Politecnico di Torino

POLITECNICO DI TORINO

Master's Degree in Mechatronic Engineering

Master's Degree Thesis in collaboration with Nazari Automazioni Srl

Development of a preventive maintenance software for an automation plant

Supervisors

Prof. Alessandro RIZZO

Company tutors

Luca NAZARI

Nicola GALLIANO

Candidate

Maria Chiara CALDERARO

October 2021

Abstract

In this work of thesis, a real problem is faced concerning the need for maintenance on an automation plant for water transport. The plant in question already has a SCADA system developed on Ignition, so the same program was used to develop software for preventive maintenance of the plant. One of the main problems of the plant is related to ordinary maintenance such as the cleaning of the areas and the lubrication of the components. Although it may seem trivial, these tasks are of fundamental importance and carrying them out periodically would result in better functioning of the system and greater safety. The Python programming language was used to customize the behaviour of graphical objects already present in Ignition. The first step was to realize the structure of a Relational Database in which to store the data created through Microsoft SQL Server. The database was divided into three main parts, one relating to the storage of plant data, one relating to the operators and their tasks and the last one concerns the organization of the interventions and the algorithms that regulate them. Then it was used the Ignition platform to develop the SW divided into two parts, one to collect data and populate the DB and the other to allow operators to manage interventions on the plant and consulting the data. Once the development was finished, the software was tested on the real plant to verify the correct functioning.

Acknowledgements

Vorrei ringraziare il Professor Alessandro Rizzo per l'attenzione che ha dimostrato nei miei confronti durante lo svolgimento di questa tesi, per aver sempre trovato un momento per confrontarci e per avermi guidato fino alla conclusione di questo elaborato.

Un sentito ringraziamento va all'azienda Nazari Automazioni Srl, in particolare a Nicola, Alessandro e Andrea per avermi accompagnato in questo percorso offrendomi costante disponibilita', supporto, professionalità ed esperienza. Ci tengo a ringraziare anche Luca, Claudia, Matteo e Salvatore per avermi accolto in azienda e essermi stati di supporto in questi ultimi mesi.

Ringrazio i miei genitori per aver reso possibile questi anni a Torino e per essere stati presenti in ogni momento difficile durante il mio percorso e per aver sempre creduto in me.

Table of Contents

List of Figures VII

Acronyms X

Introduction 1

1 Theory 5
- 1.1 Preventive maintenance . . . 5
- 1.2 Graphical User Interface . . . 6
- 1.3 Relational Database Management System and Structured Query Language . . . 7
- 1.4 SCADA Supervisory Control And Data Acquisition and Ignition . . . 13

2 Database 17
- 2.1 Plant . . . 17
- 2.2 Operators . . . 19
- 2.3 Interventions . . . 20
- 2.4 Other . . . 23

3 Software for database population 25
- 3.1 Plant configuration . . . 27
- 3.2 Operator configuration . . . 30
- 3.3 Intervention configuration . . . 32

4 Operator interface 37
- 4.1 Consultation Plant . . . 38
- 4.2 Management of interventions . . . 40
- 4.3 Alarm display . . . 44

5 Testing 46

Conclusion 54

A Rules: Parameters and algorithms 56

B Query SQL 71

Bibliography 75

List of Figures

1 RTU plan of the upstream plant . . . 3

1.1 Maintenance costs . . . 6
1.2 Relational Database . . . 9
1.3 Classification of command in SQL . . . 11
1.4 Different types of JOIN . . . 12
1.5 SCADA System Implementation Example [7] . . . 13
1.6 PLC Control System Implementation Example [7] . . . 15
1.7 Ignition Logo . . . 16

2.1 Database of Plant . . . 17
2.2 Database of Operators . . . 19
2.3 Database of interventions . . . 20
2.4 Other tables . . . 23
2.5 Full Database . . . 24

3.1 Navigation bar . . . 25
3.2 Advanced settings . . . 26
3.3 Example of a page for entering plant data . . . 28
3.4 Many-to-many category and subcategory association . . . 29
3.5 Operator configuration . . . 30
3.6 Many-to-many operator type and intervention type association . . . 31
3.7 General intervention page . . . 32
3.8 Opening intervention page . . . 33
3.9 Select location . . . 35
3.10 Complete interventions page . . . 36

4.1 Opening page . . . 38
4.2 Browse Plant . . . 39
4.3 Filter . . . 40
4.4 Planned Interventions . . . 41
4.5 Ongoing Interventions . . . 42

4.6 All Interventions 43
4.7 Alarms 44
4.8 Development over time 45

5.1 Creation of tank level check intervention 46
5.2 Development of tank level over time 47
5.3 Conclusion of the intervention 48
5.4 Time Line 48
5.5 Creation of a genset running status intervention 49
5.6 Switching the UPS connection bit over time 50
5.7 Table ALLARMI 50
5.8 Device configuration 51
5.9 Creation of an intervention on the time life of the device 51
5.10 Creation of a battery discharge check 52
5.11 Time Line 53
5.12 Table INTERVENTI 53

A.1 Algorithm for the rule:"TEMPO" 58
A.2 Algorithm for the rule:"VITA A TEMPO" 59
A.3 Algorithm for the rule:"VITA A CONTEGGIO" 61
A.4 Algorithm for the rule:"CONTROLLO BIT" 62
A.5 Algorithm for the rule:"CONTROLLO BIT A TEMPO" 64
A.6 Algorithm for the rule:"CONTROLLO BIT RITARDO" 65
A.7 Algorithm for the rule:"ANALISI ANALOGICO" 68
A.8 Algorithm for the rule:"ANALISI SOGLIE" 69
A.9 Algorithm for the rule:"ANALISI SOGLIE A TEMPO" 70

B.1 Table with the output of the query SQL written above 74

Acronyms

SCADA
Supervisory Control And Data Acquisition

RTU
Remote Terminal Unit

GUI
Graphical User Interface

DBMS
Database management system

RDBMS
Relational database management system

SQL
Structured Query Language

HMI
Human Machine Interface

Introduction

My thesis took place in the company "Nazari Automazioni Srl" situated in Verzuolo, Cuneo. The company deals with industrial automation. In particular, my work has been carried out on a plant of "Consorzio Irriguo", which deals with the transport of the waters in the area surrounding Cuneo. Most of the problems are related to poor cleaning and maintenance of the system, or undetected abnormal values, or to sudden breakage of parts due to failure to replace them at the right time thus, to solve the problems of the plant and optimize its operation, a software has been developed that deals with the preventive maintenance of the plant. The software was developed through Ignition which was already used to implement the SCADA that allows monitoring the plant. A database has also been developed, through SQL Server, which serves to store data useful for the operation of the software.

Automation aims to automate machines and plants to such an extent that they can work autonomously, efficiently and with a low error rate. An automated system consists of the plant to be controlled, the automation computer control system and the peripheral systems. This environment has sensors, which collect the control data, and actuators, which apply the commands. Another element is the interconnecting field buses, which connect the different components of the automation system and to the control unit. The control of installations in the industrial automation sector is entrusted to a flexible digital control system with programmable logic (PLC). Commands are sent to the system via actuators such as motors, valves or magnets. Industrial automation has a complex communication system with many different sensors and actuators. The components are interconnected by field buses such as Profibus or CAN-Bus. Based on standardised protocols, the system regulates how different users use the line. Today, communication is also via Ethernet, so that systems can be maintained remotely. Users interact with the automation technology systems via the HMI, gaining a real-time overview of all key operating processes. They can observe the plant, control the machine and intervene in the processes. The control panel is equipped with lights, buttons and displays. Software-based visualisation is via the operator panel. Industrial automation is considered the key industry for Industry 4.0. This term denotes intelligent, digitally interconnected

systems that enable largely autonomous industrial production. In a Smart Factory, people, plants, products and logistics are interconnected with a communication system.

Plant overview

The plant consists of several stations scattered around the territory, called Remote Terminal Unit (RTU). It is an automated and remote management system that transports water to the fields and hydroelectric plants through penstocks.
RTU is a microprocessor-controlled electronic device that interfaces objects in the physical world to SCADA system by transmitting data to a master system, and by using messages from the master supervisory system to control connected objects.
The system is made up of eighteen RTUs, each with an electrical panel containing a PLC, analogue cards for retrieving measurements in the field, batteries or UPS or a solar panel, an EWON and an operator panel. The EWON serve to communicate via sim or wired internet and provides the VPN connection to the control centre. Outside, there may be floodgates, actuators and various sensors such as level or flow sensors. As it is a plant scattered over a vast territory in the middle of nature, it is rarely cleaned and maintained as it would be used for the proper functioning of it. One of the main problems is precisely related to this aspect. For example, the floodgate in the plant are not moved for a long time during the year, so they may remain blocked at the time it would be necessary to move them. This problem could be avoided by periodically lubricating the gears and cleaning the leaves and branches that can be deposited in the mechanisms. Another problem may be related to the incorrect detection of a measurement that may result from the presence of a foreign body in the vicinity of a sensor, monitoring its trend constantly it possible to detect and report an anomaly and solve the problem. The plant is divided into two macro plants, the downstream plant and the upstream plant. Each plant is managed by a SCADA that, as mentioned before, has been developed using the Ignition software.

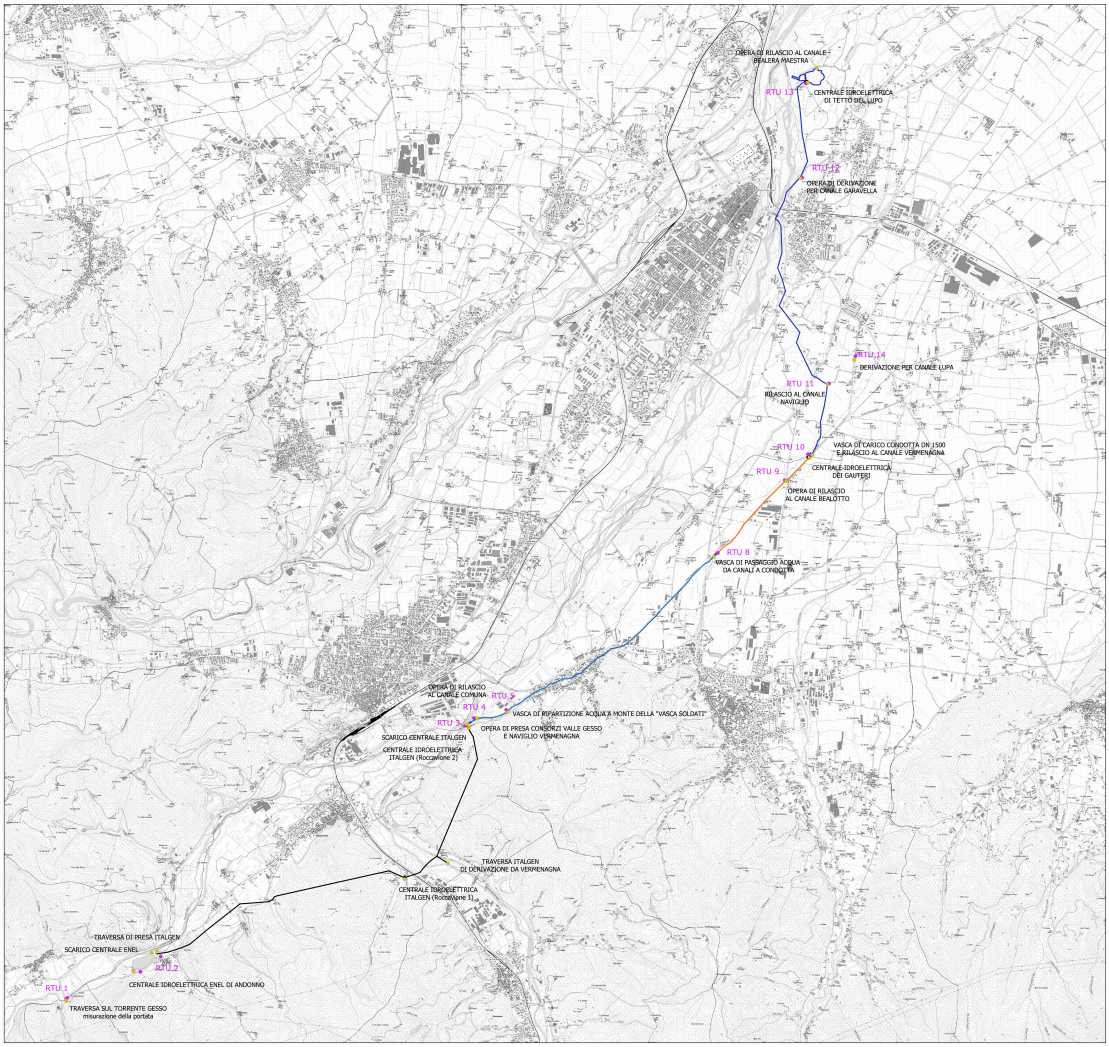

Figure 1: RTU plan of the upstream plant

Thesis overview

My thesis is organized as follows:
Chapter one covers the theory needed to understand my work, so it includes an explanation of what preventive maintenance is and what preventive maintenance is for, what aspects should be taken into account when creating a graphical user interface, what are the Relational Databases and in particular Microsoft SQL Server and what is a SCADA and the SCADA Ignition in particular.
Chapter two explains the Database which was developed as a basis for storing Software data.

Chapter three explains the first part of the software, which is the one that deals with the population of the Database, then the insertion of data relating to the plant, the operators and the creation of the necessary interventions.
Chapter four explains the second part of the software that provides the operator interface for the management of interventions.
Chapter five tests the software on the actual plant and reports the results.
There are two appendices, one dedicated to the rules determining interventions and their algorithms and the other to SQL queries.

Chapter 1

Theory

1.1 Preventive maintenance

Preventive maintenance is a type of maintenance whose main objective is to extend the life cycle of components and to limit corrective maintenance actions.
A preventive maintenance programme shall consist of:

- Non-destructive tests,
- Periodic inspections,
- Planned activities like:
  - Cleaning
  - Lubrication
  - Replacement of end-of-life components
  - Component calibration

The cleaning and lubrication part is really important and is often left out, but with ordinary maintenance, it is possible to prevent a small problem from becoming more serious. This makes it possible to avoid the costs of replacing equipment and downtime and to improve the safety of the working environment by minimizing the risks to the installations.

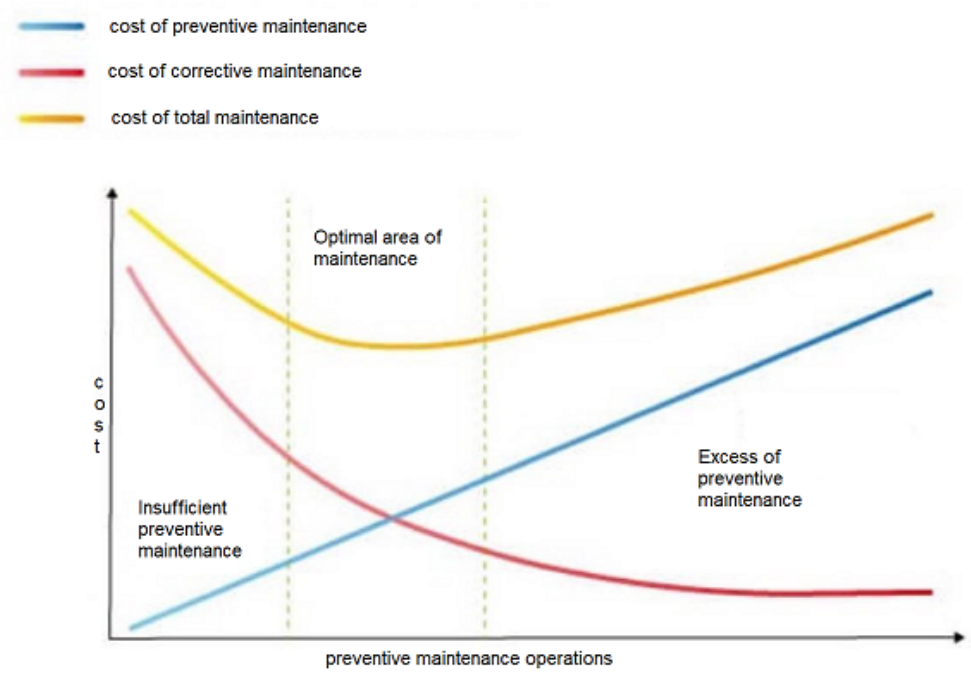

Figure 1.1: Maintenance costs

As it is possible to see from the image above, taken from [1], an excess of prevention can lead to an increase in maintenance costs, in the same way, if it is not carried out with a proper frequency will have to carry out more corrective interventions that will influence a lot on the total costs.

1.2 Graphical User Interface

The graphical user interface (GUI) allows users to interact with the computer using object user-friendly like graphical icons. One of the most important aspects of a GUI is usability. According to Jakob Nielsen, as explained in [2], usability has these five attributes:

- Learnability: The system should be easy to learn.
- Efficiency: The system should be efficient to use.
- Memorability: The system should be easy to remember.
- Errors: The system should have a low rate error.
- Satisfaction: The system should be pleasant to use.

Nielsen suggest a model with multiple steps [2] but since the interface for this thesis work is aimed at a low-level user, the Nielsen model has been reduced to the following key points:

- Know the user;
- Setting usability goals;
- Parallel design;

Knowing the user before starting the development of software is essential because depending on the level of education, computer knowledge, user work experience the complexity of the software must be adapted. Another fundamental step is to understand what are the main objectives to be met in terms of measured usability. Usually, not all the usability attributes listed before have the same weight for the user who will have to use the GUI. Another strategy is to adopt a parallel design to explore more alternatives before one chooses a single model.

1.3 Relational Database Management System and Structured Query Language

Databases play a crucial role in scientific research, the useful information is obtained from processed data. So, data has to be interpreted to obtain information. Database Management System (DBMS) refers to a collection of interrelated and persistent data and a set of application programs used to access, update and manage data. The principal goals of DBMS are [3]:

- Availability: More users can easily access the data
- Integrity: The data available in DB is a reliable data
- Security: Only users with permission can access the data
- Independence: The user can store, update and retrieve data in a structured way. Complex data structured are implied to represent data.

The relational data model was introduced by C. F. Codd in 1970. The book [4] says that "The relational data model describes the world as a collection of inter-related tables". Codd defines a set of 12 rules call "Codd's Rule" designed to define what is required from a DBMS to be considered relational [5]:

1. The Information Rule: Data are saved in tabular form.
2. Guaranteed Access Rule: Each data element should be univocal approachable.

3. Systematic Treatment of NULL Values: Null values are used to means missing information.
4. Dynamic On-line Catalog Based on the Relational Model: The database description should be accessible to the users.
5. Comprehensive Data Sublanguage Rule: There must be at least one language whose statements are expressible, as character strings and whose ability to support all the following are comprehensive: data definition, view definition, data manipulation, integrity constraints, authorization and transaction boundaries.
6. View Updating Rule: Data should be able to be substituted from the user through any view.
7. High-Level Insert, Update and Delete: All records must be modified with singular command.
8. Physical Data Independence: How data are stored or retrieved should not influence how a user approaches the data.
9. Logical Data Independence: A user's view should not be affected by its form in files.
10. Integrity Independence: To maintain data integrity can exist some constraints on user input.
11. Distribution Independence: A database design should enable the distribution of data over various sites.
12. Non-Subversion Rule: It is not possible to change the data concerning the organization of the database.

