IPS Instituto Politécnico de Setúbal

Escola Superior de Tecnologia de Setúbal

Rui Bernardo SoFIA

Shop Floor Intelligence Awareness

Trabalho de Projeto submetido como requisito parcial para obtenção do grau de **Mestre em Engenharia de Software**

Júri

Presidente (Doutora, Rossana Santos, ESTSetúbal-IPS)

Orientador (Doutor, Cláudio Miguel Sapateiro, ESTSetúbal-IPS)

Vogal (Doutor, Jacinto Estima, Universidade Europeia/ESTSetúbal-IPS)

Dezembro de 2019

Acknowledgments

I would like to dedicate this achievement in my academic life to the memory of my grandfather, my very first teacher, who taught me to read, write and perform my first math calculations, even before going to school. He was undoubtedly the person who most influenced my continuous chase for knowledge.

I would like to thank in the first place to my wife, for all the support, comprehension and motivation. Without her possibly I wouldn’t achieve this goal.

I would also like to express all my gratitude to the people that were more close to me in my life, that I believe, somehow, contributed to the person I am today, namely my parents, brothers, grandfathers, my daughter and closest friends.

I would like to thank my advisor, Professor Cláudio Sapateiro, for believing that I would be up to the challenge he presented me, and by all the support and guidance given in the execution of this project.

Finally, I would also like to thank to Crown, in the person of Engineer Fernando Lima, for all the availability, enthusiasm and all the conditions they promptly made available, in order to let nothing miss or jeopardize the success of this project.

Resumo

Nos dias que correm, os dados são um dos ativos mais importantes de uma empresa. O conceito de Business Intelligence surge precisamente como um conjunto de técnicas que permite extrair valor desses dados, transformando-os em informação.

É neste contexto que se insere a execução deste projeto. Fazendo uso de um conjunto de dados, que já eram recolhidos pela empresa, e trabalhando-os, de forma a gerar informação que permita gerar impacto positivo quer nos operadores da fábrica (motivando uma competição saudável entre os mesmos, para que tenham o mínimo de paragens possíveis nas linhas de produção pelas quais são responsáveis), quer na equipa de gestão e coordenação (permitindo aferir informação disponibilizada através de indicadores de desempenho).

A informação extraída a partir do processamento destes dados é disponibilizada através de dashboards para várias áreas da empresa, refletindo ocorrências verificadas nas linhas de produção, que de acordo com um conjunto de regras definidas pela equipa de gestão, devem ser evidenciadas.

Palavras-chave: Business Intelligence, Linhas de produção, Gestão de alertas de incidentes, Paragens de linha, Dashboard

Abstract

Nowadays, data is one of the most important assets of a company. The Business Intelligence concept emerges precisely as a set of techniques that aim to extract value from this data, transforming it in information.

It is in this context that this project execution belongs to. By making use of a set of data, that is already gathered by the company, and working it, in a way to create information that allows to generate a positive impact either on factory operators (by promoting a healthy competition among them, in order to prevent break downs as much as possible, in production lines they are responsible for) and management or coordination teams (by providing information available through performance indicators).

The information extracted from this data processing is made available through dashboards, to several company areas, reflecting incidents verified in production lines, that according to a set of rules, created by the management team, should be highlighted.

Keywords: Business Intelligence, Factory production lines, Incident warnings management, Line break downs, Dashboard.

Contents

Resumo ..... **iv**
Abstract ..... **v**
Contents ..... **vi**
List of Figures ..... **viii**
List of Tables ..... **ix**
List of Acronyms ..... **x**
Chapter 1 ..... **1**
Introduction **1**
1.1. Motivation and Goals **1**
1.1.1. Context *1*
1.1.2. Operational Motivation *1*
1.1.3. Requirements *2*
1.2. Methodology **3**
Chapter 2 ..... **5**
Project Scope **5**
2.1. Business Intelligence Project Framework **5**
2.2. Technological Options **6**
Chapter 3 ..... **8**
Conception **8**
3.1. Components Architecture **8**
3.2. Logical Architecture **9**
3.3. Sitemap **10**
3.4. Modeling and Processing Data **11**
3.4.1. Data Model *11*
3.4.2. Data Processing (ETL) *13*
Chapter 4 ..... **15**
Developed System **15**
4.1. Home Page **15**
4.2. Authentication Page **15**
4.3. Full Screen Dashboard **16**
4.4. Dashboard History **17**
4.5. Lines **19**
4.6. Warning Sets **21**
4.7. Edit Plant **23**

