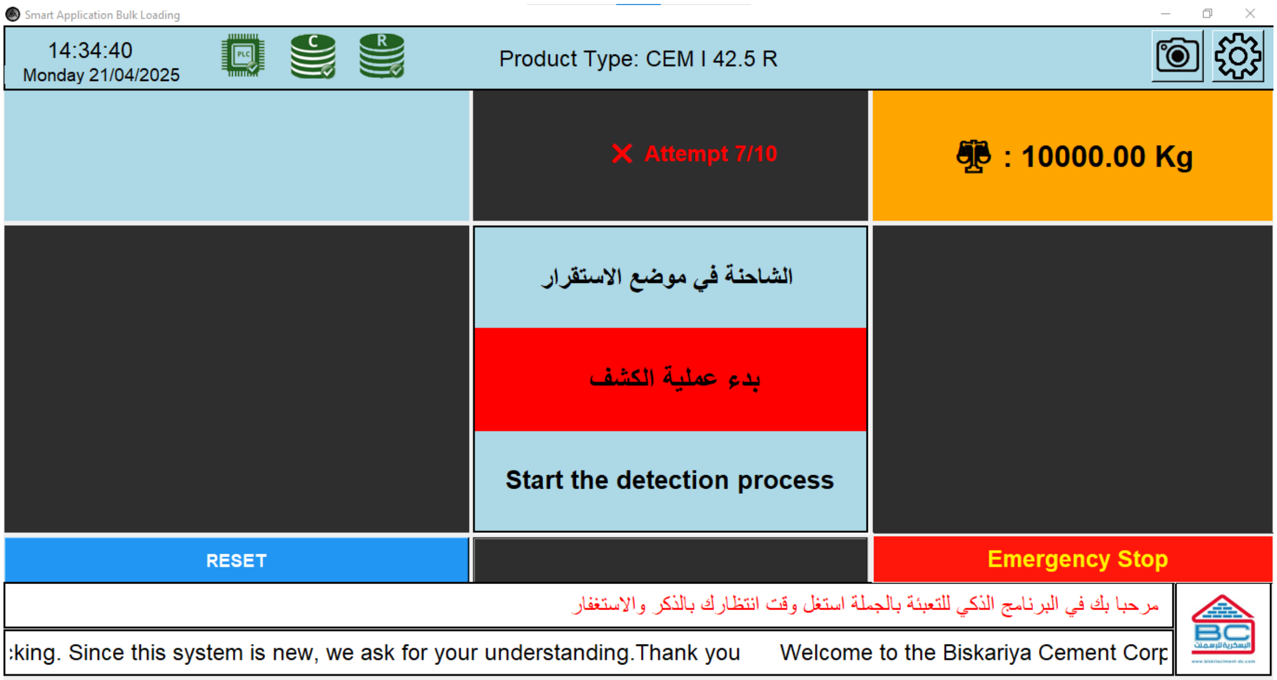


Figure III.54: User interface to start the process

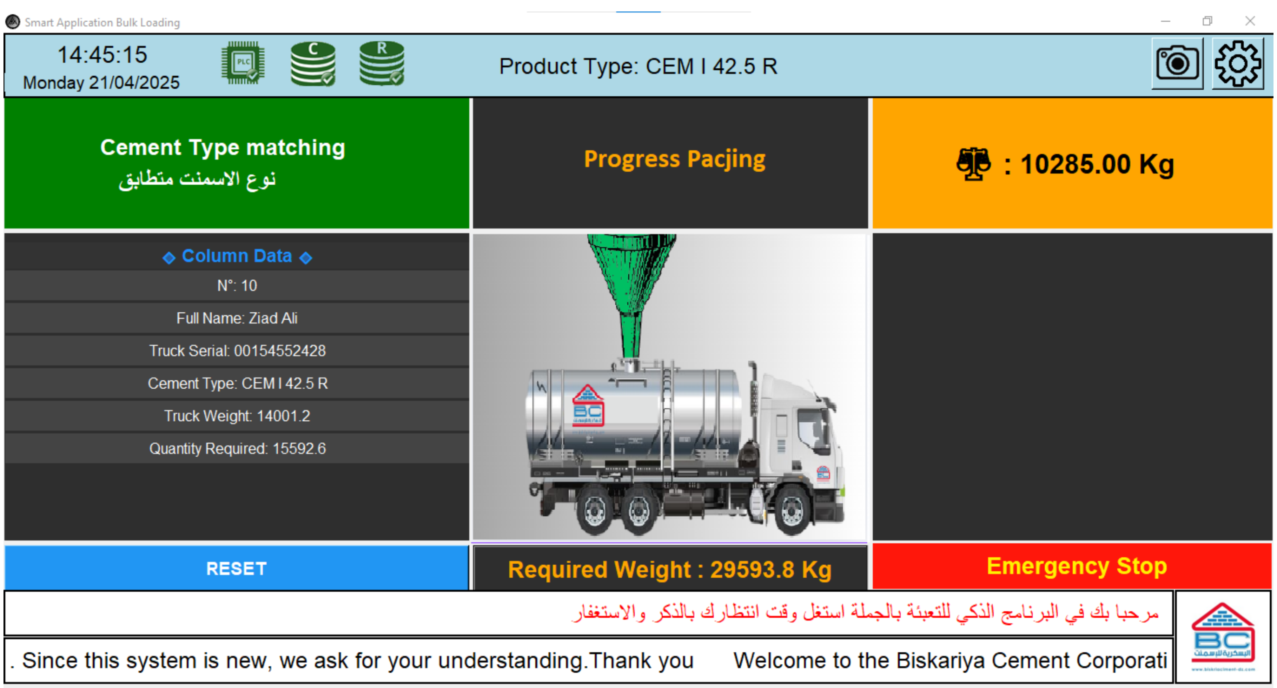


Figure III.55: User interface when the match is successful