To manipulate data are used relational algebra and relational calculus.
The important keys of the relational data model are as follows [3]:

- Data are saved in tables, each row of a table is called "Tuple", each column is called "Attribute".
- The order of Tuple and Attribute does not matter.
- The intersection of Tuple and Attribute will have a single value called "Atomic Value".
- By definition, it is not possible to have two rows that are the same.
- Each column will have a unique name, and all column entries will be of the same type.

- Each relation must have a key, and keys can be a set of attributes.
- For each column, there is a set of possible values called "Domain".
- The degree of a relationship is the number of attributes. The cardinality of the relation is the number of tuples.

There are several types of relationships between tables [4]:

- One-to-many: it is the normality, found in any relational database.
- One-to-One: it is the relationship with only one other entity, it is rare in relational databases.
- Many-to-many: The association table must contain the primary keys of the original tables.

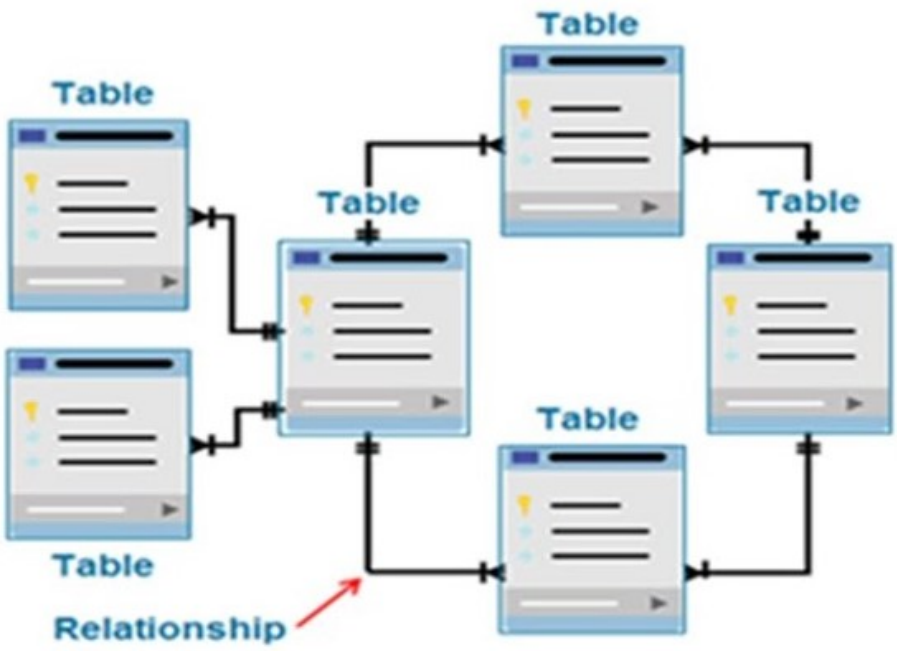

Figure 1.2: Relational Database

There are several constraints that are imposed on a table to regulate its behaviour, the most used are: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY. If an attribute is NOT NULL means that it can not have a null value, if an attribute is UNIQUE means that it can not have the same value in two different rows.

Tables in the database are linked together through primary key and foreign key relationships. These relationships are used in RDBMS to define relationships between tables. A primary key is a column whose values uniquely identify the row in the table. A foreign key, on the other hand, is one or more columns in a table whose values are equal to the primary key of another table.
When updating or deleting a value of a primary key that has a link to another table via a foreign key, the result depends on the referential action that has been specified in the subclause of the foreign key clause. It is possible to have ON UPDATE/ON DELETE:

- CASCADE: modifying the value in the main table automatically modifies the value of the corresponding rows in the association table.
- SET NULL: by modifying or deleting the row in the main table you set the column of the foreign key equal to NULL.
- NO ACTION: It is not possible to modify or delete the primary key.

[4] "The relational model has provided the basis for a standard database access language called Structured query language (SQL)".
SQL commands consist of statements that are used to query, insert, update, and delete data. They resemble English language sentences in their construction and use and it easy to learn[6]. SQL commands can be classified into three types:

1. Data Definition Language commands (DDL): used to define a database, including creating, altering, and dropping tables and establishing constraints.
2. Data Manipulation Language commands (DML): used to maintain and query a database, including updating, inserting, modifying, and querying data.
3. Data Control Language commands (DCL): are used to determine whether a user is allowed to carry out a particular operation or not.

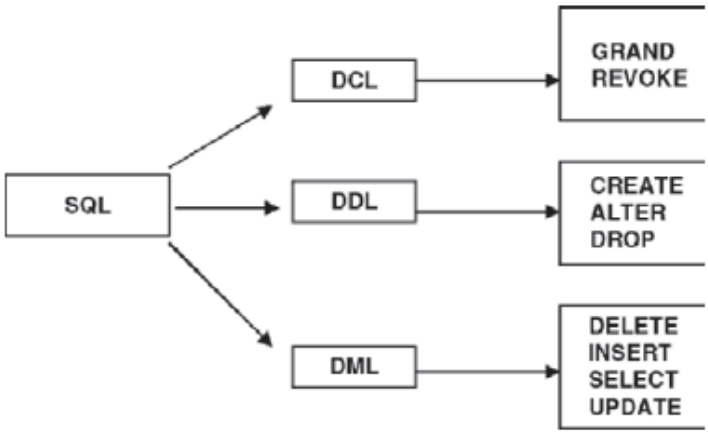

Figure 1.3: Classification of command in SQL

A base table is created with the CREATE TABLE statement and is used to hold persistent user data.
The steps in table creation are:

1. Identify data types for attributes.
2. Identify columns that can and cannot be null.
3. Identify columns that must be unique.
4. Identify primary key–foreign key mates.
5. Determine default values.
6. Identify constraints on columns (domain specifications).
7. Create the table.

The following commands are mainly used to manipulate the data:

- INSERT is used to add data into the database.
- UPDATE is used to modify the data in the database.
- DELETE is used to delete data in the database.

Another recurring command is the JOIN. A JOIN clause is used to combine rows from two or more tables, based on a related column between them.
There are different types of JOINs in SQL:

- (INNER) JOIN: Returns records that have matching values in both tables.
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table.
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table.
- FULL (OUTER) JOIN: Returns all records when there is a match in either the left or right table.

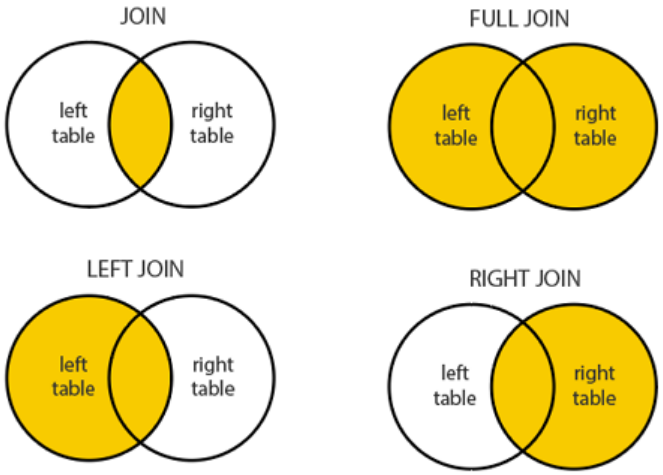

Figure 1.4: Different types of JOIN

1.4 SCADA Supervisory Control And Data Acquisition and Ignition

SCADA stands for Supervisory Control And Data Acquisition, SCADA industry was essentially born out of a need for a user friendly front-end to a control system. SCADA systems are highly distributed systems used to control assets, often spread over a wide geographic location, where centralized data acquisition and control are critical to system operation. They integrate data acquisition systems with data transmission systems and HMI software to provide a centralized monitoring and control system for numerous process inputs and outputs.

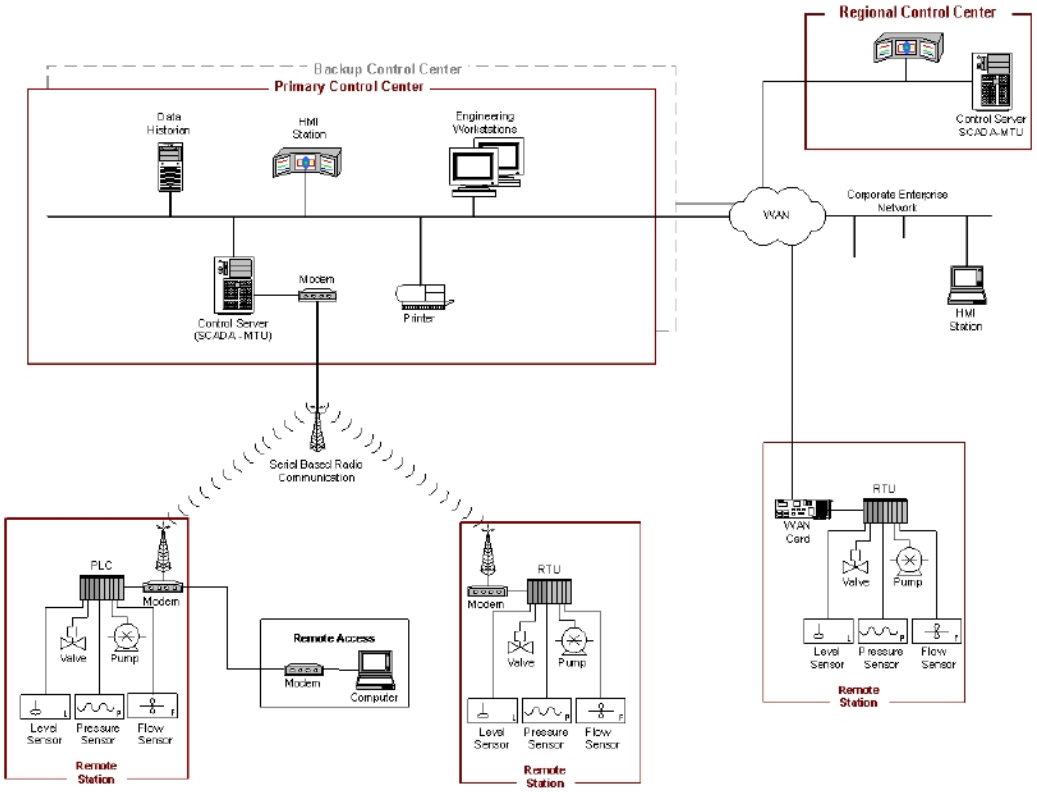

Figure 1.5: SCADA System Implementation Example [7]

The Human Machine Interface (HMI) is software and hardware that allows human operators to monitor the state of a process, modify control settings, and manually override automatic control operations in the event of an emergency. It also displays process status information to operators and authorized users [7].
Some rules have been developed to develop a high-performance HMI.

In Industrial HMI, the use of colours is very limited and is used to draw attention to important situations. High contrast can be uncomfortable for the eye so, the perfect background should be a light grey. For the foreground, a minimum number of colours should be used and very sparingly. In addition, the amount of text should be reduced to a minimum but not eliminated, "a graphic screen is not an instruction manual". There is a big difference between data and information, most poorly constructed operator displays show a lot of data, but little information. "Information is data in a context made useful" [8].

The Hardware of SCADA includes a Master Terminal Unit (MTU) placed at a control centre, communications equipment and one or more geographically distributed field sites. A Remote Terminal Unit (RTU) is a standalone data acquisition and control unit, generally microprocessor-based, which monitors and controls equipment at some remote location from the central station[9].

PLCs have replaced hardwired relays with a combination of the ladder–logic software and solid-state electronic input and output modules. They are often used in the implementation of RTU. PLCs are used as the control components to provide local management of processes through feedback control. An Example of PLC Control System Implementation is represented in Figure 1.6.
PLCs have a user-programmable memory for storing instructions to implement specific functions such as I/O control, logic, timing, counting and data and file processing. The PLC is accessible via a programming interface located on an engineering workstation, and data is stored in a data historian, all connected on a LAN. [7]
The communication system can be wire, fibre optic, radio, telephone line, microwave and possibly even satellite. Server-client and server-server communication is in general based on a publish-subscribe and event-driven and uses a TCP/IP protocol[10]. In a Master-Slave arrangement, only one machine (MTU) can initiate the communication. MTU calls one RTU, gives instruction, ask for information updates and order the RTU to respond. The RTU answers and MTU move to the second RTU. RTU can send a message only when ordered by MTU [11].

The software is programmed to tell the system what and when to monitor, what parameter ranges are acceptable, and what response to initiate when parameters go outside acceptable values. The products often have built in software redundancy at a server level, which is normally transparent to the user.

The products are based upon a real-time database (RTDB) located in one or more servers. Some products support logging to a Relational Data Base Management System (RDBMS)[10].

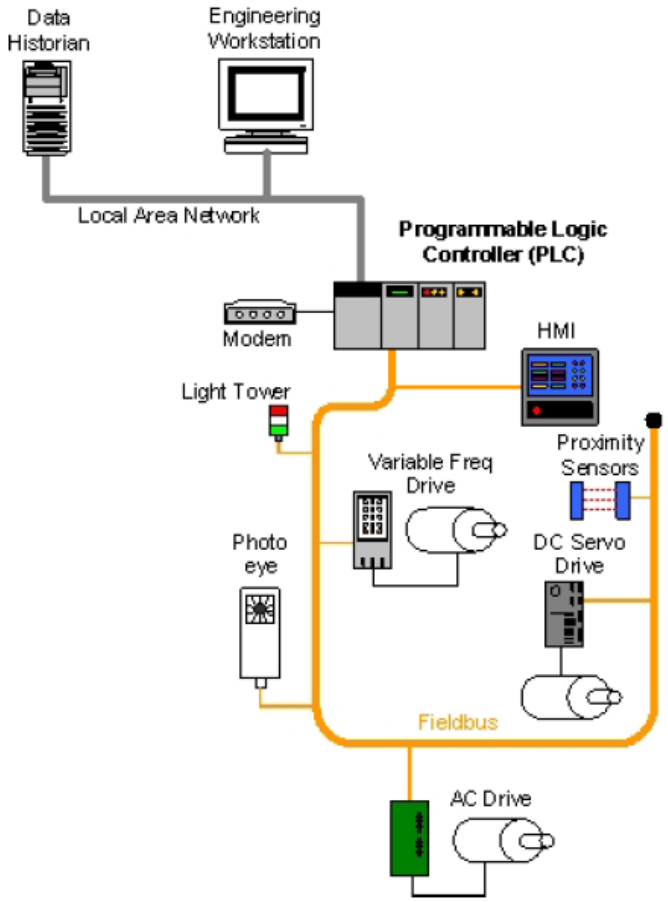

Figure 1.6: PLC Control System Implementation Example [7]

One of the features of SCADA is to allow grouping of users, so that each group is given a different type of permission to read and write the product. A SCADA also supports multiple screens that can be navigated to each other. There are several ready-made graphical objects, which can be used and customized as desired by the programmer. Most SCADAs also use Tags to connect graphical objects to devices. Another important part of most SCADA implementations are alarms. Alarms can be created in such a way that when their requirements are met, they are activated. The SCADA operator's attention is drawn to the part of the system requiring attention by the alarm. Multiple alarm priority levels are supported. Emails and text messages are often sent along with an alarm activation alerting managers along with the SCADA operator.

Figure 1.7: Ignition Logo

Ignition SCADA by Inductive Automation® combines an unlimited licensing model, with instant web-based deployment, and the industry-leading toolset for supervisory control and data acquisition, all on one open and scalable universal platform. It unify all the mayor software layers into a single solution. It offers unlimited clients,tags,connections. Ignition SCADA software comes standard with a comprehensive set of data acquisition tools which includes built-in OPC UA to connect to practically any PLC, and the ability to seamlessly connect to any SQL database. Ignition can also turn any SQL database into a high-performance industrial historian and connects to IIoT devices through MQTT. [12] Ignition is engineered to streamline data-throughput so it is possible to see true real-time tag values. It also allows pages and graphic objects to be customised using the Python programming language.

Chapter 2

Database

The first step for this thesis work was to design a SQL Database in which to store the plant data and all the data useful for the maintenance software. In the database there are three main macro sections. The first one is related to all plant data, the second one is related to the operators and the third one to the creation of interventions and their management. Some tables are in common between several sections. The complete database can be seen in the Figure 2.5.

2.1 Plant

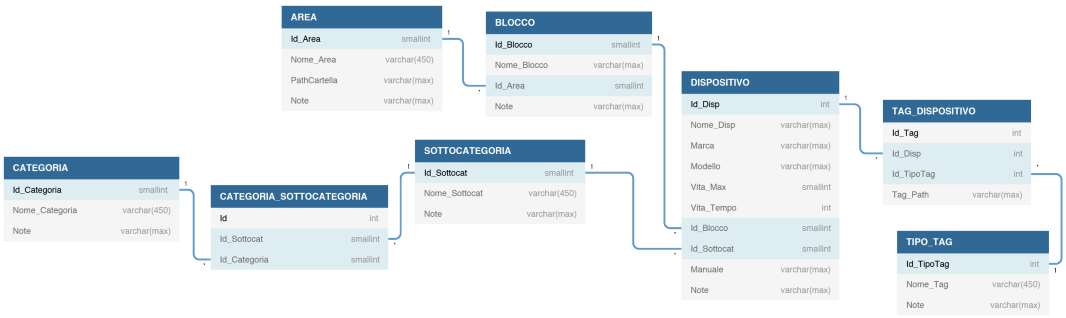

Figure 2.1: Database of Plant

The system has been schematised as follows:

- Each RTU scattered throughout the territory corresponds to an "AREA".
- Each area is composed of a series of "BLOCCHI" that enclose several elements, in the case of the installation under study, examples could be a floodgate, or a tank, or the environment itself.

- Each block consists of a set of "DISPOSITIVI", which are the physical devices that are installed in the RTU.
- Each device may have one or more "TAG DISPOSITIVO" connected to it. Tags are a property of SCADA mentioned in the previous chapter in section 1.4, they are used to connect graphic objects to devices.
- Each Tag will be associated with a "TIPO TAG" that indicates the type of tag, examples may be: faut, misura.

A further classification of device has been made. There are therefore:

- "CATEGORIE": An example could be sensors, motors or valves.
- "SOTTOCATEGORIE": Each sub-category will belong to one or more categories, an example could be level sensor or throttle valve.

Each table, except the many-to-many association tables, will have an ID which will serve as the PRIMARY KEY, and which will be called up where necessary as a FOREIGN KEY. All tables with a named row inside will have the UNIQUE CONSTRAINT.

The table "DISPOSITIVO" has two important values that can be specified:

- "Vita Max": for a device with switching, this indicates the maximum number of switching operations possible before the end of life.
- "Vita tempo": indicates the maximum number of life in hours for a device.

These values will be used to set interventions on the life of the devices.
A device may have either zero or one or more tags attached to it. Each tag will be linked to a tag type and for simplicity a device may have only one tag per type, this will be managed by software.

The table "CATEGORIA_SOTTOCATEGORIA" manages the relationship between categories and subcategories through a many-to-many association. Each subcategory may be associated with at least one or more categories. Devices will belong to one subcategory and consequently to one to several categories.

2.2 Operators

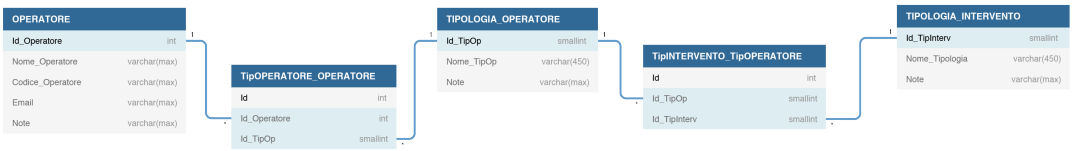

Figure 2.2: Database of Operators

To manage the operator's part, two tables have been created:

- "TIPOLOGIA INTERVENTO": this is the set of types of operations that may occur.
- "TIPOLOGIA OPERATORE": is the set of operator tasks. Each operator type can carry out one or more intervention types.