4.8. Stats **25**
Chapter 5 ..... **27**
Conclusions and Limitations Analysis **27**
Bibliography **29**

List of Figures

Fig. 1 – Agile Methodology 3
Fig. 2 - Kimball's BI Framework 6
Fig. 3 - Components Arquitecture 8
Fig. 4 - Logical architecture 9
Fig. 5 – Sitemap 10
Fig. 6 - Entity Relationship Diagram 12
Fig. 7 - ETL Workflow 13
Fig. 8 - Home Page 15
Fig. 9 - Sign in Page 16
Fig. 10 - Full Screen Page 16
Fig. 11 - Dashboard Page 17
Fig. 12 - Dashboard Warning 18
Fig. 13 - Dashboard Warning Expanded 18
Fig. 14 - Dashboard Warning Disabled 19
Fig. 15 - Lines Page 19
Fig. 16 - Line Edit 20
Fig. 17 - Line View 21
Fig. 18 - Warning Sets List Page 22
Fig. 19 - Warning Set Edit 22
Fig. 20 - Edit Plant Page 24
Fig. 21 - Edit Plant - Line and Blocks 24
Fig. 22 - Stats Page 25
Fig. 23 - Project in production environment 27

List of Tables

Table 1 - MuSCoW Requirements Table 3

List of Acronyms

BI	*Business Intelligence*
CRUD	*Create, Retrieve, Update and Delete*
D3	*Data-Driven Documents*
EF	*Entity Framework*
LINQ	*Language-Integrated Query*
KPI	*Key Performance Indicator*
ETL	*Extract Transform and Load*

Chapter 1

Introduction

In this first chapter it will be presented the motivation that led to the project, as well as the generic approach for the application purpose.

It is also detailed the requirements gathered and the methodology followed in this project that led to a fully implemented pilot that was conceived with a strict communication with the company, where it is now being used.

Part of the concepts used in the developed solution derive from the work "**Bringing Human Factor to Business Intelligence**" published in co-authorship with professor Cláudio Sapateiro in the "**10th Innovation, Entrepreneurship and Knowledge Academy Conference**", in June of 2019, in Verona, Italy. Subsequentially, this presentation resulted in an invitation to submit an extended version to be published in the "**International Journal of Enterprise Information System".**

1.1. Motivation and Goals

1.1.1. Context

With the purpose of having a dashboard to monitor break downs in a continuous production lines plant, Crown Portugal has contacted Escola Superior de Tecnologia de Setúbal, in order to promote a real working life project development partnering with academy.

Regarding this, Prof. Cláudio Sapateiro challenged me to embrace this as my final project in the Master Degree in Software Engineering.

Assuming its strong Business Intelligence and Data Analysis components, I understood it as a project that would consolidate my knowledge in these areas and accepted the challenge.

1.1.2. Operational Motivation

Crown primary motivation consisted in reducing production line break downs, to

promote line efficiency related KPIs. Crown understands that by developing this project they will be promoting an effective mechanism of warning operators to incidents, better managed since they are additionally furnished with gravity levels through a system of rules underlying a trigger mechanism for notifications. Additionally, the public (in house) display of such warnings, besides, bringing awareness for opportunities for timed and efficient interventions, will also promote an healthy (peer pressure) competition among operators responsible for bringing production lines back to live.

Nowadays, incidents are reported manually by line operators, in paper sheets designed for that purpose. These sheets are then taken to the office, to someone who is responsible for standardizing the reported information, and transferring this data into a digital format, through excel files, and after this step, also report this data in their own central system. This process is somehow outdated and very prone to errors, so this project could also be faced as the start of a digital transformation regarding the data gathering in what concerns line production incidents.

1.1.3. Requirements

In order to better understand and manage business priorities, we followed the prioritization technique known as MuSCoW. This technique enables us to identify the priority of each requirement, setting each one as "**Mu**st Have", "**S**hould Have", "**Co**uld Have" or "**W**ould Have" (which in this last case may not be implemented, but it is already identified).