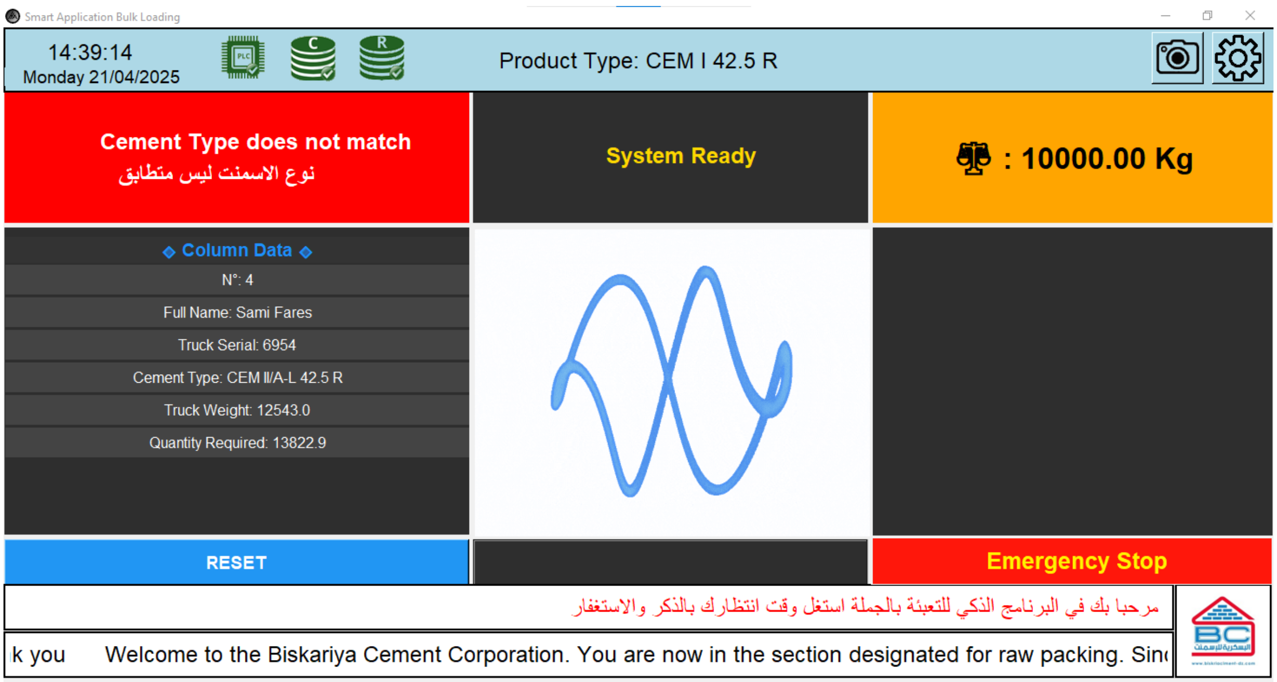


Figure III.56: User interface when matching fails

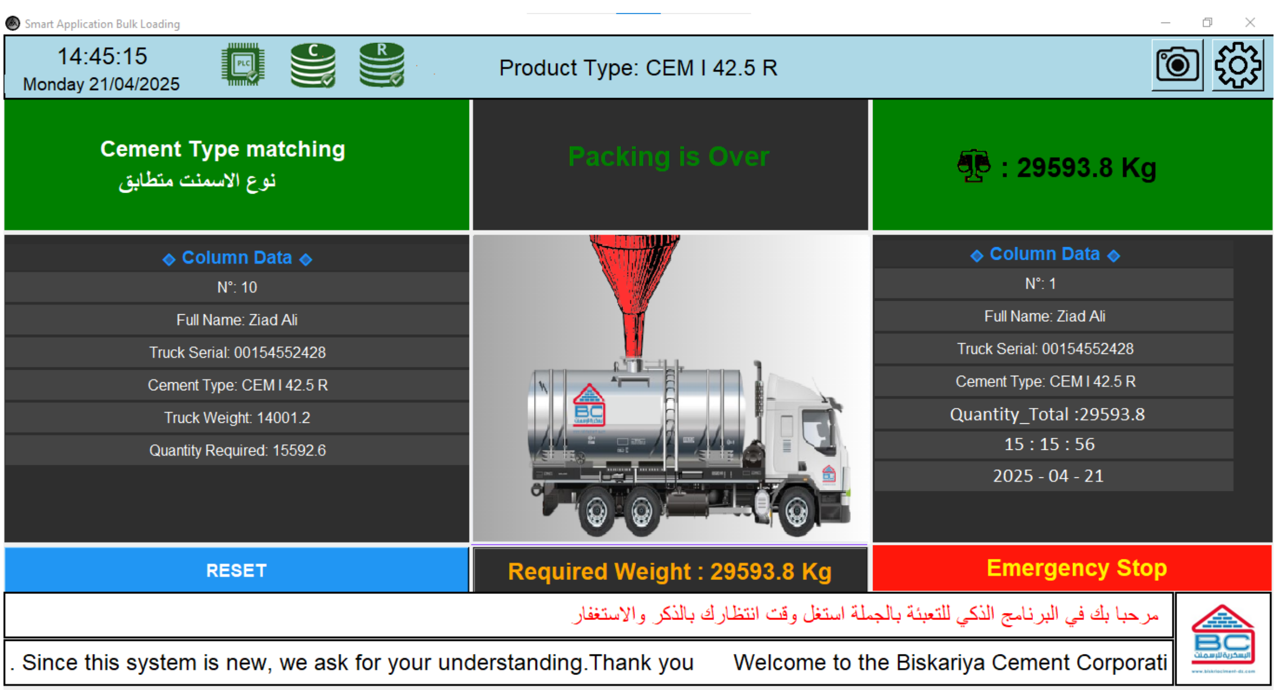


Figure III.57: User interface when the packing is over

Figure III.58: User interface during emergency stop

III.11 Features of the system

One of the most important features of the application:

- **Live Weight Tracking :**The system displays the truck’s weight in real time during loading, ensuring precise control over the filling process.
- **Automatic License Plate Detection :**The application uses image processing to automatically detect and read the truck’s license plate via a connected camera.
- **Reliable PLC Communication :**Stable and continuous communication with the S7-1200 PLC is maintained through the Snap7 library, enabling real-time interaction.
- **Integrated Database Access** :The system connects to a MySQL database to retrieve and store critical information about trucks, clients, and filling history.
- **Connection Watchdog** A built-in watchdog monitors communication with the PLC. If the connection is lost, the system reacts immediately to prevent issues.
- **Intuitive User Interface** Developed using Tkinter, the GUI allows operators to easily manage and monitor all system operations with minimal training.
- **Automated Filling Control** The application automatically stops the filling process once the target weight is reached, improving efficiency and reducing errors.
- **Cement Type Validation** The system verifies the required cement type for each truck by matching its license plate with database entries.
- **Real-Time Error Alerts** If a problem occurs—such as an unrecognized plate, connection failure, or type mismatch—the system instantly notifies the user.

Figure III.59: Features of the system

III.12 Conclusion

In this chapter, we presented the practical aspect of the project, which involved the development of a smart application for bulk cement filling. The application was implemented and tested in the field at the Biskria Cement Plant, where it showed positive results in improving the accuracy of the filling process and automatically verifying the material type.

The field trial confirmed the effectiveness of the application in a real industrial environment, highlighting its potential to enhance control and production systems in factories. Moreover, the system's flexibility and scalability make it suitable for deployment in other industrial facilities

General conclusion

This thesis has developed and implemented a smart application to enhance the efficiency and reliability of bulk cement loading operations within an industrial setting (Biskria Cement). By integrating key technologies—**Python, OpenCV, MySQL, and TIA Portal**—the system successfully bridges the gap between traditional industrial automation and modern software-driven solutions.

Leveraging image processing for **automatic license plate recognition**, the application enables accurate truck identification, cross-verification of cement types via a central database, and seamless synchronization with the **Distributed Control System (DCS)**. Real-time communication with **Programmable Logic Controllers (PLCs)** through the **Snap7** library allowed for precise control and monitoring, reducing operational errors and increasing overall accuracy.

Field tests conducted at the Biskria Cement Plant validated the effectiveness of the solution, highlighting the potential of smart automation in enhancing productivity, traceability, and reducing dependence on manual interventions.

While the system successfully meets the current operational requirements, several enhancements can be explored in future work, including:

- Integration of **RFID technology** for faster and more secure truck identification.
- Development of a more advanced and intuitive **Graphical User Interface (GUI)**.
- Implementation of **machine learning algorithms** to support predictive maintenance and process optimization.

In conclusion, this work demonstrates that adopting smart technologies in the cement industry not only improves operational performance but also lays a strong foundation for scalable, intelligent, and sustainable industrial solutions aligned with Industry.

Horizons

- Seeking to establish a company in order to start manufacturing and marketing the product.
- Developing the system to include:
  - Artificial intelligence techniques to predict malfunctions and improve decision-making.
  - Enhance the system by integrating it with RFID and GPS technologies to enable real-time tracking and identification of trucks inside and outside the factory.
- Adopting this type of system in the future in other industries that depend on transportation and loading, which contributes to accelerating digital transformation in the industrial sector in general.

BIBLIOGRAPHY

[I] Medcem. (n.d.). Artificial intelligence applications in cement industry. "https://medcem.com.tr/en/blog/artificial-intelligence-applications-in-cement-industry-122 "

[II] Vokuda, R. (2025). How manufacturing 4.0 initiatives are transforming the cement industry. CIOTechOutlook."https://www.ciotechoutlook.com/cxoinsight/How-Manufacturing-40-Initiatives-Are-Transforming-the-

[1]. SPA Biskria cimenterie. (n.d.). Documentations de l'usine.

[2]. Ben Hassin, L. (2023). Calculer de Bilan aéraulique et énergétique pour une ligne de production de 500KT de clinker [Mémoire de master, Université Mohamed Khider de Biskra].

[3]. Creusot-Loire Entreprise. (1978). Manuel de conduite cimenterie de Beni Saf.