Also in this case the tables have an identification Id indicated as primary key.

There is a first many-to-many association table, "TipINTERVENTO_TipOPERATORE", that links the operator types to the intervention types.

In the "OPERATORE" table, the names of the operators and their e-mails are listed.

There is another many-to-many association table, "TipOPERATORE_OPERATORE", in which an operator is associated with one or more operator types, and consequently with one or more intervention types.
As each intervention will then be associated with an intervention type, only operators associated with that type will be able to carry out that intervention.

2.3 Interventions

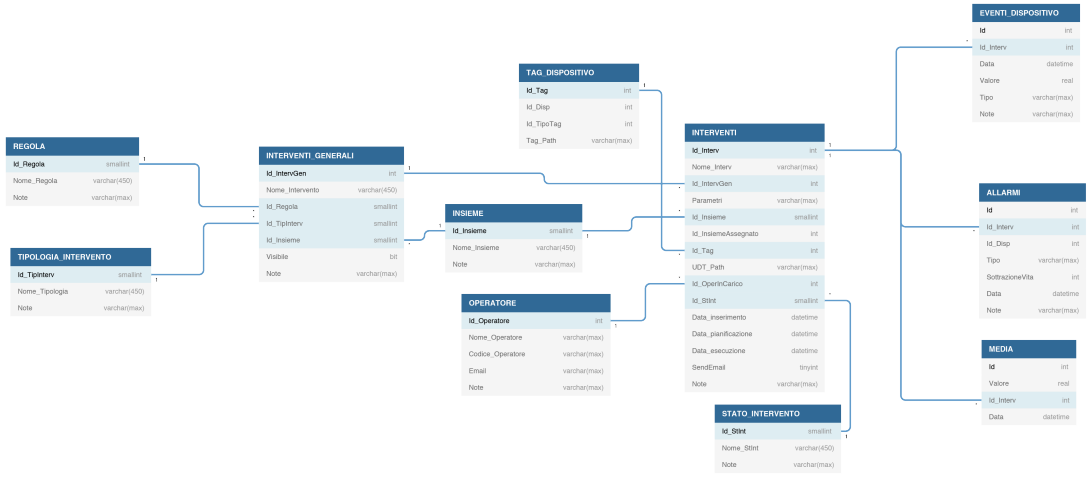

Figure 2.3: Database of interventions

The first table to be mentioned to explain the functioning of the interventions is "REGOLA", this table includes the logic that have been studied to define the interventions. Rules and their parameters are managed in Ignition using Datatypes. Each rule datatype has its parameters saved and a script that manages control over them. Each action will have an instance of the reference Datatype attached to it. A series of rules have been identified to which each intervention is linked:

- TEMPO: For interventions to be scheduled after a fixed period of time.
- VITA A TEMPO: Control over the time life of the device. The linked intervention is created by default.
- VITA A CONTEGGIO: Control over the switched life of the device. The linked intervention is created by default.
- CONTROLLO BIT: Control over one-bit switching.
- CONTROLLO BIT A TEMPO: Control over one bit switching in a given time.
- ANALISI SOGLIE: Control over the output of an analog value from a given range of thresholds.
- ANALISI SOGLIE A TEMPO: Control over the output of an analog value from a specified range of thresholds over a specified period of time.

- ANALISI ANALOGICO: Control over the variation of an analogue value.
- CONTROLLO BIT RITARDO: Control over the switching of a bit that remains at a fixed value for a specified period of time.

Each rule has its own set of parameters that will be entered by the user when creating the intervention and according to which it will be checked when the intervention is to be carried out. It is possible to find a deepening in the appendix A.

Table "INSIEME" is a support table pre-filled with Area, Block, Device, Category, Subcategory. Each element is associated with an "Id_Insieme".

The table "INTERVENTI GENERALI" is used to create generic operations that can be carried out. When creating a general operation, it is mandatory choose the rule of which it is part, the type of related operation and through the Id_Insieme, the set on which the general operation is to be carried out. The table is pre-filled by default with two component life interventions.

- CONTROLLO VITA TEMPORALE: is linked to the "VITA A TEMPO" rule and performs a check on the time life of the device.
- CONTROLLO VITA COMMUTAZIONI: is linked to the "VITA A CONTEGGIO" rule and and performs a check on the counting life of the device

The table "STATO INTERVENTO" is pre-filled by default with the available statuses of an intervention. They can be:

- CHECK PARAMETRI: The intervention has been entered but the conditions to trigger the alarm have not yet occurred.
- PIANIFICATO: Conditions occurred to trigger the alarm related to the intervention.
- IN CORSO: The intervention was taken over by an operator.
- CONTROLLO: The intervention is just a background check and no email should be sent.
- RISOLTO: The intervention is complete and successful.
- NON RISOLTO: The intervention is completed but not successful.
- CANCELLATO: The intervention has been deleted.

The table "INTERVENTO" is perhaps the most complex of the entire database. Therefore, each attribute will be explained below. When an intervention is created, after having assigned it a name ("Nome_Interv"), the general intervention to which it belongs is selected using the foreign key (Id_IntervGen); once this choice has been made, it is possible to fill in the parameters linked to the chosen rule; these parameters will be saved in JSON format ("Parametri"). Based on the set chosen in the previous step, it will be possible to select where the intervention is to be carried out by selecting from the plant data available ("Id_Insieme", "Id_InsiemeAssegnato") and "Id_Tag" if the intervention is connected to a PLC tag. Id_InsiemeAssegnato refers to the id of the table chosen via Id_Insieme. "UDT_Path" indicates the path of the Datatype instance linked to the rule. "Id_OperInCarico" is a foreign key indicating the operator taking over the task. It is only filled in when an operator takes over the task, changing the task status from "PIANIFICATO" to "IN CORSO". There are then three dates:

- "Data_inserimento" is the date of when the task was created and the status of the intervention becomes "CHECK PARAMETRI";
- "Data_pianificazione" is the date of when the status of the intervention becomes "PIANIFICATO";
- "Data_esecuzione" is the date when the status of the task becomes "RISOLTO/NON RISOLTO";

Three others tables are used for the management of interventions. To better understand how it works it is possible to see the appendix A.

- "EVENTI DISPOSITIVO": Records all Tag changes. Counting the repetitions in the table determines when an action is to be performed.
- "MEDIA": This is used for interventions linked to the "ANALISI ANALOGICO" rule. N samples every sampling time are recorded and the values in the table are averaged.
- "ALLARMI": Every time an intervention goes from "CHECK PARAMETRI" to "PIANIFICATO" it is saved in a record. Moreover, if the parameter "RegistraAllarmi" is selected in the rules, the check on the intervention continues even after it has passed to planned, and every time the conditions to trigger the intervention would occur, it is written in the table.

2.4 Other

Figure 2.4: Other tables

There is a table calls "PARAMETRI SISTEMA", in case of need of general information about the system.

The last table to be mentioned is "GESTIONE EMAIL". This is used to manage the sending of emails asynchronously to the software progress. Each time an email is requested for intervention or to remind of a deadline, a record of the table is filled in, indicating the recipient's email, the subject and the message text. There is then an asynchronous process that every ten seconds read the first record of the table, sends the email and deletes the record. This prevents several emails from being sent at the same time, which would block the graphics of the software.

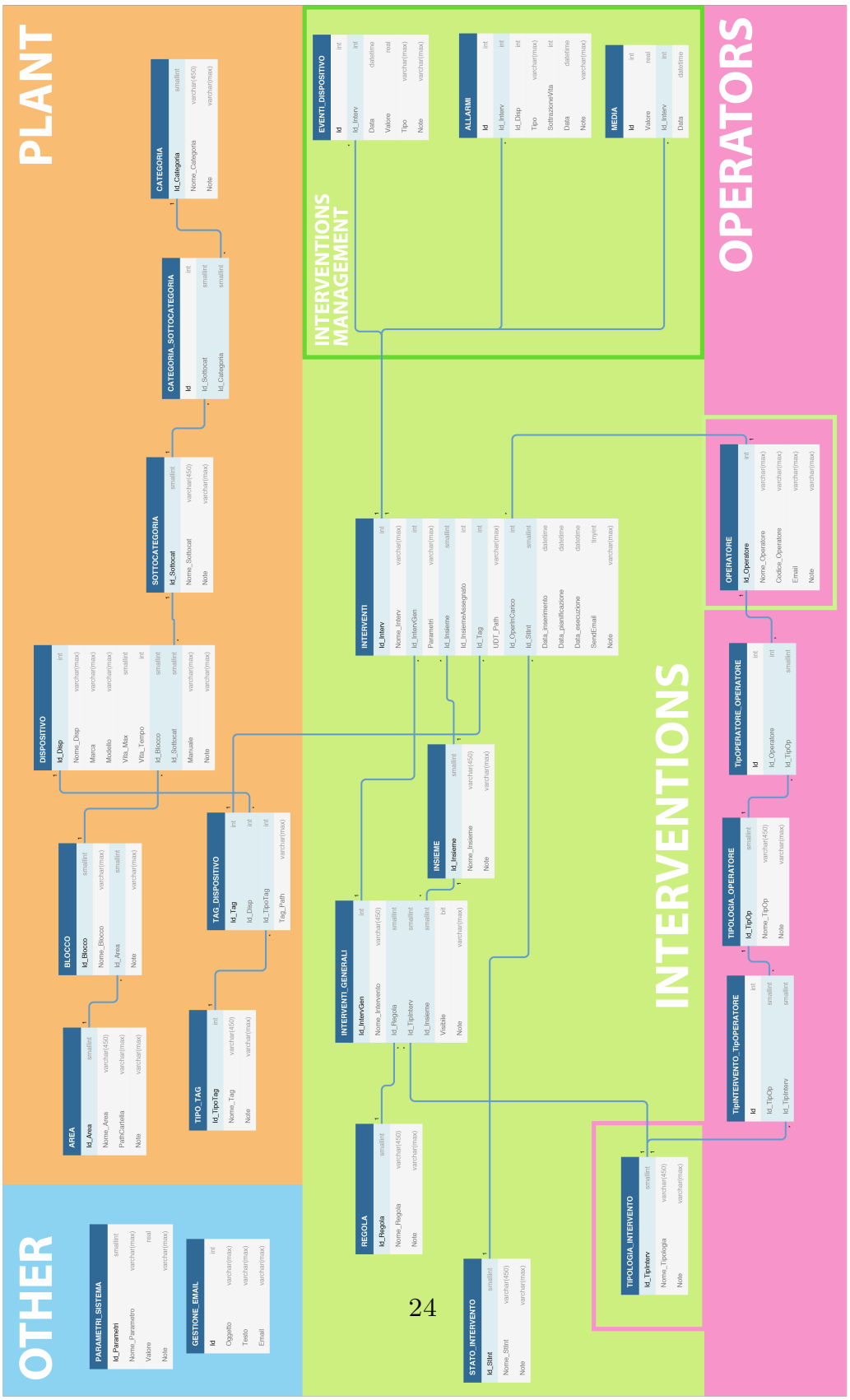

Figure 2.5: Full Database

Chapter 3

Software for database population

Following the guidelines for an effective HMI, an attempt was made to develop the software in the simplest and most intuitive way possible. The pages were mostly structured in the same way with buttons in the same places to be easy to remember. It is tried to make the pages orderly by aligning all the graphic objects. Few colours were used so as not to confuse, grey for the background, red was used to indicate alarms and green for the positive feedback. Pop-up windows were used to indicate errors. The icons used are all commonly used and therefore easy to understand.

The screen is divided in this way: At the top is a fixed navigation bar, immediately below is another bar with the name of the installation, and throughout the remaining space below will alternate the various pages of the software. The general structure can be seen in the Figure 3.2.

The navigation bar appears as in Figure 3.1.

Figure 3.1: Navigation bar

On the right-hand side is information about the logged-in user and the possibility of changing users or exiting the programme.
There are four buttons at the top left. The first one is for returning to the HOME page, the second one opens a page on figure 3.2 from which it is possible to access advanced functions, from there it is possible to change the name of the installation,

enter the system parameters, view the rules for interventions and the status of interventions by editing the explanatory notes. The third one opens a pdf of an instruction manual with an explanation of the software and useful tips that can be consulted by the user. The fourth one shows a flashing red warning triangle, but it is only visible if there is some compilation error in the programme. Clicking on it opens a pop-up with a table showing the list of errors.
Below this is a first bar formed by Tabs that allows you to navigate between the three main sections of the software. Depending on the Tab selected, a second bar appears below it that allows you to navigate between the pages of the chosen section.

Figure 3.2: Advanced settings

The Home page allows navigation through the various parts of the software by selecting between:

- System configuration
- Operators configuration
- Intervention configuration

3.1 Plant configuration

There are two types of pages for the plant part.

An example page for the first structure is shown in the figure 3.3. The pages of the areas, blocks, devices, tag type and device tags, categories and subcategories are structured in this way.

Four zones can be distinguished:

- At the top there is the title.
- At the top left, there is a fillable part in which the plant data are entered. Some fields are writable, others have dropdown boxes from which it is possible to choose an element.
- At the bottom left is a table in which the data already in the database can be consult. The table has filters above it which can be use to filter it.
- On the right-hand side there are buttons: in all pages of this type there are mainly three buttons to manage the data: Add, Edit, Delete. Other secondary keys may be present for shortcuts along the way.
  The Edit and Delete buttons can only be edited if a record is selected in the table.

When a record from the table is selected, all the fields that can be filled in above will automatically fill in with the data from the selected record so that it can be edited. After performing any operation a green square appears at the bottom to give positive feedback on the action. All visualisation tables are made with SQL queries, an example can be seen in the appendix B.

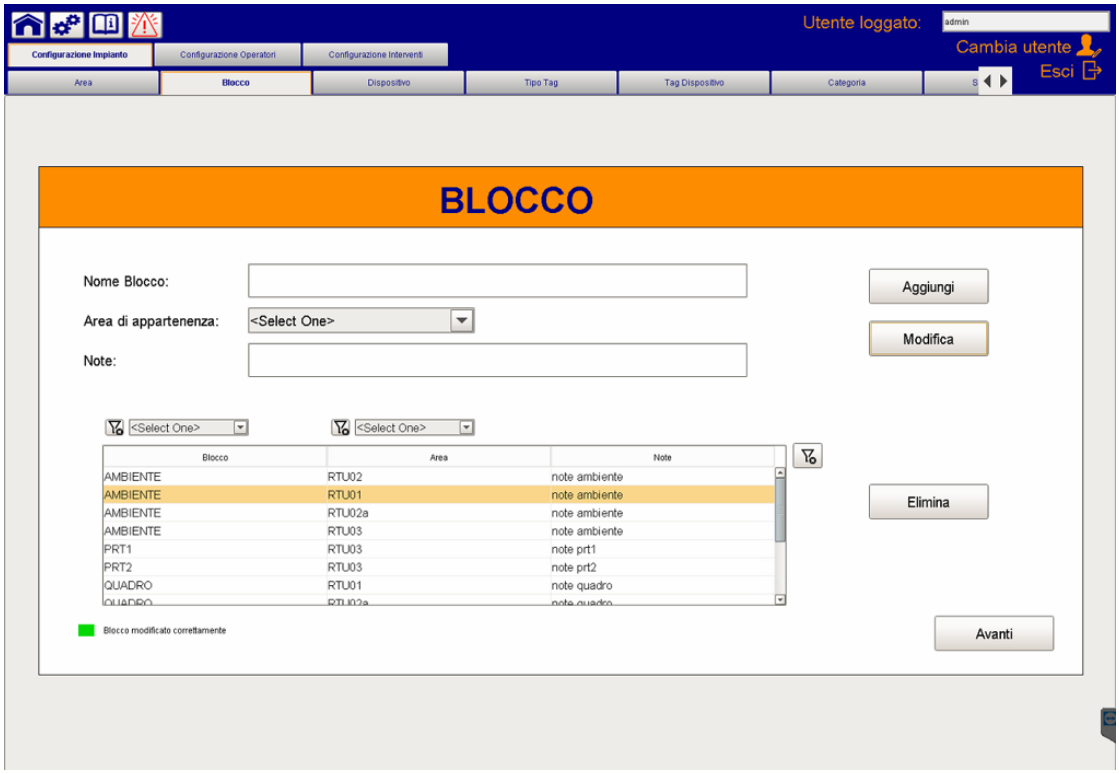

Figure 3.3: Example of a page for entering plant data

The second type of page is structured as in the figure 3.4. It is used to fill in the many-to-many association tables in the database. In this case, it is used for the association between categories and subcategories. Each subcategory can belong to one or more categories. There are four spaces on this page:

- By means of a dropdown list in the first line at the top, it is possible to select the subcategory to which associate the categories .
- There are then two boxes, the one on the left indicates all categories present, the one on the right indicates the chosen categories by selecting from the available categories. There is also a "New category" button that serves as a shortcut to the categories page.
- At the bottom of the page is a table showing the associations already made.
- On the right are the buttons, in this case, there is an Ok, a Delete and a Next button. Selecting the table enables the delete button.

Again, after any operation, a green square appears as positive feedback.

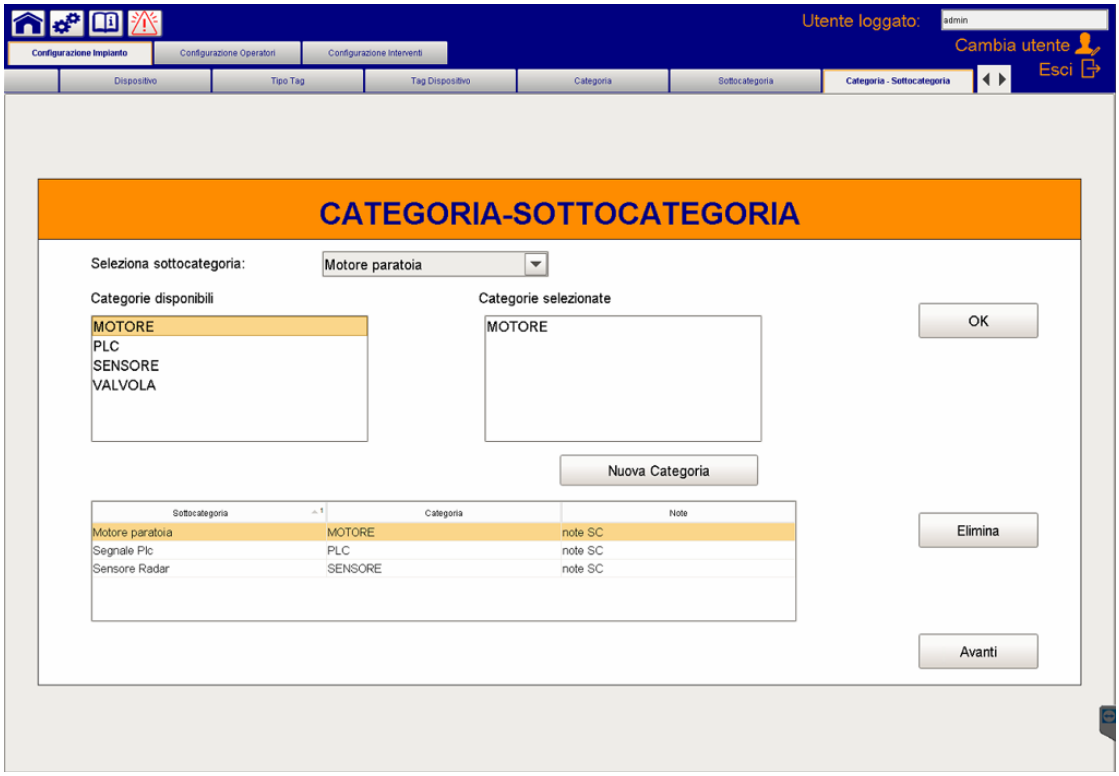

Figure 3.4: Many-to-many category and subcategory association

3.2 Operator configuration

To manage the operators, a preformed Ignition object called "User Management" was used (Figure 3.5) in which both the operator types, called roles, and the operators with their relative data can be inserted. In this section is important insert the e-mail in the space "Contact Info", because through it the operator will be informed about a new interventions and their deadline. For each new operator inserted, it is already possible to make the many-to-many association with the operator types. An operator can be associated with one or more operator types.