The following table reflects the requirements and its priority that were gathered in this process.

Code	Requirement	Must Have	Should Have	Could Have	Would Have
R01	Incidents data imported from excel files	x			
R02	Warnings are based on rules applied to incidents	x			
R03	Rules will be defined by incident code, scope time, minimum incident duration time, and minimum number of incidents	x			
R04	Warnings should be visible in a full screen dashboard	x			
R05	Rules have to be manageable	x			
R06	Production Lines should be manageable		x		
R07	Plant should be editable			x	
R08	Warnings should have 2 levels of gravity (Red and Orange)			x	

R09	Warnings should present details regarding incidents that generated them			x	
R10	Incidents data processed in real time				x
R11	3 access profiles: public, supervisor, manager			x	
R12	Stats chart regarding total warnings per month		x		
R13	Stats chart regarding blocks with more warnings grouped by month		x		
R14	Stats chart regarding incidents per units produced, grouped by month		x		
R15	Production Lines should have line blocks integrated		x		
R16	Internal network access only		x		
R17	Be able to manually deactivate a warning			x	
R18	When rules are processed, ignore days without production (regarding scope time)			x	
R19	Warnings history should be available			x	
R20	Incidents should be reported digitally, instead of manually				x
R21	Dashboard should auto refresh			x	

Table 1 – MuSCoW Requirements Table

All requirements were successfully accomplished except the “Would Have”, that were relegated to posterior phases.

1.2. Methodology

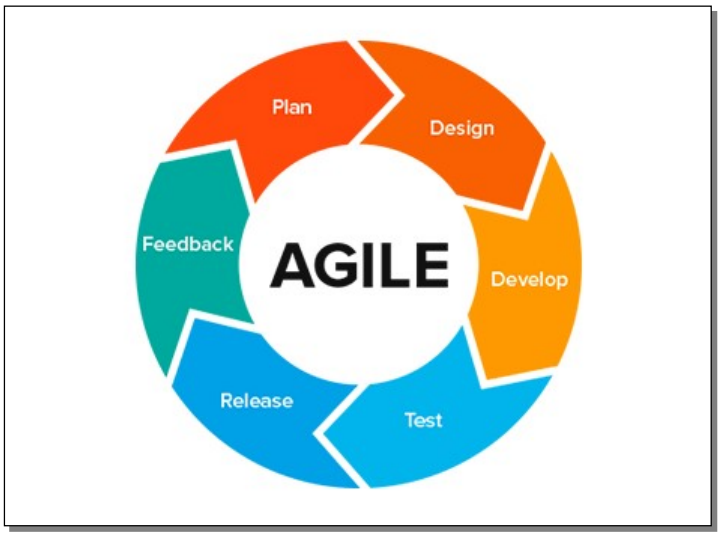

Fig. 1 – Agile Methodology

This project was developed using Agile methodology, since it was supported on iterative incremental cycles with each client iteration including:

- Evaluate new developed features (except on first iteration)
- Gather feedback and eventual requirement changes and additions

After each client iteration a new cycle was followed that included:

- Analysis and (re) design (for accommodating requirements)
- Development
- Tests
- Deploy

Chapter 2
Project Scope

This chapter presents general considerations regarding the project scope and scientific areas and technologies that this project was based on.

2.1. Business Intelligence Project Framework

Business Intelligence (BI) has evolved from the earlier conception of decision support systems and executive information systems on the 80s and 90s to the early days of the current agile, decentralized and data analytics driven orientation. Technological evolution and (consequent) data collection and processing capabilities nowadays have allowed to further sustain traditional BI goals as well as move toward the exploitation of new ones. In fact, besides the typical focus on historical data reconnaissance and future forecasting, we currently assist to emphasis given to inform present action, based on immediate analysis of high pace generated information.

A BI project encompasses dedicated organizational and technological processes and resources, with associated skills and tools to derive pertinent and assertive information to support decision and action. Literature reports the historical need for BI projects' structuring to promote success by complying with Critical Success Factors (CSF) and further engage into agile development (Larson & Chang, 2016).