[4]. Mebarki, H. (2013). Étude comparative des clinkers de l'ouest Algérien : Analyse physicochimique et Applications [Mémoire de master, Université Abdo Bekr Belkaid Tlemcen].

[5]. Benguedouar, A. (2013). Synthèse et caractérisation de silicates de calcium hydrates (C.S.H) hybrides [Mémoire de Magister, Université frères Mentouri Constantine 1].

[6].[8]. Jabeen, H. (2023, July 10). What is Distributed Control System (DCS)? Automation Articles.

[7]. Schlick, J. (2012). Cyber-physical systems in factory automation – Towards the industrial revolution. In 9th IEEE International Workshop on Factory Communication Systems (WFCS) (p. 55).

[9]. Colombo, A. W., Bangemann, T., Karnouskos, S., Delsing, J., Stluka, P., Harrison, R., Jammes, F., & Lastra, J. (2014). Industrial Cloud-based Cyber-Physical Systems: The IMC-AESOP Approach. Springer. ISBN: 978-3-319-05623-4.

[10].[11]. Boughrira, A. (2019). Exploitation d'un système de contrôle distribué « DCS : Distributed Control System » pour la réalisation d'un système de Télé-contrôle SCADA [Mémoire de master, Université Batna-2].

[12]. Chemical Engineering Department King Saud University. (2002). Process Control in the Chemical Industries.

[13]. Siemens. (n.d.). What is the PCS7? Retrieved from https://www.siemens.com/global/en/search.html?originLanguage=EN&originRegion=GLOBAL&activeTab=all&q=What+is+the+PCS7+

[14]. Gobeshona. (n.d.). Introduction to Simatic PCS7-2. Retrieved from https://gobeshona.com.bd/introduction-to-simatic-pcs7-2/

[15]. Siemens. (n.d.). TIA Portal. Retrieved from https://www.siemens.com/global/en/search.html?q=TIA+Portal

[16]. Hosseinzadeh, H. (n.d.). Master of Science in Mechatronic Engineering, Politecnico di Torino.

[17]. IndMall. (n.d.). What is the TIA Portal in Siemens? Retrieved from https://www.indmall.in/faq/what-is-the-tia-portal-in-siemens/

[18]. Electro-Matic. (n.d.). Siemens TIA Portal version 20. Retrieved from https://blog.electro-matic.com/siemens-tia-portal-version-20

[19]. Université Paris Cité. (n.d.). Introduction à la programmation Python pour la biologie [Cours de Python].

[20]. (n.d.). Comparative review of JAVA and Python. International Journal of Research and Development in Applied Science and Engineering (IJRDASE), ISSN: 2454-6844.

[21]. Analytics Vidhya. (2024, January). Advantages of Python over other programming languages. Retrieved from https://www.analyticsvidhya.com/blog/2024/01/advantages-of-python-over-other-programming-languages/

[22]. Ahmed, K. A. (n.d.). Introduction to a server: Web design.

[23]. Department of Computer Application, SRCHI School of Management and Tech. (n.d.). An introduction to client/server computing.

[24]. Fibermall. (n.d.). Networking servers. Retrieved from https://www.fibermall.com/ar/blog/networking-servers.htm

[25]. MySQL. (n.d.). MySQL 8.0 Reference Manual Including MySQL NDB Cluster 8.0 (p. 4).

[26]. (n.d.). Working with MySQL. Informatics Practices Class XI (CBSE Board).

[27]. SolarWinds. (n.d.). MySQL database. Retrieved from https://www.solarwinds.com/resources/it-glossary/mysql-database

[28]. Hamill, M. J. (2020). Industrial Communications and Control Protocols (p. 3).

[29]. [32]. Roe, N., & Miller, D. (Ed.). (n.d.). Industrial networking 101: Everything you need to know. Control Engineering. Retrieved from https://www.controleng.com/industrial-networking-101-everything-you-need-to-know/

[30]. Lapp Holding AG. (2019, July). Industrial Communication (pp. 22–23).

[31]. Lapp Holding AG. (2019, July). Industrial Communication (p. 32).