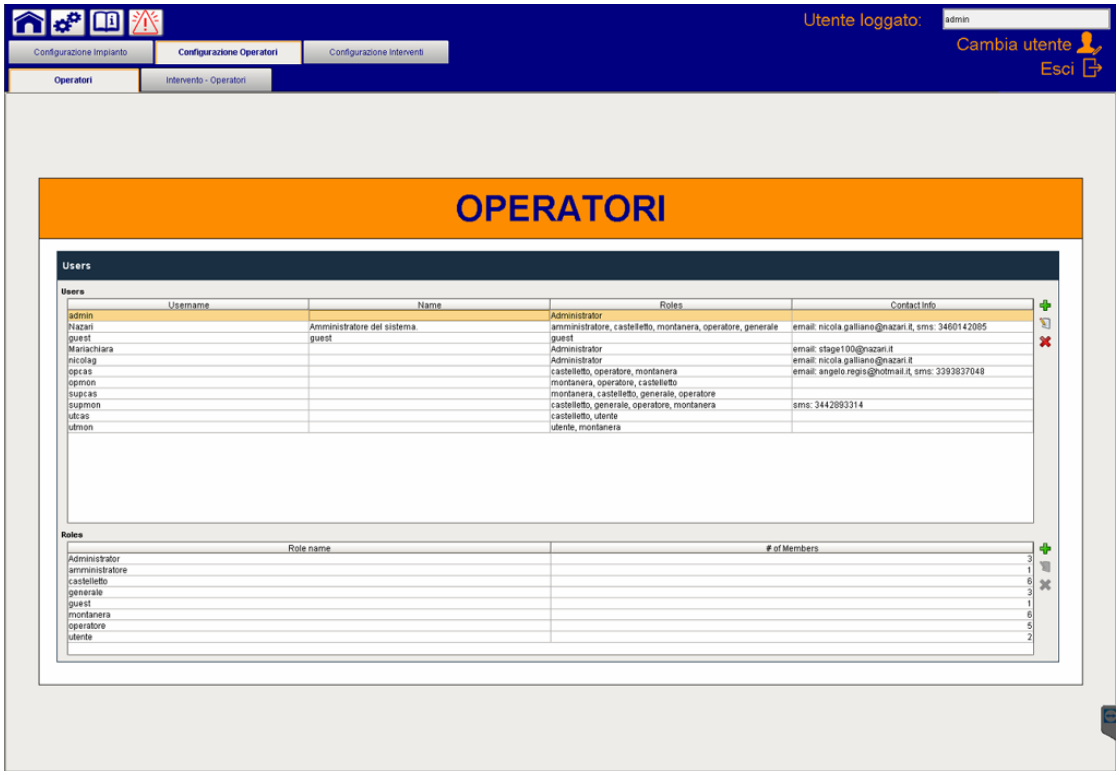

Figure 3.5: Operator configuration

For the association Operator type - Intervention type, on the other hand, a page (Figure 3.6) structured in the same way as that explained in the previous section for categories and subcategories is used. An operator type can be associated with one or more intervention types.

Through this double association, an operator will be associated with one or more intervention types, and will therefore be able to carry out all interventions related to those types.

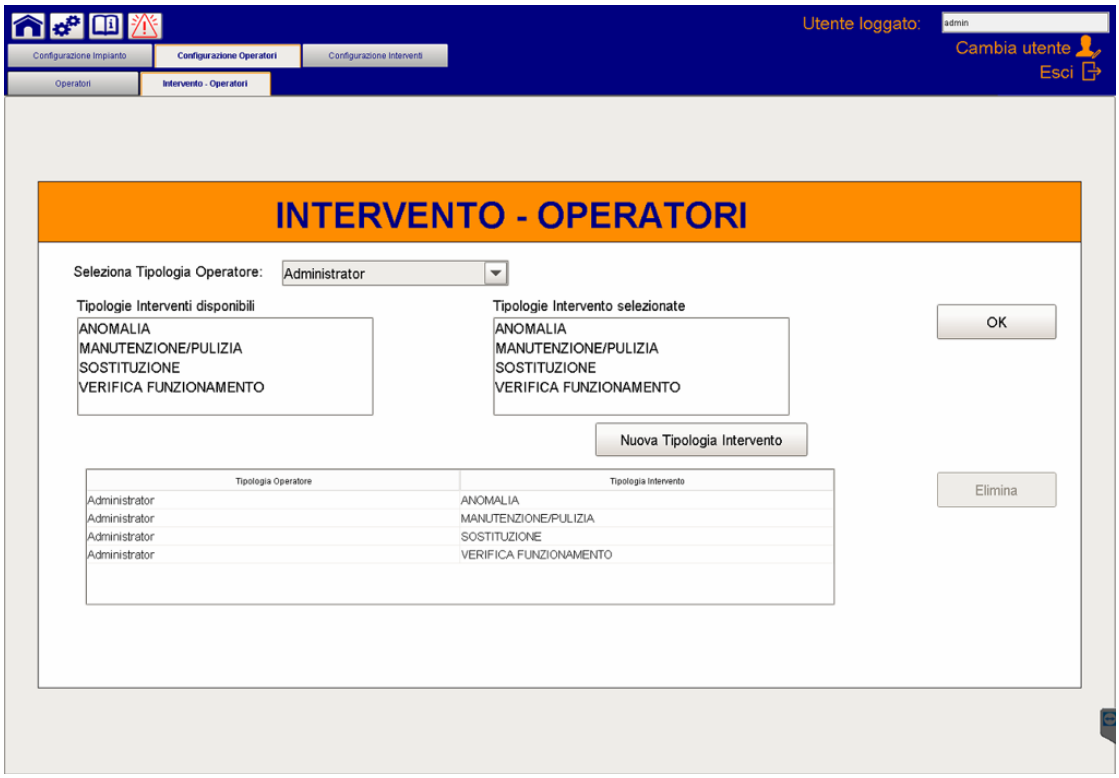

Figure 3.6: Many-to-many operator type and intervention type association

3.3 Intervention configuration

From the intervention configuration section,it is possible to add a new intervention type, general interventions and interventions. The first two pages are structured in the same way as the pages for entering plant data.

When a new general intervention is inserted using the page shown in the Figure 3.7, the rule, the intervention type and the group to which it is associated must be selected using dropdown lists. The checkmark visible or not is used to make the task visible in the operator interface software, even when the task is in the check parameters status. It may be useful to know, for example, that in two months time the grass will have to be cut, even if the intervention is unplanned. According to the choices made in this phase, the page of interventions will change.

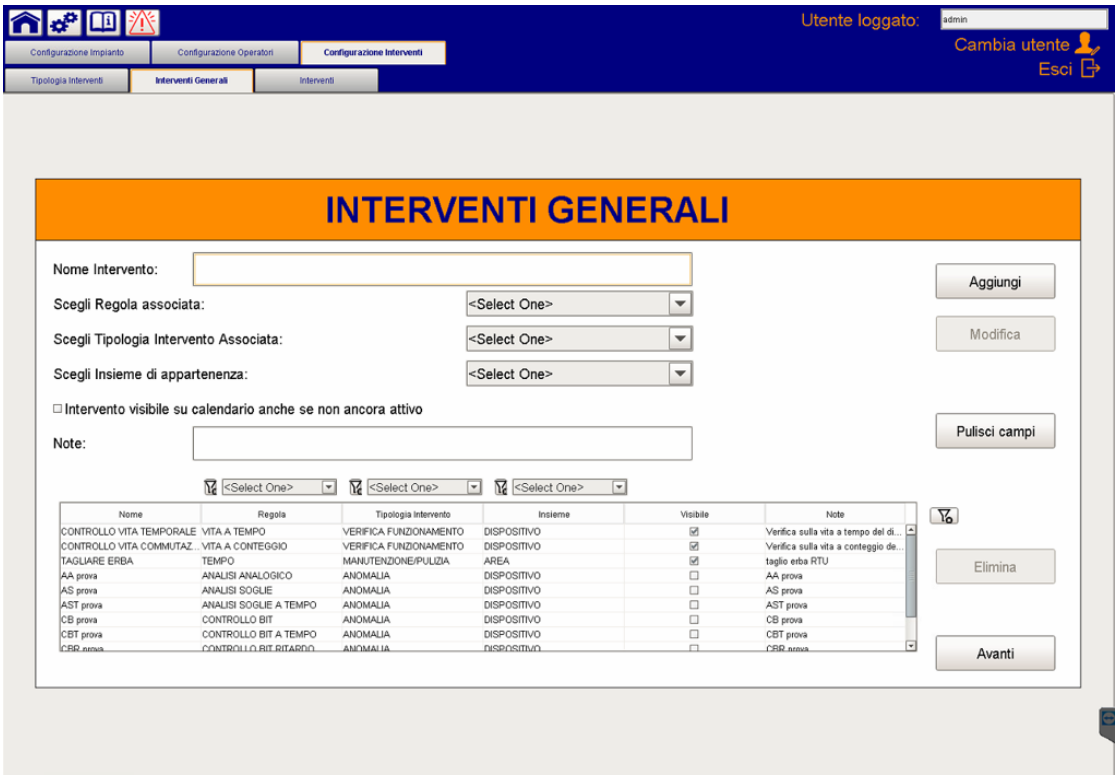

Figure 3.7: General intervention page

The intervention page is by far the most complicated. Each new intervention is associated with a general intervention, consequently it will have parameters to be inserted by the user according to the rule associated with the general intervention and will be associated only with the set defined in the general intervention.

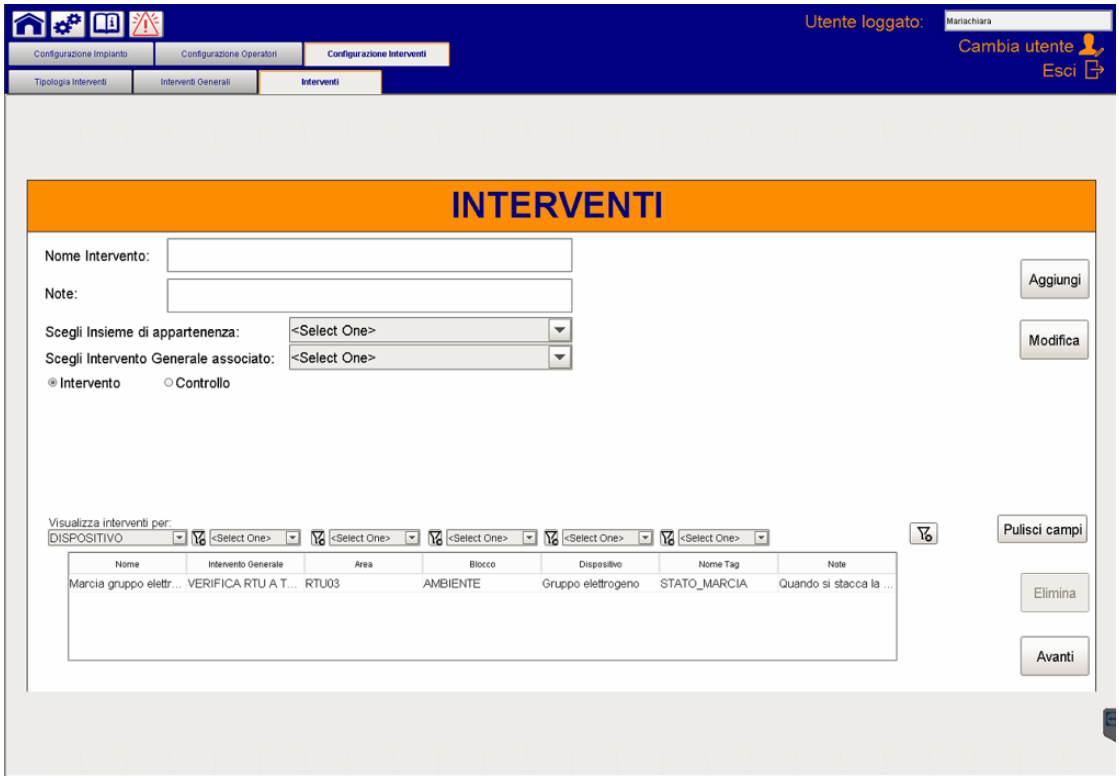

Figure 3.8: Opening intervention page

When the interventions page is opened, it looks like the Figure 3.8. It is possible to see two white spaces, one above the intervention display table and the other on the right, close to the buttons.

A distinction must be made between Control and Intervention:

- An intervention is communicated to the operator when it changes to the schedule.
- Control is an intervention with the intervention status fixed to "CONTROLLO", it is an operation that is done in the background, every time the conditions to satisfy the control are met the alarms table is updated, but an email is never sent to the operator. If, for example, the switchboard overheats above a certain temperature, the components inside may shorten their life as they are not in optimal conditions, but no action can be taken, so there is no point in notifying the operator, but the subtraction of a little life is recorded each time it occurs and will be counted in the component life interventions.

The choice between intervention and control is made through a selection.

By selecting the membership set, the general jobs dropdown list will update to show only the general jobs associated with the set selected and boxes will be displayed in the first white space, through which it is possible to choose where to perform the intervention.
The choices of the set can be:

- Area: only the first box of the first row in figure 3.9 is visible, it contains all the areas of the plant. More than one area can be selected for the same intervention.
- Block: the first two boxes of the first row of the figure 3.9 will be displayed. Initially, it is possible to see all areas and all blocks of the installation. By selecting the area, the blocks box will update displaying only the blocks of the selected area. One or more blocks from the selected area can be selected.
- Device: All the panes in the first row of the Figure 3.9 will be displayed, with all the areas, blocks and devices and type tags in the installation. Selecting an area and a block will only display the devices in the block. One or more devices can be selected, but only one tag at a time. If multiple devices are selected, only tags with the same type in common between the devices will be displayed. If an operation that does not have a tag attached is selected, the last pane will disappear.
- Category: The first box of the second row in the Figure 3.9 is visible. Only one category is selectable.
- Subcategory: Both boxes of the second row in the Figure 3.9 are visible. Once a category has been selected, only the associated subcategories are displayed. Only one sub-category can be selected.

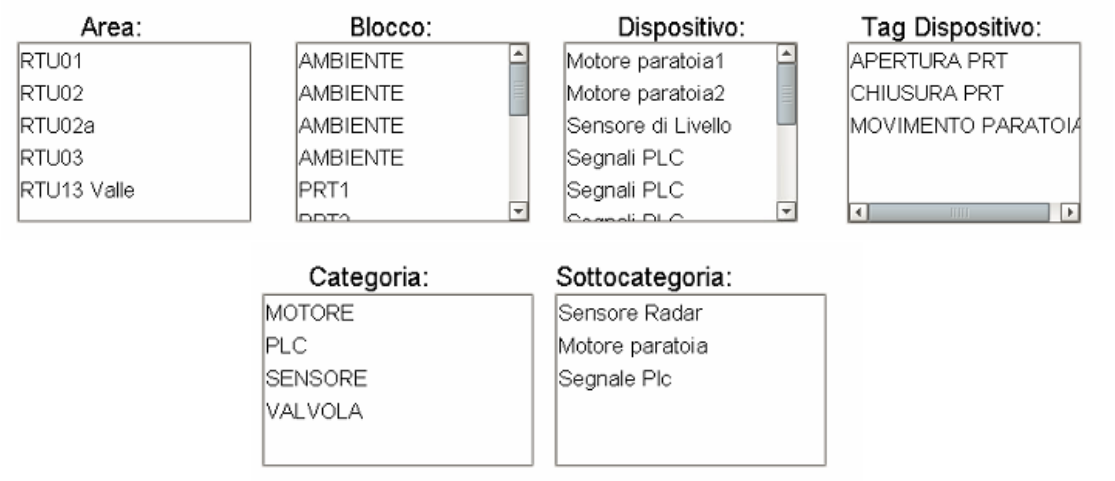

Figure 3.9: Select location

Placing an action on a category or sub-category means placing an action on all devices that are part of that category or sub-category. Only general jobs without linked tags will be available, i.e. time-based jobs.
Once the general task has been selected, a window with the parameters of the rule associated with the general task will appear in the white space next to it.

An example of a complete page is shown in the figure 3.10. A full explanation of the rules and their parameters can be seen in the appendix A.

The visualization table changes according to the set of the chosen intervention, there is a filter above with which it is possible to choose which table visualize. There are also filters above the columns of the table to filter the display of the table.

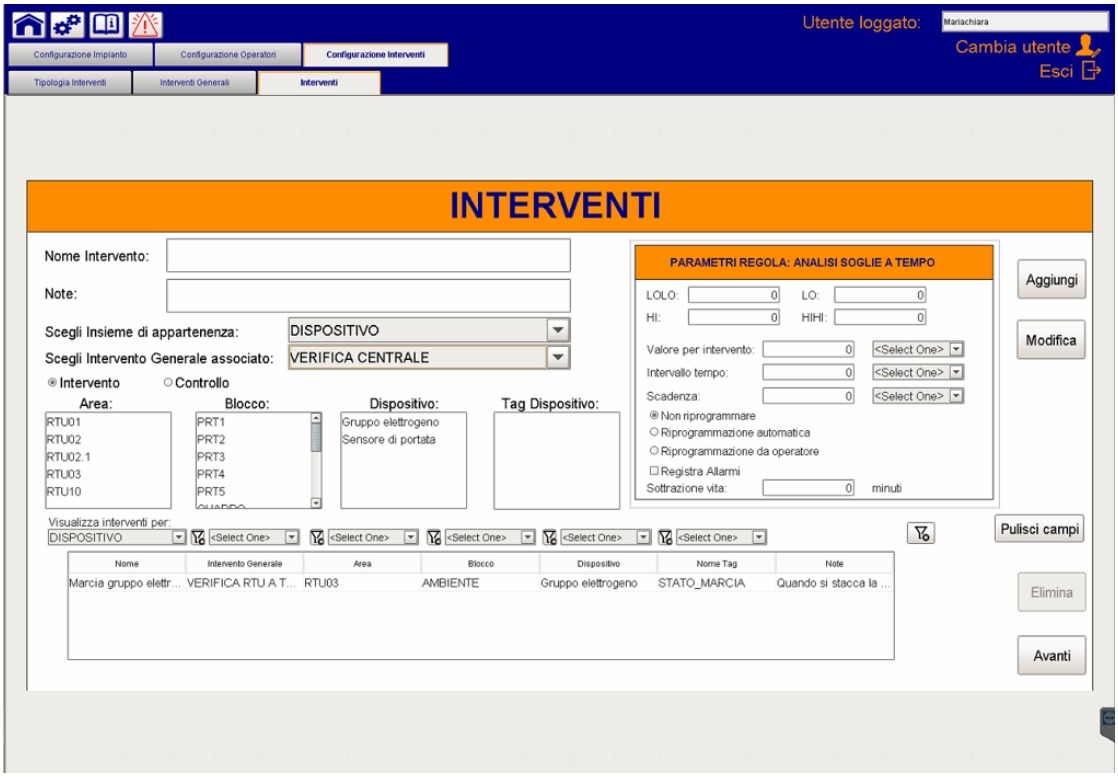

Figure 3.10: Complete interventions page

Chapter 4

Operator interface

For this part of the software, it is tried to keep the same graphical appearance as the previous software, but simplifying some steps as it will be used by people with a low level of education and with little confidence in the technology.
When the software is started, the Home page will open (like in Figure 4.1), allowing navigation through the three main parts of the software: system consultation, intervention management and consultation of the alarm history.
Also in this case there will be a horizontal bar fixed at the top, starting from the right it is possible to find:

- The Home button to return to the main page.
- A button to open the PDF of the instruction manual.
- A flashing red button that opens a PopUp with the interventions expiring (in white) and expired (in red). The button is visible only if there are overdue or expired jobs.
- In the centre there is the name of the installation.
- To the left is the section showing the logged in user, and two shortcuts to change the logged in user and exit the software.