Traditionally, two major frameworks had established the reference toward guidance on BI projects. The Corporate Information Factory (CIF) by Bill Inmon (Inmon, Imhoff, & Sousa, 2001) and the Dimensional Modelling (DM) approach by Richard Kimball (Kimball & Ross, 2002). Although originally strongly coupled with data warehouse design, both still cover a full range of concerns on BI projects enactment on a domain independent manner. That merit and, therefore, their pertinence remains even when data warehouse or current alternatives for (big) data storage and analytics are accomplished through state-of-the-art technologies. As a natively bottom-up, incremental, and processes-oriented approach, Kimball's framework more readily offers possibilities to align with the current tenets of agile project management methodologies for BI projects enactment that pursue expeditious deliveries. Therefore this work builds from Kimball's framework presented in figure 2.

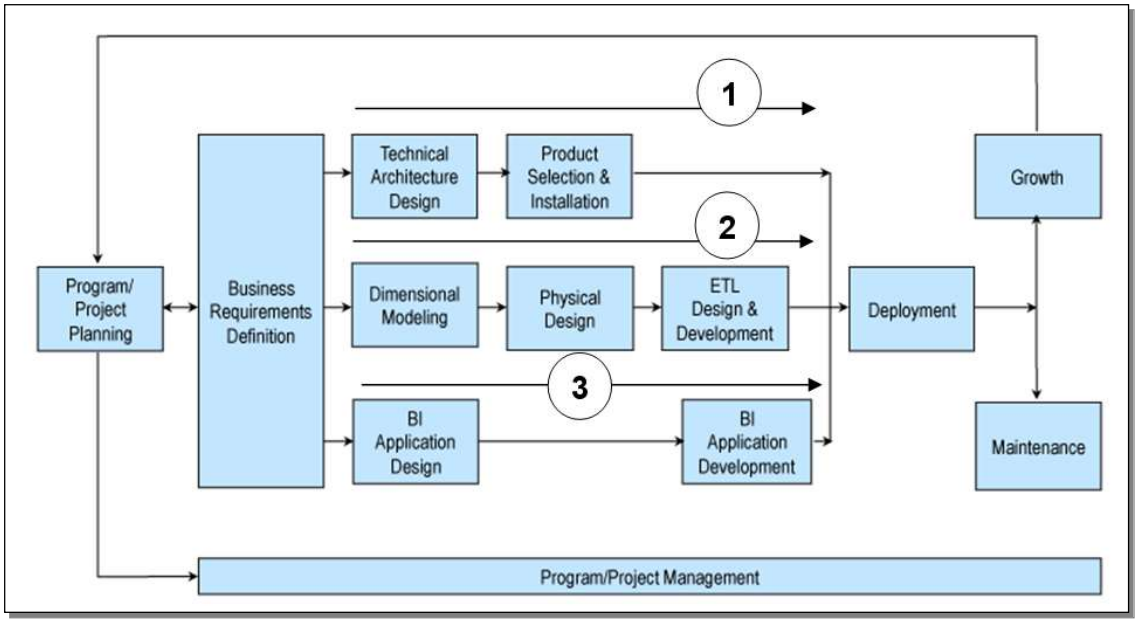

Fig. 2 - Kimball's BI Framework

Besides the typical project management functions (e.g. tasks division and structuring, roles definition and assignments, and scheduling) the framework envisions 3 main operative workflows developed over elicited project's business requirements:

1. Addresses issues related with technological infrastructure as network, licensing, products configuration, among others;
2. Refers to the core concerns on data modelling, storage design and associated processes on E-Extraction, T-Transformation and L-Loading;
3. Deals with the more visible side (related to end user) of BI applications, namely dashboard, reporting and analytic abilities design and development.

This work, to some extent given its scale, roots on Kimball's framework in its conceptual organization, to provide guidance on the interactions with the client and structure the project development.

2.2. Technological Options

Regarding technological options, a few points were taken into consideration to better support the decisions made. It was important to determine what set would better fit both the existing infra-structure and the development skills.

Having that in mind, and considering:

1. Crown Portugal already has SQL Server and Windows Servers virtual machines in their technological infra structure;
2. Easy and desirable integration between Microsoft development products and SQL Server;
3. Among Microsoft development technologies, MVC.net (Microsoft, 2018) has great development accelerators trough components like Entity Framework (Microsoft, 2016) (quick mapping of database objects), LINQ (Microsoft, 2017) (allows the usage of queries directly in database mapped objects) and scaffolding (technology that automatically generates CRUD operations and views of a given database table);
4. My own knowledge in MVC.net, and there by, representing a smaller or null learning curve regarding this technology;

The natural options ended up being the following technologies selected in order to accomplish this project development:

- **SQL Server** database;
- **MVC.net** development framework;
- **Bootstrap** (Bootstrap, 2013) framework as a basis for responsive pages;
- **D3.js** (Bostock, 2016) framework for manipulating graphical vectors, that allow charts rendering.

Chapter 3
Conception

This chapter presents the major design and analytical decisions that were taken in order to achieve the implementation of requirements gathered among the client.

Architecture and data model will be detailed ahead.

3.1. Components Architecture

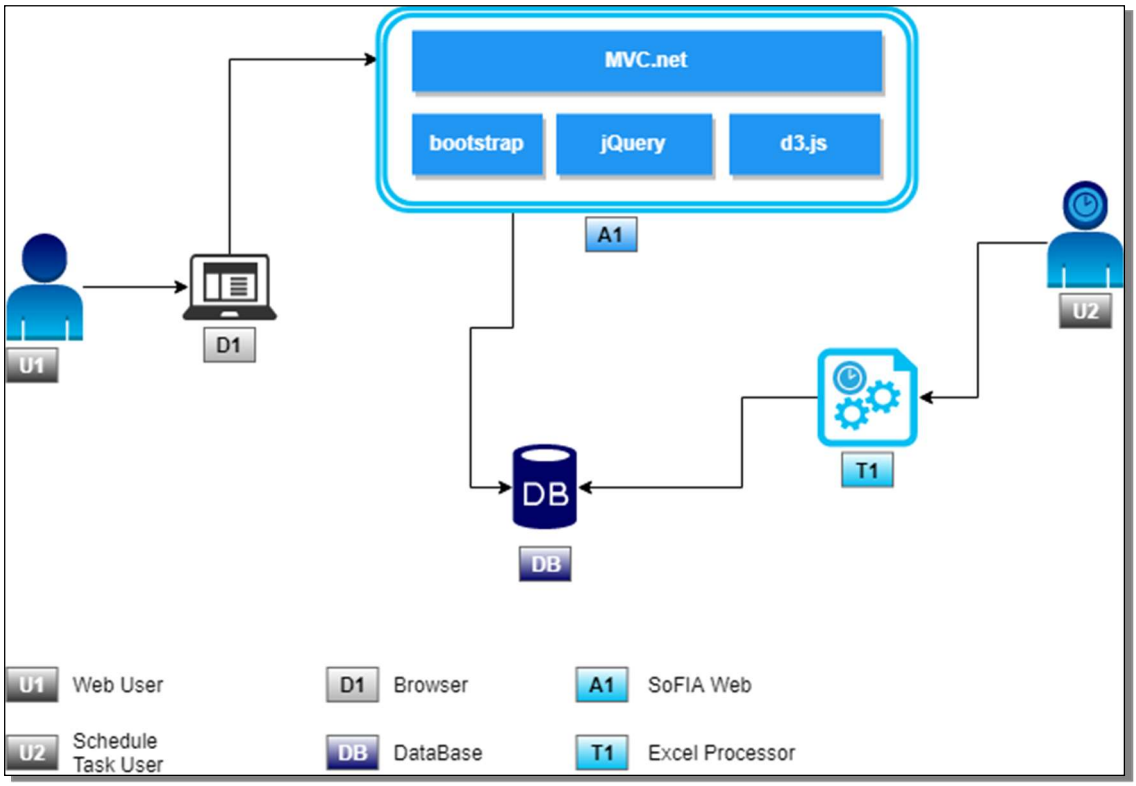

Fig. 3 - Components Arquitecture

A1 – SoFIA Web

Web application, liable of interaction with users, in order to analyze all information generated by the project.

DB – DataBase

Database where all data are kept and processed, both data generated by web application and data imported from excel files into the system. Stablishes the bridge between SoFIA Web application and Excel Processor independent automatic task.

T1 – Excel Processor

Scheduled Task, that has the main purpose of importing data from excel files, save them in the database and process them (ETL Process), generating information that will be consumed by SoFIA Web Application. Being an automatic task, it does not have any interaction with users.