Figure 4.1: Opening page

4.1 Consultation Plant

Through this page (Figure 4.2), it is possible to see all the elements of the system. In order to view the plant, the set to be viewed must choose using the dropdown lists. Depending on the choice, a series of boxes will appear below to filter the table.
Each box is structured in the same way: It has a checklist to be ticked with the word "All" and underneath a box with the list of all elements. To view all of them, check "All", otherwise select one or more records from the box to filter the table. In the case of several boxes in a row, the selection of the first box influences the display of the next one. The area selection affects the block display, the block selection affects the device display, the category selection affects the subcategory display.

Figure 4.2: Browse Plant

Depending on the chosen membership set it is possible to see different panes. Selecting:

- Area: Only Area is displayed.
- Block: Area and then Block are displayed.
- Device: Area, block, device, category and subcategory are displayed.
- Category: Category, Area and Block are displayed.
- Subcategory: Category, Subcategory, Area and Block are displayed.

The table below is then filtered according to the choices made above. Next to the filters, there is a button to remove all filters.
In the case of displaying devices, selecting a record in the table will enable the two buttons next to it. One is used to open the manual or datasheet of the device if inserted during system configuration. The other opens a graph with the trend over time of the plc tags connected to the device, if present.

4.2 Management of interventions

This page allows the operator to view and manage their interventions. Using the dropdown list at the top right it is possible to choose whether to view planned, ongoing or all interventions. Depending on the logged-in user, only the planned interventions that he/she can take over, or only the ongoing interventions that he/she has already taken over, will appear. In this type of page there is an equal level of filters for all three options: it is possible filter by intervention Id, or by area, block, device, category and subcategory via a dropdown list.

Figure 4.3: Filter

Let us study the three possible views separately.

PLANNED INTERVENTIONS
In this table, all the tasks with a task status of "PIANIFICATI" or those with a task status of "CHECK PARAMETRI" but which had checked "Visible" when creating the general task are visible. An additional filter was then added to select the task status.
Once an intervention has been selected in the table, the part below is filled in with the intervention data, the notes, the reason why the intervention was triggered and the expiry date. The "Prendi in carico" button is also enabled, allowing the operator to take over the intervention. Only planned interventions can be taken over, the status of the intervention will then change to "IN CORSO". Once the operation is taken over, a feedback message with a green square appears to indicate that the operation was successful. Selecting an intervention on a device also enables two other buttons. One is used to open the datasheet/manual of the device, if present. The other displays the time trend of the PLC Tag connected to the trip, if present.

Figure 4.4: Planned Interventions

ONGOING INTERVENTIONS

Once an operation has been selected, three buttons are enabled:

- "Lascia": to leave an operation that has been taken over.
- "Risolto": If the intervention has been successfully concluded.
- "Non Risolto": If the task has been completed but the problem has not been solved. In this case, it is mandatory to write the reason in the notes.

After any of the three operations, the usual positive feedback message will appear. In addition, selecting an intervention will automatically fill in the intervention data in the part below, and from there the date of the intervention and the notes on the intervention can be set. After completing a task, if the "AutoRiprogrammazione" parameter was selected in the rules, the task will be reprogrammed automatically, if the "RiprogrammaOperatore" parameter was selected, a popup will appear, allowing the operator to reset the task parameters if he considers it necessary. Also in this case, for an intervention on a device, it is possible to display the datasheet, if present, and the trend in time of the tag plc connected to the intervention, if present.

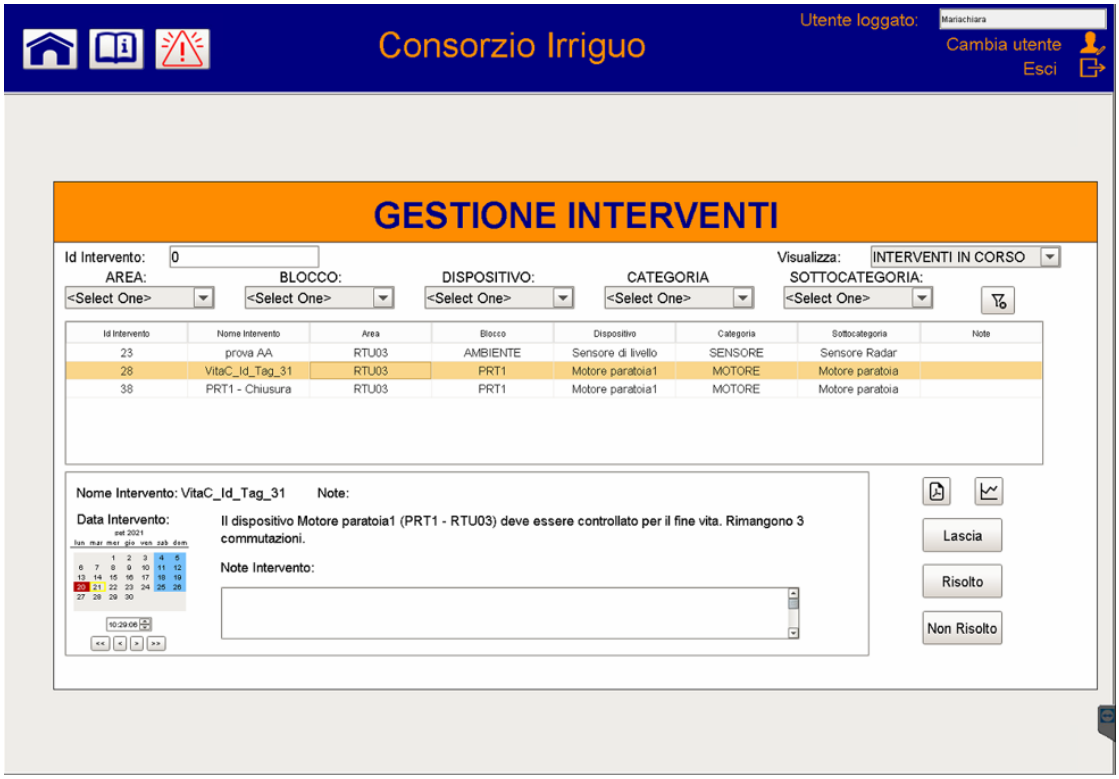

Figure 4.5: Ongoing Interventions

ALL INTERVENTIONS

On this page, it is possible to see all the interventions. The table can also be filtered by intervention status and operator code. It can also be filtered by date using a graphical ignition object called Data Range.

Figure 4.6: All Interventions

4.3 Alarm display

Figure 4.7: Alarms

On this page, it is possible to see all the alarms linked to the intervention. Every time that an intervention pass to planned, a record in the table be written. Moreover, in the rule configurations, there is a parameter called "Registra allarmi" which, if selected, continues to save alarms even after an intervention has been triggered until it ended. Whenever the conditions that would have triggered the intervention reoccur a record be written, but without sending the emails. The filters in the table are the same as those explained in section 4.1. It is also possible filter by date. Once an alarm has been selected, if the tag linked to it is historicised, clicking on the graph button will show the trend over time, as in the Figure 4.8.

Figure 4.8: Development over time

Chapter 5

Testing

The software has been tested on the real plant to prove that it works properly. The first test on the system is done on the tank level of the RTU10. The water in the basin is turbines by the hydroelectric plant, but it is only active if the basin has a certain water level; if the level drops below a critical level, the plant stops. For proper functioning of the power plant, it should be active for a continuous period and not stop repeatedly in a short time. An intervention is therefore created in connection with a general intervention "VERIFICA CENTRALE" which is connected to the "ANALISI SOGLIE TEMPO" rule which checks if the level of the tank falls below the critical value (2.925 m) five times in 48 hours. The creation of the intervention can be seen in Figure 5.1.

Figure 5.1: Creation of tank level check intervention

The intervention is created on 22/09/2021 at 7:57 AM, goes into planning on 1/10/2021 at 4.55 AM. Below is the table "EVENTI_DISPOSITIVO.

Id	Id_Interv	Data	Valore	Tipo
1	9	2021-09-24 05:26:22	2.895543	LO
2	9	2021-09-26 22:15:32	2.905484	LO
3	9	2021-09-29 08:36:49	2.924222	LO
4	9	2021-09-29 14:50:36	2.92151	LO
5	9	2021-09-30 23:19:15	2.907765	LO
6	9	2021-10-1 02:02:05	2.2881091	LO
7	9	2021-10-1 04:55:57	2.897005	LO

The first two records are not taken into account because the difference in time between the first and the next five, and between the second and the next five is greater than 48 hours.
Below is the time course of the tag calls "LIVELLO" during the period when the critical levels occurred.

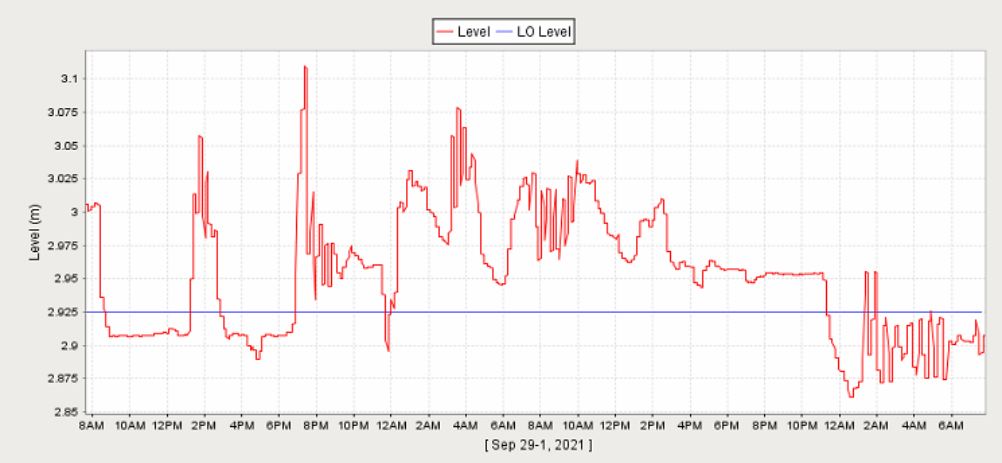

Figure 5.2: Development of tank level over time

The Figure 5.3 shows as an example the completion of the intervention management page.

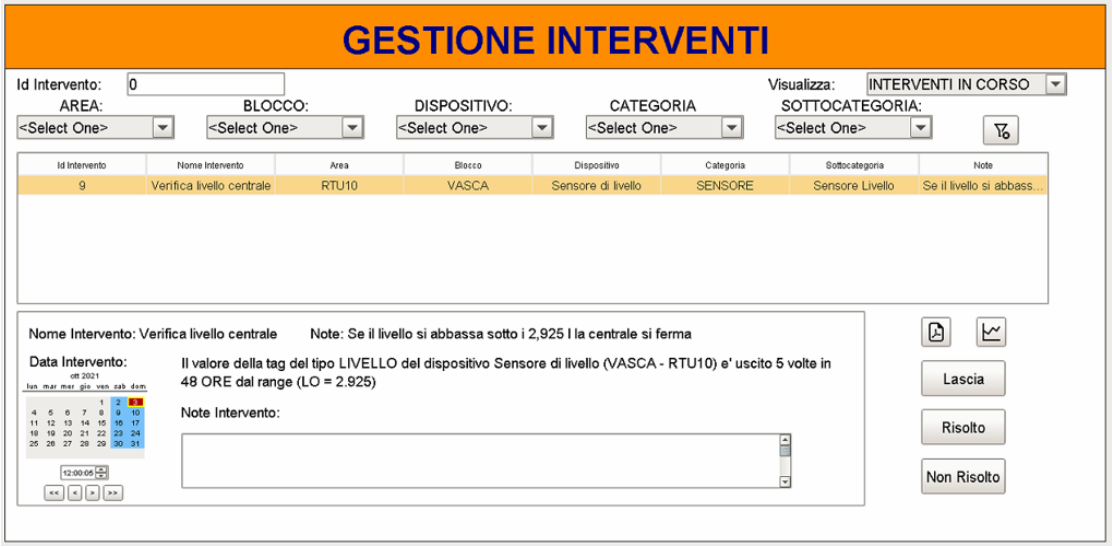

Figure 5.3: Conclusion of the intervention

The time line of the intervention is shown in the Figure 5.4 and the table "INTERVENTI" filled with the relevant data can be seen in the Figure 5.12.

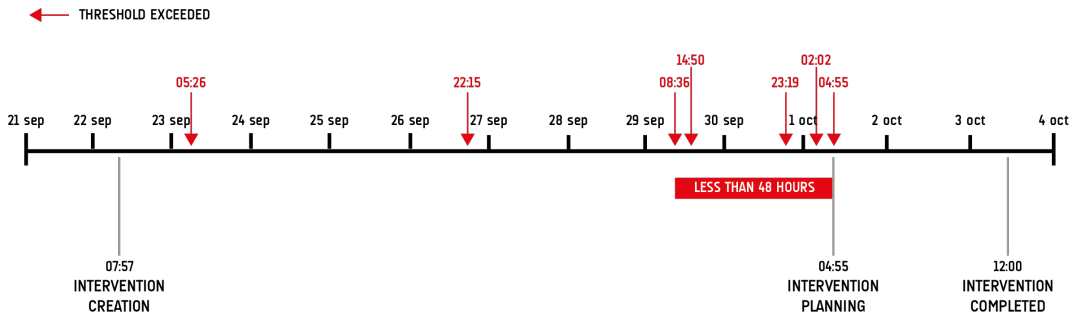

Figure 5.4: Time Line

Another test is made on the running signal of the genset, which is activated when the power fails. If the current goes out too often in a short time it may be a sign of a bigger problem. The general Intervention has been joined to "CONTROLLO BIT A TEMPO" rule. The intervention is created as shown in the Figure 5.5, it is checked if the tag changes to 1 three times in 24 hours. In this case the parameter "RegistraAllarmi" is checked, so the control will continue even after the intervention is scheduled. Once the intervention has ended, the same intervention will be automatically rescheduled.

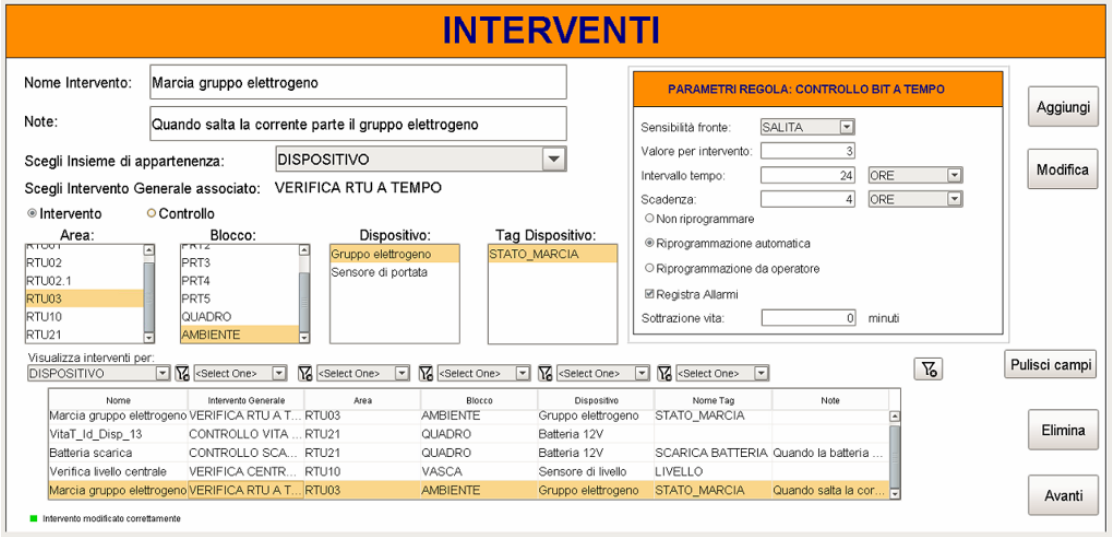

Figure 5.5: Creation of a genset running status intervention

The intervention is created on 30/8/2021 at 3:22 PM, goes into planning on 30/9/2021 at 5:30 PM. Below is the table "EVENTI_DISPOSITIVO".

Id	Id_Interv	Data	Valore	Tipo
1	1	2021-09-29 13:59:32	1	STATO_MARCIA
2	1	2021-09-29 17:58:49	1	STATO_MARCIA
3	1	2021-09-30 14:30:44	1	STATO_MARCIA
4	1	2021-09-30 17:30:36	1	STATO_MARCIA
5	1	2021-10-1 19:15:15	1	STATO_MARCIA
6	1	2021-10-1 11:32:05	1	STATO_MARCIA
7	1	2021-10-1 16:35:57	1	STATO_MARCIA
8	1	2021-10-1 17:15:57	1	STATO_MARCIA

The first record are not taken into account because the difference in time between the first and the next three is greater than 24 hours. In the period shown in the graph in figure 5.6 , it can be seen that after the first three tag changes occurred in less than 24 hours, the condition to meet the intervention requirements occurs again. Below there is the time course of the tag calls "STATO_MARCIA".

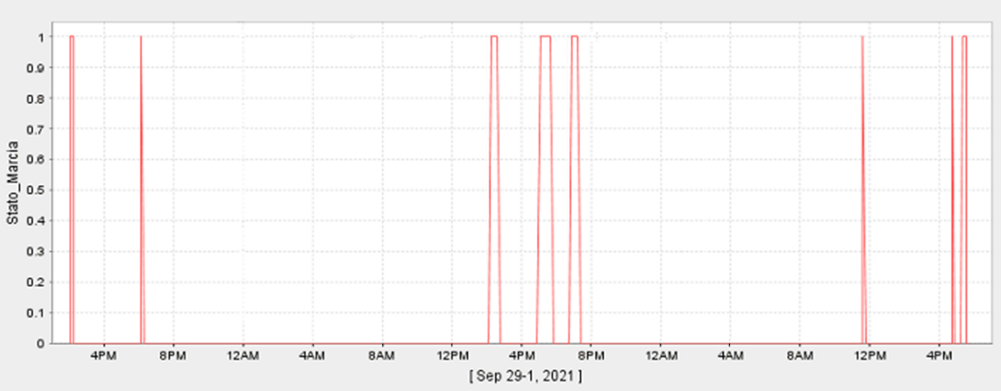

Figure 5.6: Switching the UPS connection bit over time

Having checked the "RegistraAllarmi" parameter, two records will appear in the Alarm Table, as can be seen in the figure 5.7.

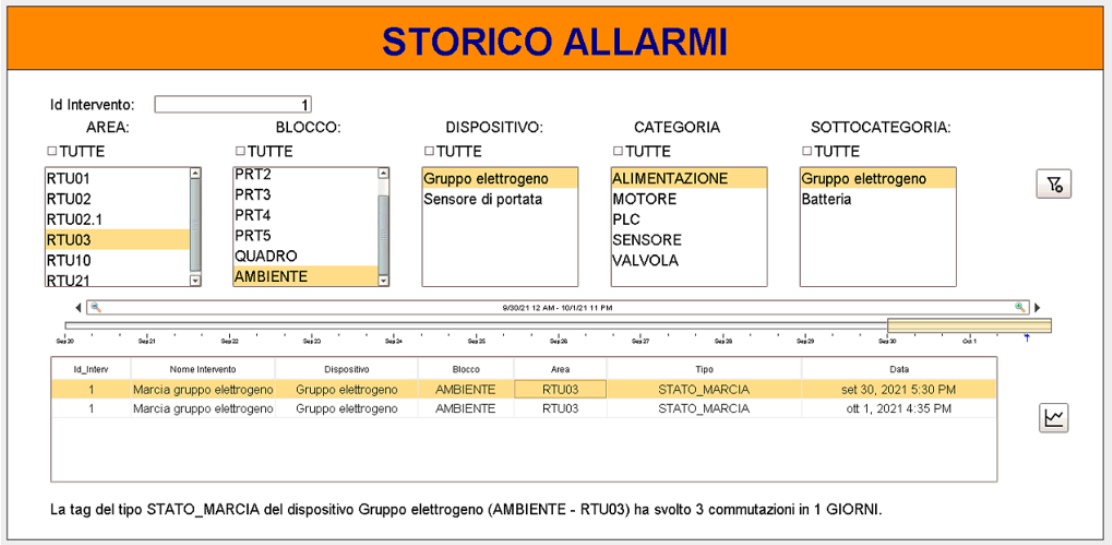

Figure 5.7: Table ALLARMI

The table "INTERVENTI" filled with the relevant data can be seen in the Figure 5.12.

The last test was on the life of the batteries. In every switchboard, some batteries intervene in the event of a power failure. The average life of the batteries indicated by the seller is two years. In ideal conditions, the battery should be used without ever being completely discharged. Each time the battery is completely discharged, 48 hours are subtracted from the total life of the device.
When the device is inserted, see Figure 5.8, a pop-up automatically opens to enter the parameters of the operation, see Figure 5.9.

The battery was installed on 12 November 2019, it is intended that the intervention should be notified one month before the end of life of the device and the batteries must be replaced within seven days of the occurrence.

Figure 5.8: Device configuration

Figure 5.9: Creation of an intervention on the time life of the device

A tag called "SCARICA BATTERIA" is connected to the device, which switches to one whenever a low battery charge is detected. A control connected to the "CONTROLLO BIT" rule is inserted on this tag, see Figure 5.10. Each time the tag switches to one, a record is recorded in the alarm table that records a life subtraction of 48 hours, i.e. 2880 minutes. As this is control, no email will be sent to the operator to indicate battery discharge.

Figure 5.10: Creation of a battery discharge check

Tag data prior to software development is retrieved from the system history and the alarm table for this check is shown below.

Id	Id_Interv	SottrazioneVita (min)	Data
1	1	2880	2019-11-28 13:59:32
2	1	2880	2019-12-4 17:58:49
3	1	2880	2020-03-30 14:30:44
4	1	2880	2020-07-22 17:30:36
5	1	2880	2020-3-1 19:15:15
6	1	2880	2020-6-3 11:32:05
7	1	2880	2021-1-5 16:35:57

The tag is changed to one seven times, so 14 days are subtracted from the total life of the device.

The intervention is passed in planned on 28 September 2021, see Timeline in Figure 5.11.

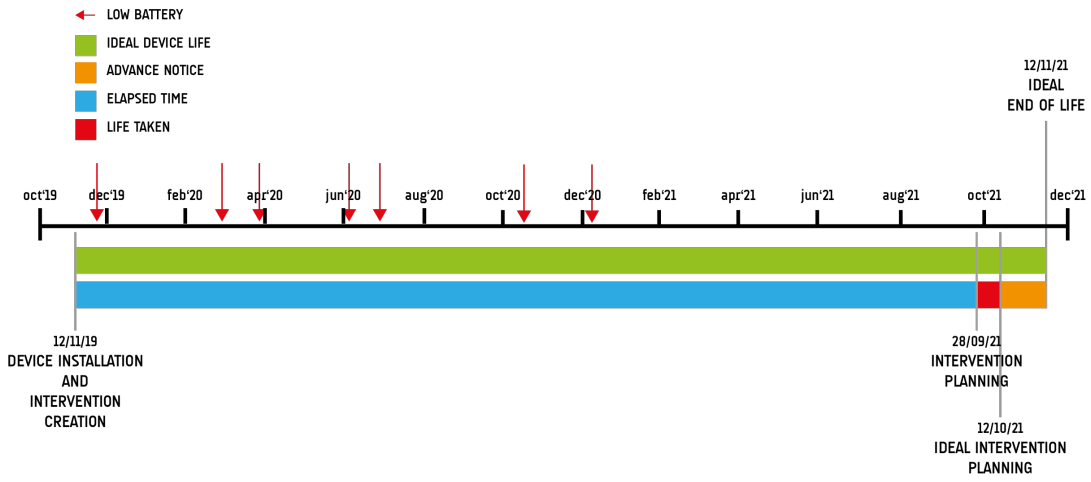

Figure 5.11: Time Line

Below is a table of the interventions listed above with the most relevant data.

Id_Interv	Intervento	Int_Generale	Dispositivo	Blocco	Area	Data_inserimento	Data_pianificazione	Data_esecuzione
1	Marcia gruppo elettrogeno	VERIFICA RTU A TEMPO	Gruppo elettrogeno	AMBIENTE	RTU03	ago 30, 2021 3:22 PM	set 30, 2021 5:30 PM	ott 2, 2021 8:01 AM
4	VitaT_Id_Disp_13	CONTROLLO VITA TEMPORALE	Batteria 12V	QUADRO	RTU21	nov 12, 2019 12:40 PM	set 28, 2021 12:40 PM	ott 3, 2021 4:02 PM
5	Batteria scarica	CONTROLLO SCARICA TOTALE BATTERIA	Batteria 12V	QUADRO	RTU21	nov 12, 2019 1:00 PM		
9	Verifica livello centrale	VERIFICA CENTRALE	Sensore di livello	VASCA	RTU10	set 22, 2021 7:57 AM	ott 1, 2021 4:55 AM	ott 3, 2021 12:00 PM

Figure 5.12: Table INTERVENTI

Conclusion

In this thesis, software for preventive maintenance of an automation system for water transport was studied and developed through Ignition. At the basis of the software is a database that has been implemented in Microsoft SQL Server.
In order to implement the graphical interface, existing graphical objects of Ignition that can be customised using the Python language were used. The graphical user interface of the software was developed in such a way that it was simple and intuitive, using colours so that important things stood out. The software was divided into two parts, one for the population of the database, and one for the management of interventions by the operator. The database was divided into three macroblocks, one for entering plant data, one for managing the operators and their tasks, and the last one for scheduling interventions and managing the logic governing them. The logic is based on three concepts: the passage of time, the switching of a bit, or the analogue trend of a measurement. A total of nine logics have been implemented to solve all the problems presented by the customer. The system's requirements related primarily to the periodic scheduling of interventions for cleaning the areas and lubricating the components, detecting abnormal measurement values and replacing components before the end of life.

There are several jobs that could still be implemented in the future. Examples include the following:

- New logic can be added in the future according to customer needs.
- There are plans to use in the future the study behind the project and the software, adapting it and creating new logic behind the interventions, for other automation plants with the SCADA developed in Ignition. The best thing would have been to implement the graphic interface on Visual studio in a language such as C# to have a software adaptable to any plant and not only those with an Ignition SCADA, and customisable at will, but the time available for this thesis would not have been necessary.
- There are a number of interventions that need to be monitored but for which it is not possible at the moment to resolve with any action, they are called

controls. Adding elements to the plant could resolve these events as well as monitor them. For example, a persistent problem is a high temperature to which devices are subjected in switchboards. Being placed in the middle of nature, in summer high temperatures are reached that shorten the life of the components inside. The temperature trend is monitored to adapt to the life of the components, but for now, it is not possible in any way to limit the raising of it. One way to solve the problem would be the installation of fans within the framework. In that case, interventions could be planned that detect the excessive temperature and activate the fans accordingly.

- It is possible to add an additional section of the software that manages the pieces in stock and orders of them according to their need them and the delivery time so you do not find yourself without them at the time of the change.
- Another improvement could be the automation of the creation of interventions based on triggers such as the passage of time or the attainment of certain states.

It is believed that the software developed during this thesis work will significantly improve the condition of the plant by helping with its maintenance. In the coming months, the software will be tested by the customer and based on the feedback received, modifications can be made to make the software even more efficient.

Appendix A

Rules: Parameters and algorithms

The logics underlying the interventions can be divided into three macro-categories:

- Temporal Analysis
- Digital Tags
- Analogue tags

Each rule has a Datatype attached, with parameters useful for control.
The algorithm that controls the intervention is executed either when the value of the plc's Tag changes or when a supporting variable called "conteggio" changes, it is incremented once per second. Some parameters are repeated the same in all interventions:

- "Scadenza": indicates how long from the planning date the intervention must take place.
- "Non Riprogrammare/Riprogrammazione Automatica/Riprogrammazione da Operatore": this is used to set whether or not, once the operation has been completed, it can be rescheduled, and if so, automatically or by the operator who carried out the operation.
- "Sottrazione vita": is visible only if the intervention is connected to a device that has a life other than 0.
- "Registra Allarmi": If selected, every time the check on the rule would trigger the intervention, it is written in the table "ALLARMI". In this way, the history of the parameter to be checked can be kept even after the intervention

has been scheduled. this is interrupted once the intervention is completed ("RISOLTO/NON RISOLTO"). If the intervention is only a control, the parameter is set to 1 by default

The user-settable parameters for each rule and the control algorithm are listed below.

TEMPORAL ANALYSIS

- TEMPO

Parameters	**DataType**
IntervalloTempo	int4
Preavviso	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean

The time rule is used for jobs that are to be scheduled after a fixed period ("IntervalloTempo"). The "Preavviso" parameter indicates how long before the time interval expires you want to schedule the intervention. This rule can be associated with an area, block or device. The algorithm is explained in the Figure A.1.

- VITA A TEMPO

Parameters	**DataType**
Preavviso	int4
Scadenza	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean

The intervention linked to this rule is created by default when a device with a time life other than 0 is inserted. "Preavviso" indicates how long before the end of the device's life ("vitaT") the intervention should be scheduled. The algorithm is explained in the Figure A.2.

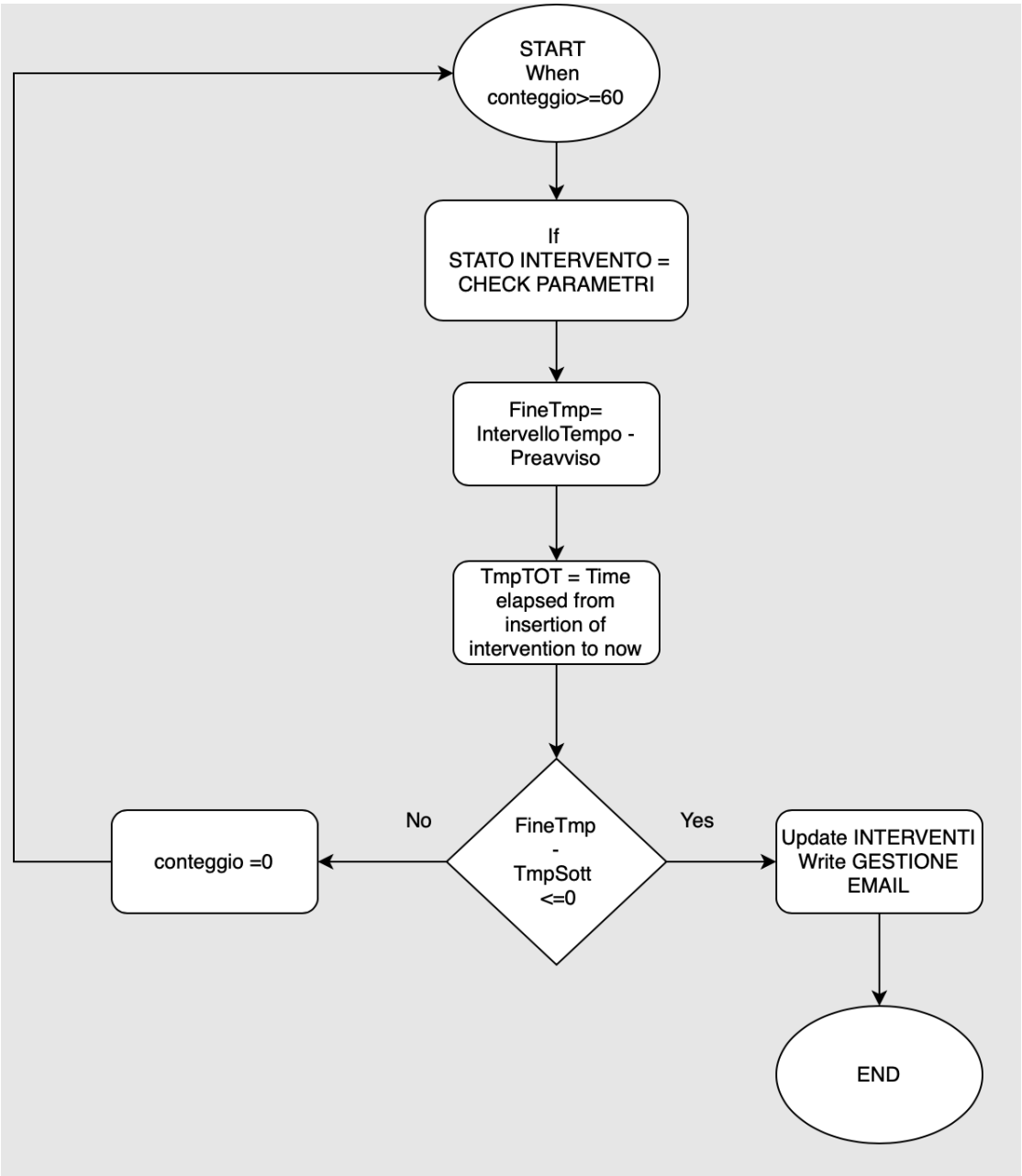

Figure A.1: Algorithm for the rule:"TEMPO"

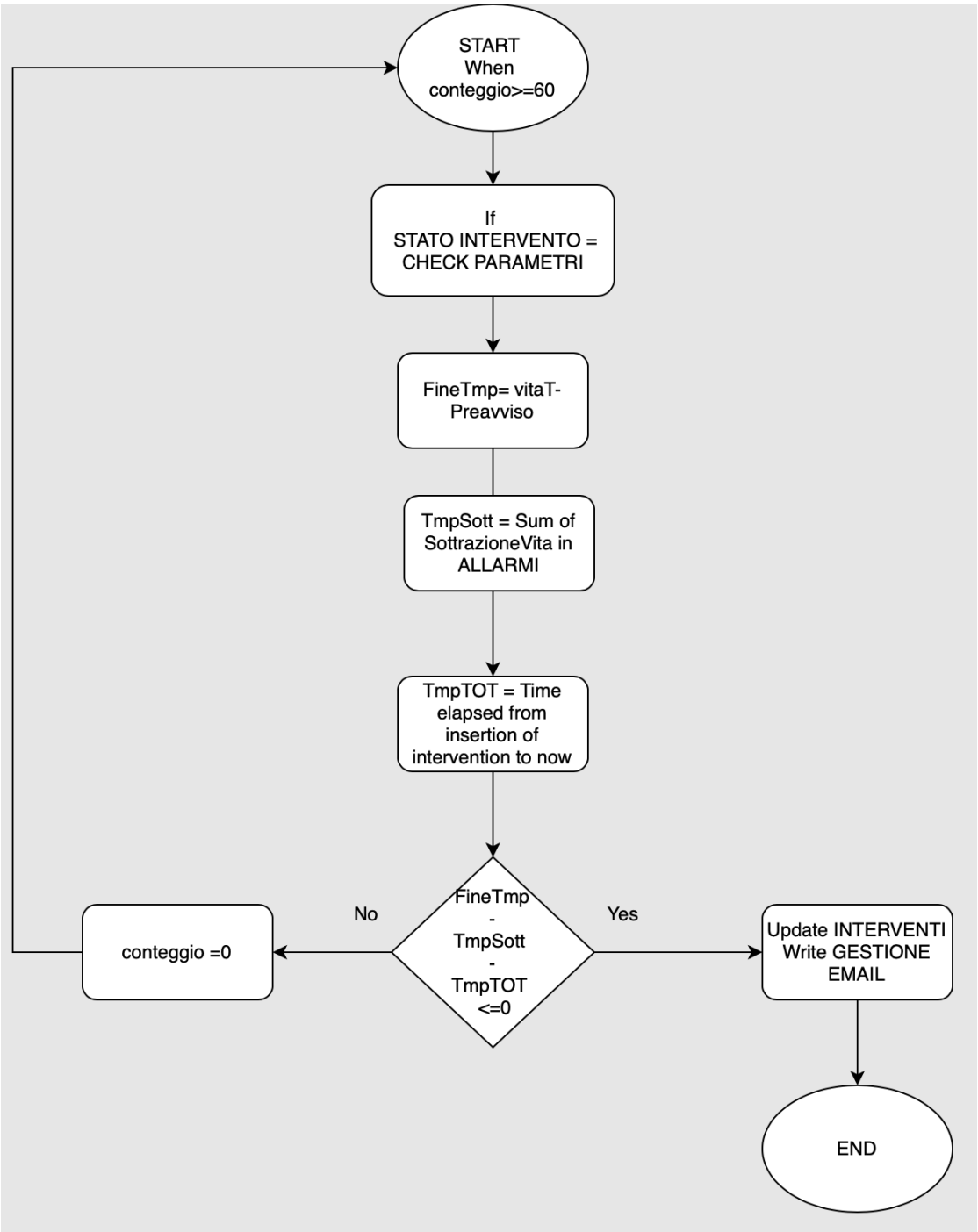

Figure A.2: Algorithm for the rule:"VITA A TEMPO"

DIGITAL TAGS

- VITA A CONTEGGIO

Parameters	**DataType**
Preavviso	int4
Scadenza	int4
OperatoreRiprogrammazione	Boolean

The intervention linked to this rule is created by default when a tag of a device with a life count different from 0 is inserted and check the box indicating that the selected Tag subtracts life("vitaC") from the device (Selectable for only one Tag per device). "Preavviso" indicates how long before the end of the device's life it should be scheduled. The algorithm is explained in the Figure A.3.

- CONTROLLO BIT

Parameters	**DataType**
SensFronte	Boolean
Limite	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean
RegistraAllarmi	Boolean

All rules of the type "CONTROL BIT .." have a parameter called "SensFronte": it decides whether to count when the bit switches to 1 (UP) or 0 (DOWN) and another called "Limite": Indicates how many times after the bit switches the trip should be scheduled. The algorithm is explained in the Figure A.4.

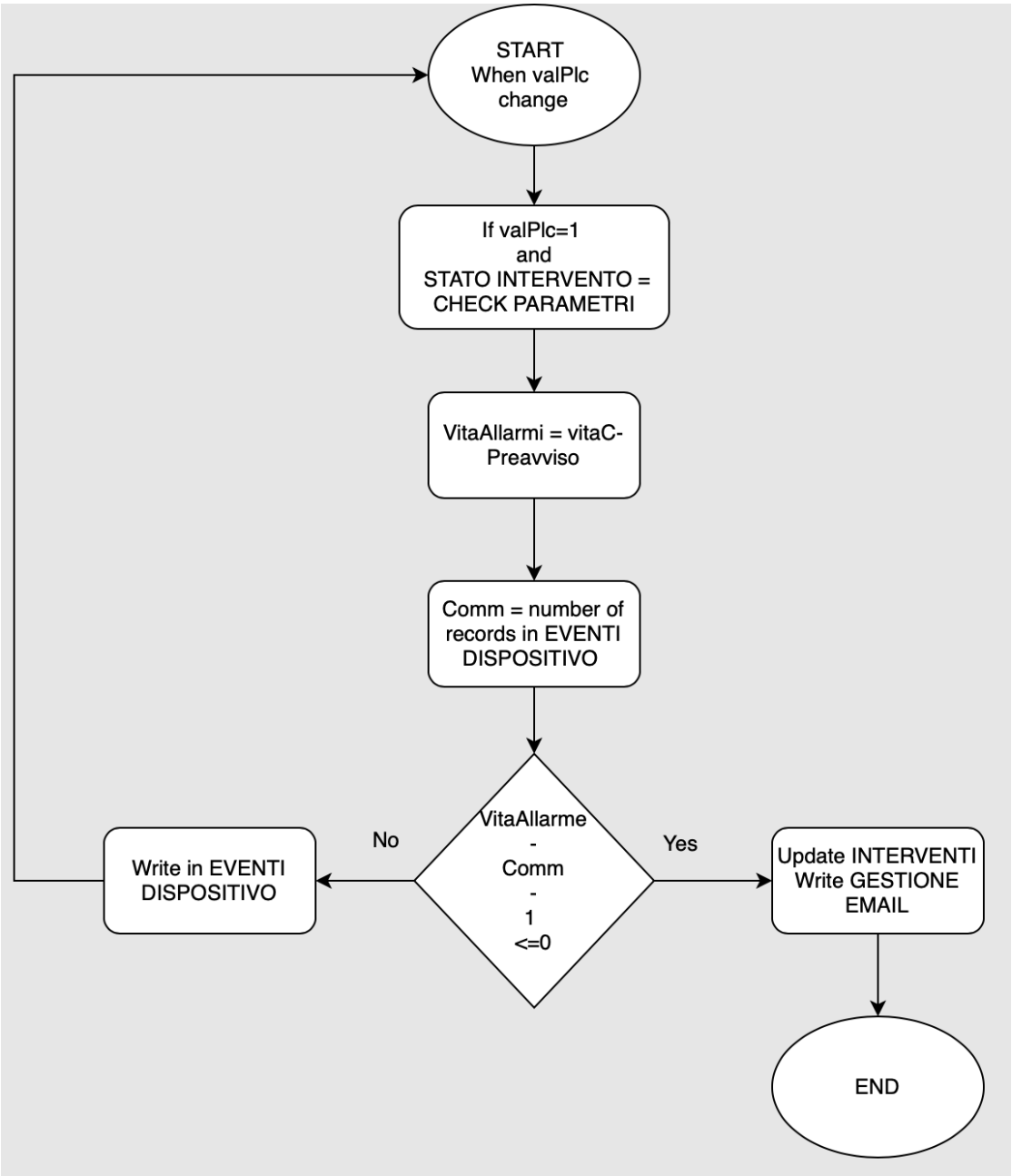

Figure A.3: Algorithm for the rule:"VITA A CONTEGGIO"

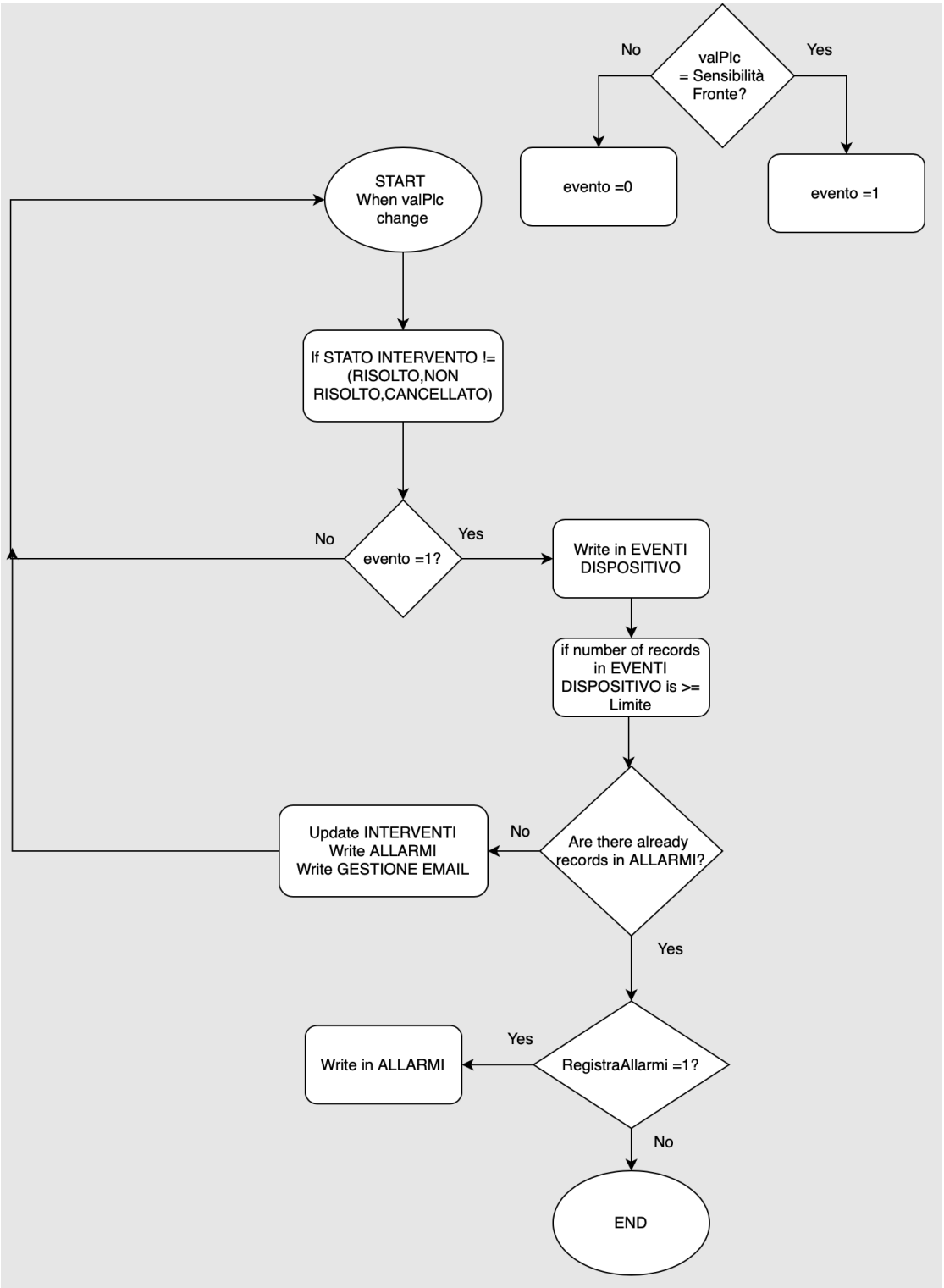

Figure A.4: Algorithm for the rule:"CONTROLLO BIT"

- CONTROLLO BIT A TEMPO

Parameters	**DataType**
SensFronte	Boolean
Limite	int4
LimiteTempo	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean
RegistraAllarmi	Boolean

"LimiteTempo": Indicates the time during which repeated switching of the bit can be critical. For example: A task can be created to check if a bit switches fifteen times in one day. If the fifteen switchings occur over a longer time, no trip will be scheduled. The algorithm is explained in the Figure A.5.

- CONTROLLO BIT RITARDO

Parameters	**DataType**
SensFronte	Boolean
LimiteTempo	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean
RegistraAllarmi	Boolean

This rule is used to check whether a bit remains in a certain state for a certain time. For example, to see if the connection bit goes to 0 and stays there for 10 minutes. The algorithm is explained in the Figure A.6.

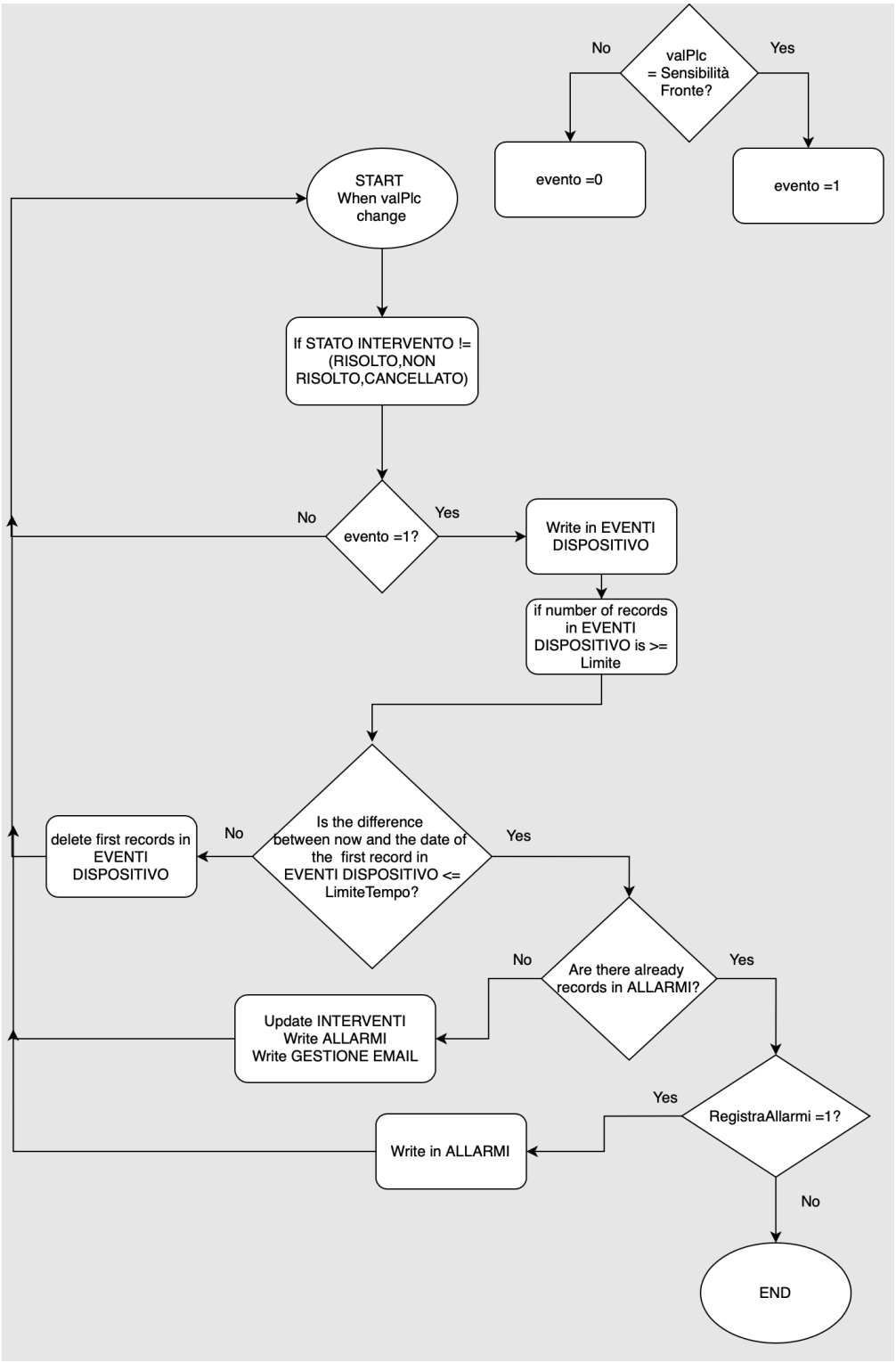

Figure A.5: Algorithm for the rule:"CONTROLLO BIT A TEMPO"

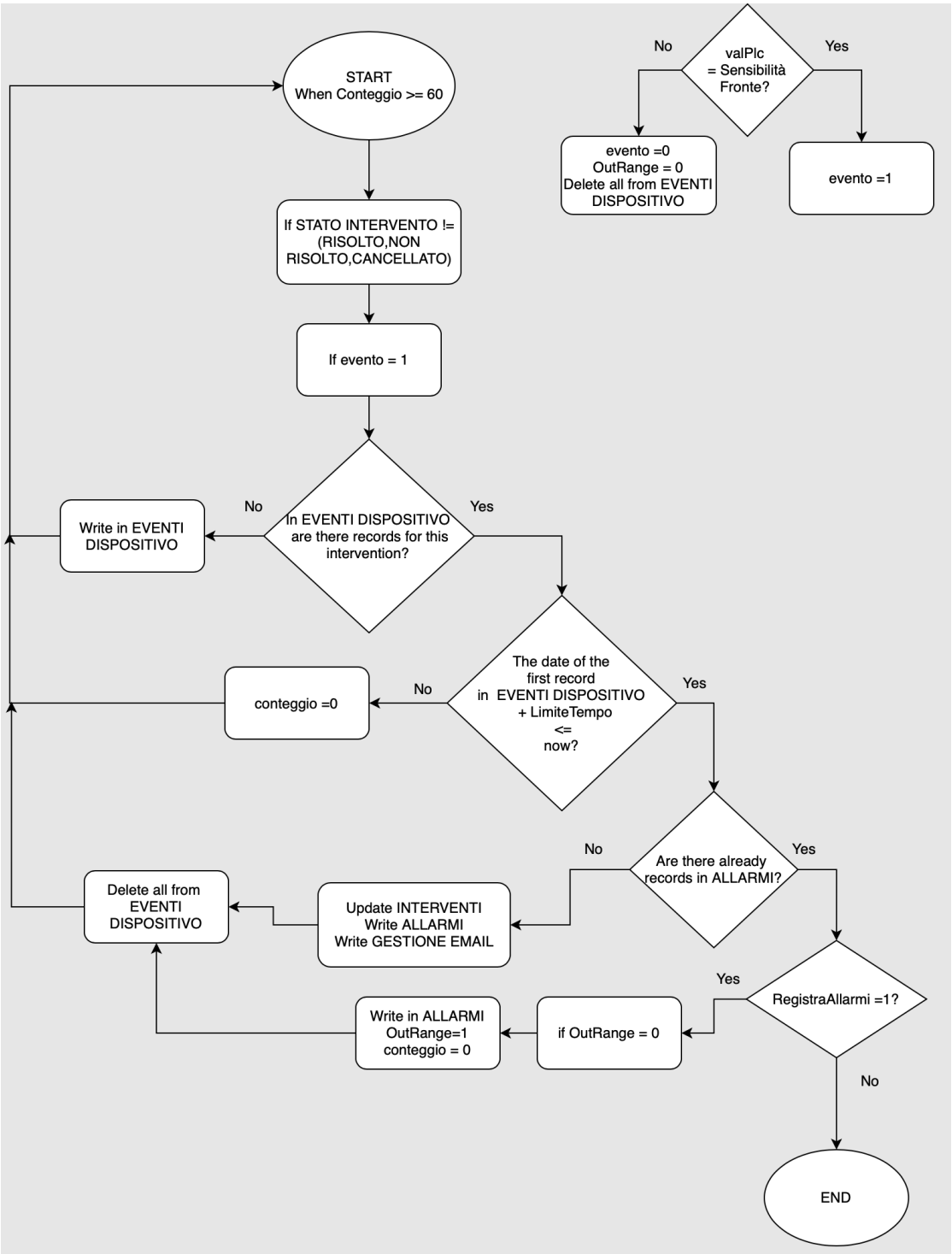

Figure A.6: Algorithm for the rule:"CONTROLLO BIT RITARDO"

ANALOGUE TAGS

- ANALISI ANALOGICO

Parameters	**DataType**
Campioni	int4
TempoCampionamento	int4
Errore	Float4
Limite	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean
RegistraAllarmi	Boolean

This rule is used to detect too abrupt changes in an analogue value which may therefore indicate a measurement anomaly. A certain number of samples ("Campioni") with a certain sampling time ("TempoCampionamento") are saved. Each saved value is compared with the average of the N previous samples; if the difference between the two values is greater than the parameter set in error ("Errore"), a counter is incremented. When the counter reaches the parameter "Limite", the intervention is scheduled. The algorithm is explained in the Figure A.7.

- ANALISI SOGLIE

Parameters	**DataType**
HIHI	float4
HI	float4
LOLO	float4
LO	float4
Range	String
Limite	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean
RegistraAllarmi	Boolean

First of all, the range to be controlled is chosen, and based on this choice, the threshold values are entered. LO and HI are the values of the upper and lower thresholds.

LOLO and HIHI are the threshold values plus a certain tolerance. The parameter calls "Limite" sets how many times the selected range (HI,LO,HIHI,LOLO,HIHI and LOLO, HI and LO) can be exceeded before the trip is scheduled. The algorithm is explained in the Figure A.8.

- ANALISI SOGLIE A TEMPO

Parameters	**DataType**
HIHI	float4
HI	float4
LOLO	float4
LO	float4
Range	String
Limite	int4
LimiteTempo	int4
Scadenza	int4
SottrazioneVita	int4
AutoRiprogrammazione	Boolean
OperatoreRiprogrammazione	Boolean
RegistraAllarmi	Boolean

The threshold values are set. LO and HI are the values of the upper and lower thresholds. LOLO and HIHI are the threshold values plus a certain tolerance. "Limite": This sets how many times the selected range (HI,LO,HIHI,LOLO,HIHI and LOLO, HI and LO) can be exceeded in a Time Interval before the trip is scheduled. "LimiteTempo": Indicates the time during which repeated range departure may be critical. The algorithm is explained in the Figure A.9.

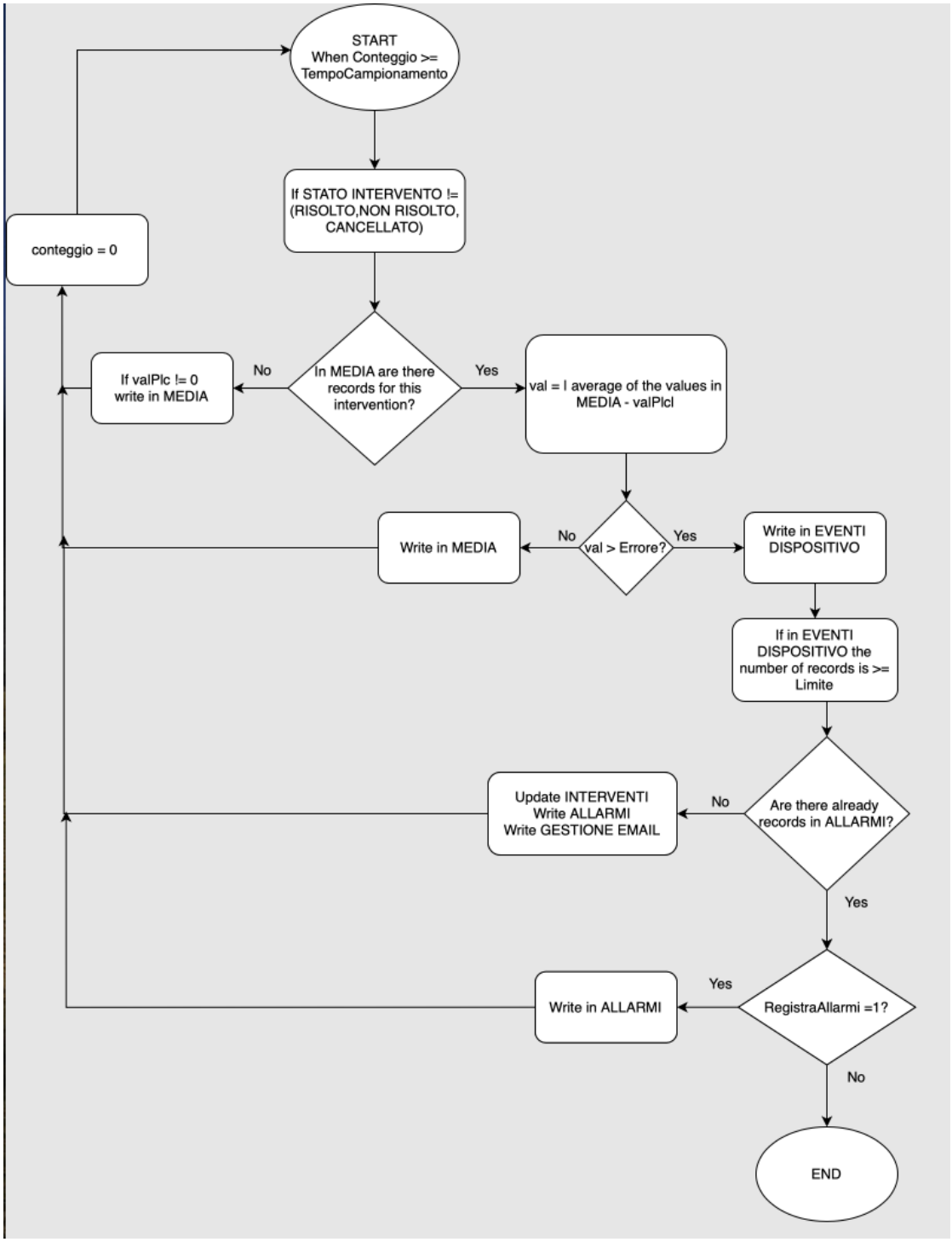

Figure A.7: Algorithm for the rule:"ANALISI ANALOGICO"

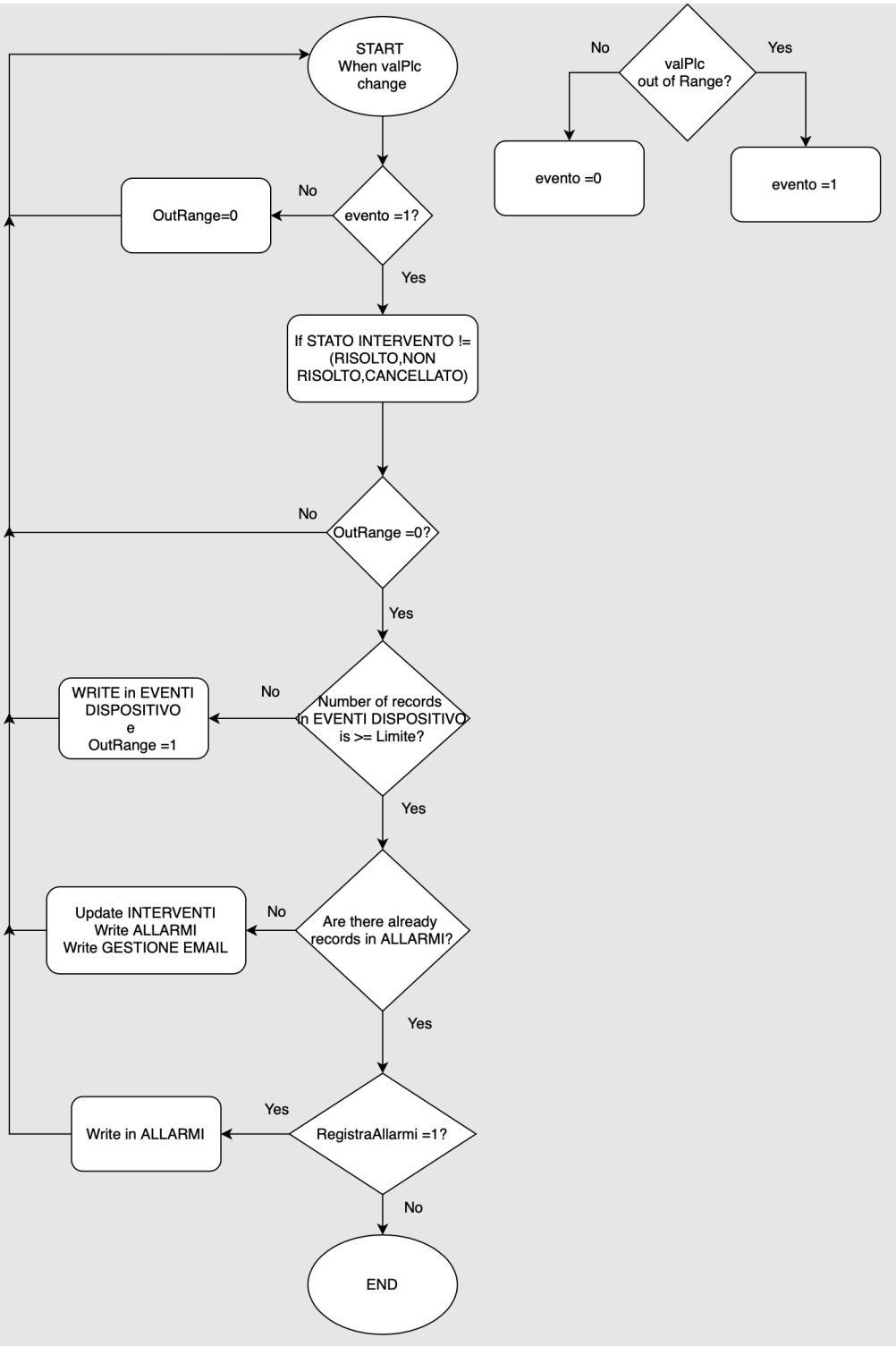

Figure A.8: Algorithm for the rule:"ANALISI SOGLIE"

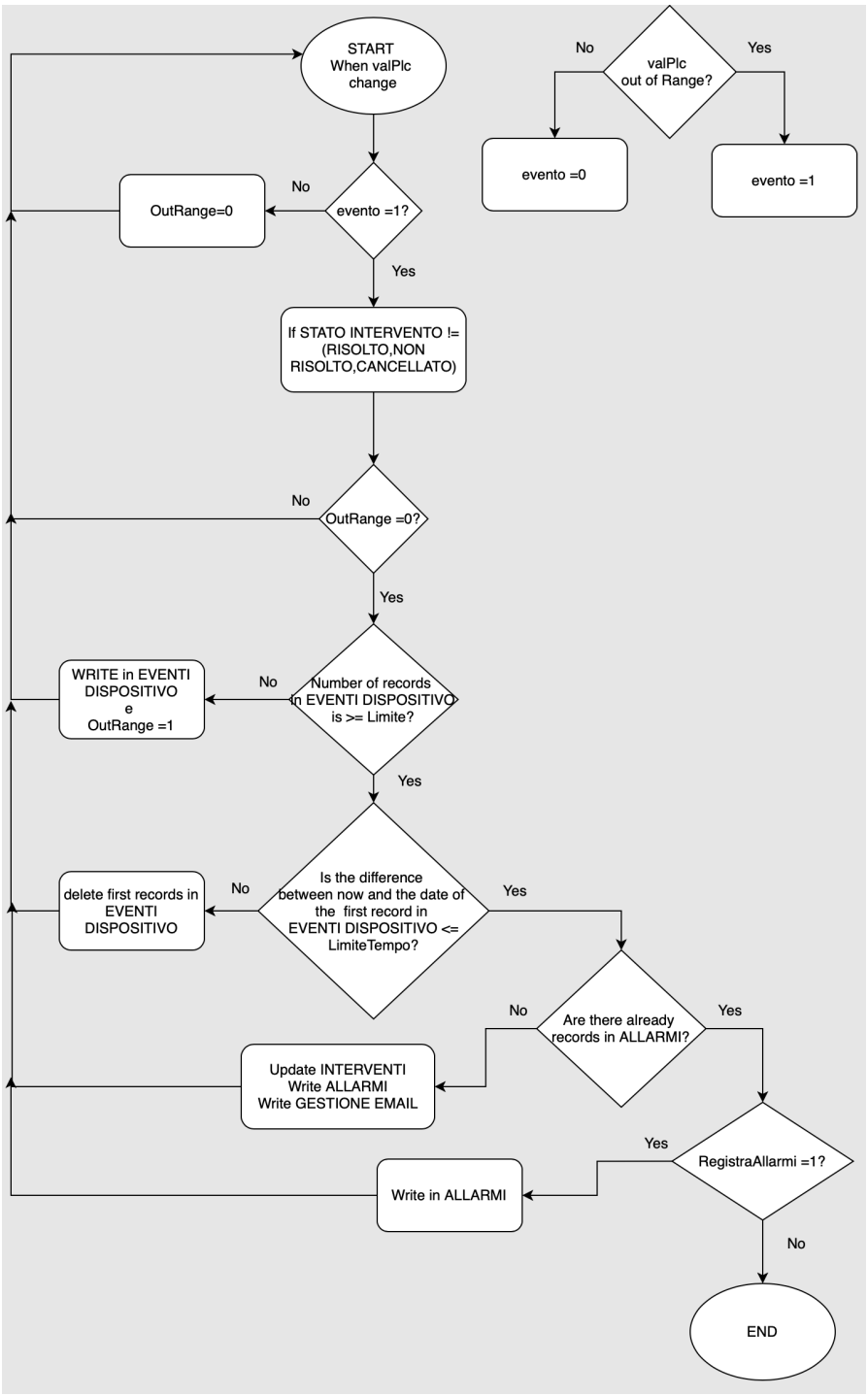

Figure A.9: Algorithm for the rule:"ANALISI SOGLIE A TEMPO"

Appendix B

Query SQL

All tables of the data display database in the ignition software are developed using more or less complex SQL queries. The "JOIN" command is mainly used to display data from different tables. The code of the scheduled operations table in the operator interface software is given as an example.

```
SELECT
    I.Id_Interv
   ,I.Nome_Interv
   ,I.Id_Tag
   ,CASE
       WHEN I.Id_Insieme = 1
       THEN
            (SELECT
                Nome_Area
            FROM AREA
            WHERE Id_Area=I.Id_InsiemeAssegnato)
       WHEN I.Id_Insieme = 2
       THEN
            (SELECT
                A.Nome_Area
            FROM AREA A
                INNER JOIN BLOCCO B ON A.Id_Area = B.Id_Area
                WHERE B.Id_Blocco = I.Id_InsiemeAssegnato)
       WHEN I.Id_Insieme = 3
       THEN
            (SELECT
                A.Nome_Area
            FROM AREA A
                INNER JOIN BLOCCO B ON A.Id_Area = B.Id_Area
                INNER JOIN DISPOSITIVO D ON D.Id_Blocco = B.Id_Blocco
                WHERE D.Id_Disp = I.Id_InsiemeAssegnato)
   END Nome_Area
```

```
28      ,CASE
29          WHEN I.Id_Insieme = 2
30          THEN
31              (SELECT
32                  B.Nome_Blocco
33              FROM BLOCCO B
34                  INNER JOIN AREA A ON A.Id_Area = B.Id_Area
35                  WHERE B.Id_Blocco = I.Id_InsiemeAssegnato)
36          WHEN I.Id_Insieme = 3
37          THEN
38              (SELECT
39                  B.Nome_Blocco
40              FROM BLOCCO B
41                  INNER JOIN AREA A ON A.Id_Area = B.Id_Area
42                  INNER JOIN DISPOSITIVO D ON B.Id_Blocco = D.Id_Blocco
43              WHERE D.Id_Disp = I.Id_InsiemeAssegnato)
44      END Nome_Blocco
45      ,CASE
46          WHEN I.Id_Insieme = 3
47          THEN
48              (SELECT
49                  Nome_Disp
50              FROM DISPOSITIVO
51              WHERE Id_Disp = I.Id_InsiemeAssegnato)
52      END Nome_Disp
53      ,CASE
54          WHEN I.Id_Insieme = 3
55              THEN
56                  (SELECT
57                      Id_Disp
58                  FROM DISPOSITIVO
59                  WHERE Id_Disp = I.Id_InsiemeAssegnato)
60      END Id_Disp
61      ,CASE
62          WHEN I.Id_Insieme = 3
63          THEN
64              (SELECT
65                  TOP 1 Nome_Categoria
66              FROM CATEGORIA  C
67                  INNER JOIN CATEGORIA_SOTTOCATEGORIA CSC ON
68                  C.Id_Categoria = CSC.Id_Categoria
69                  INNER JOIN SOTTOCATEGORIA SC ON
70                  SC.Id_Sottocat = CSC.Id_Sottocat
71                  INNER JOIN DISPOSITIVO D ON
72                  D.Id_Sottocat = SC.Id_Sottocat
73              WHERE D.Id_Disp = I.Id_InsiemeAssegnato)
74      END Nome_Categoria
75      ,CASE
76          WHEN I.Id_Insieme = 3
```

```
77                  THEN
78                       (SELECT
79                            Nome_Sottocat
80                       FROM SOTTOCATEGORIA SC
81                            INNER JOIN DISPOSITIVO D ON
82                            D.Id_Sottocat = SC.Id_Sottocat
83                       WHERE D.Id_Disp = I.Id_InsiemeAssegnato)
84       END Nome_Sottocat
85       ,ST.Nome_StInt
86       ,I.UDT_Path
87       ,I.Note
88       ,IG.Id_IntervGen
89       ,I.Id_StInt
90       ,I.Parametri
91  FROM INTERVENTI I
92       INNER JOIN INTERVENTI_GENERALI IG ON I.Id_IntervGen=IG.
         Id_IntervGen
93       INNER JOIN STATO_INTERVENTO ST ON I.Id_StInt = ST.Id_StInt
94       INNER JOIN INSIEME INS ON I.Id_Insieme = INS.Id_Insieme
95  WHERE
96       (ST.Nome_StInt = 'PIANIFICATO'
97       OR (ST.Nome_StInt = 'CHECK PARAMETRI' AND  IG.Visibile = 1))
98       AND IG.Id_TipInterv IN (
99       SELECT
100           Id_TipInterv
101       FROM TipINTERVENTO_TipOPERATORE
102       WHERE Id_TipOp IN
103            (SELECT
104                Id_TipOp
105            FROM TipOPERATORE_OPERATORE
106            WHERE
107                Id_Operatore =
108                     (SELECT
109                          Id_Operatore
110                     FROM OPERATORE
111                     WHERE
112                     Codice_Operatore = '{[System]Client/User/Username}'))
       )
113                     AND IG.Id_TipInterv IN
114                          (SELECT
115                               Id_TipInterv
116                          FROM TipINTERVENTO_TipOPERATORE
117                          WHERE Id_TipOp IN
118                                    (SELECT
119                                         Id_TipOp
120                                    FROM TipOPERATORE_OPERATORE
121                                    WHERE Id_Operatore =
122                                         (SELECT
123                                              Id_Operatore
```

```
124                                    FROM OPERATORE
125                                    WHERE Codice_Operatore =
126                                    '{[System]Client/User/Username}')))
127     AND {Root Container.Area Dropdown.Area}
128     AND {Root Container.Blocco Dropdown.Blocco}
129     AND {Root Container.Categoria Dropdown.Cat}
130     AND {Root Container.Dispositivo Dropdown.Disp}
131     AND {Root Container.Sottocategoria Dropdown.Sottocat}
132     AND {Root Container.Inserisci Id Interv.id_Interv}
133     AND {Root Container.IntPianificati.StatoInt.statoInt}
```

The output will look like the Figure B.1.

Id Intervento	Intervento	Area	Blocco	Dispositivo	Categoria	Sottocategoria	Stato Intervento	Note
20	Prova Connession...	RTU02a	QUADRO	Segnali PLC	PLC	Segnale Plc	PIANIFICATO	Prova Connessione
22	prova AST	RTU03	AMBIENTE	Sensore di livello	SENSORE	Sensore Radar	PIANIFICATO	
24	VitaC_Id_Tag_19	RTU13 Valle	PRT1	Motore Paratoia1	MOTORE	Motore paratoia	CHECK PARAME...	
25	VitaC_Id_Tag_22	RTU13 Valle	PRT2	Motore Paratoia2	MOTORE	Motore paratoia	CHECK PARAME...	
26	VitaC_Id_Tag_25	RTU13 Valle	PRT3	Motore Paratoia3	MOTORE	Motore paratoia	CHECK PARAME...	
27	VitaC_Id_Tag_28	RTU13 Valle	PRT4	Motore Paratoia4	MOTORE	Motore paratoia	CHECK PARAME...	
37	PRT1 - Apertura	RTU03	PRT1	Motore paratoia1	MOTORE	Motore paratoia	PIANIFICATO	

Figure B.1: Table with the output of the query SQL written above

Bibliography

[1] *Sito Web Italiano per SW di manutenzione*. URL: https://www.mainsim.com/academy/manutenzione-preventiva/. (accessed: 03.09.2021) (cit. on p. 6).

[2] Jakob Nielsen. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993 (cit. on pp. 6, 7).

[3] Ramez Elmasri; Shamkant B. Navathe. *Fundamentals of Database Systems – 7th Edition*. Pearson, 2016 (cit. on pp. 7, 8).

[4] Nelson Eng Adrienne Watt. *Database Design – 2nd Edition*. BCcampus, October 24, 2014 (cit. on pp. 7, 9, 10).

[5] Mark Whitehorn and Bill Marklyn. *Inside Relational Databases with Examples in Access*. Springer, 2007 (cit. on p. 7).

[6] Esakkirajan S. Sumathi S. *Fundamentals of Relational Database Management Systems*. Springer-Verlag Berlin Heidelberg, 2007 (cit. on p. 10).

[7] Karen Kent Keith Stouffer Joe Falco. «Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security». In: (2006) (cit. on pp. 13–15).

[8] Nimmo Hollifield Oliver. *The High Performance HMI Handbook*. PAS, 2008 (cit. on p. 14).

[9] Edwin Wright David Bailey. *Practical SCADA for Industry*. ELSEVIER, 2003 (cit. on p. 14).

[10] Muhammad Hamza El-Saba. «Supervisory Control And Data Acquisition (SCADA)». In: (2021) (cit. on p. 14).

[11] Stuart A. Boyer. *SCADA Supervisory Control And Data Acquisition 3rd Edition*. Iliad Engineering Inc, 2004 (cit. on p. 14).

[12] *Sito Web ufficiale di Ignition*. URL: https://inductiveautomation.com/scada-software/. (accessed: 16.09.2021) (cit. on p. 16).