3.2. Logical Architecture

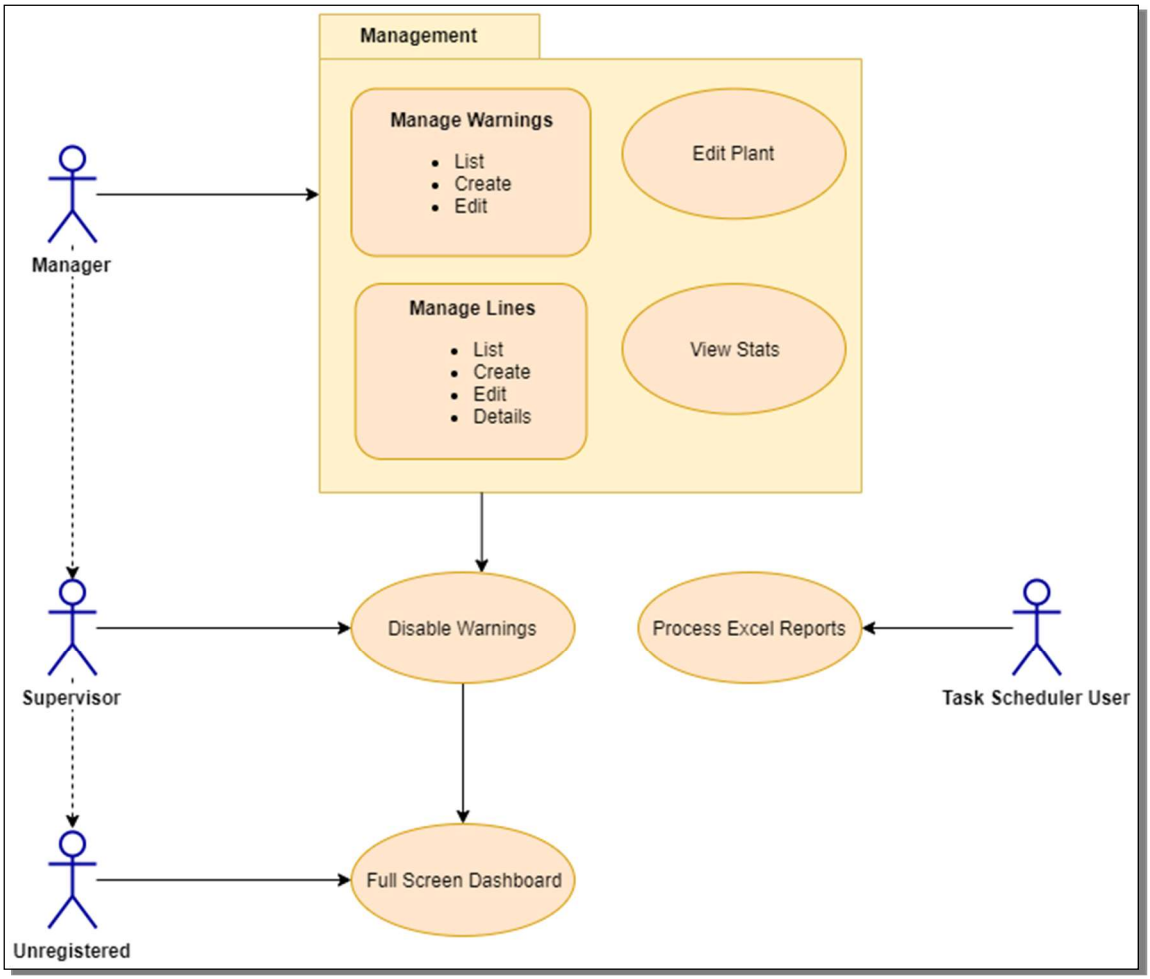

Fig. 4 - Logical architecture

Unregistered

Public access, no authentication needed.

Supervisor

User profile that requires authentication. Allows access to features available for Unregistered profile and also the ability to disable warnings.

Manager

User profile that requires authentication. Allows access to features available for Supervisor profile and also all management modules, plant edit (geographic warning disposal) and business intelligence area, where all the statistical charts regarding warnings and production lines information are available.

Task Scheduler User

System user, used to invoque the execution of the data processing automatic scheduled task (ETL Process).

3.3. Sitemap

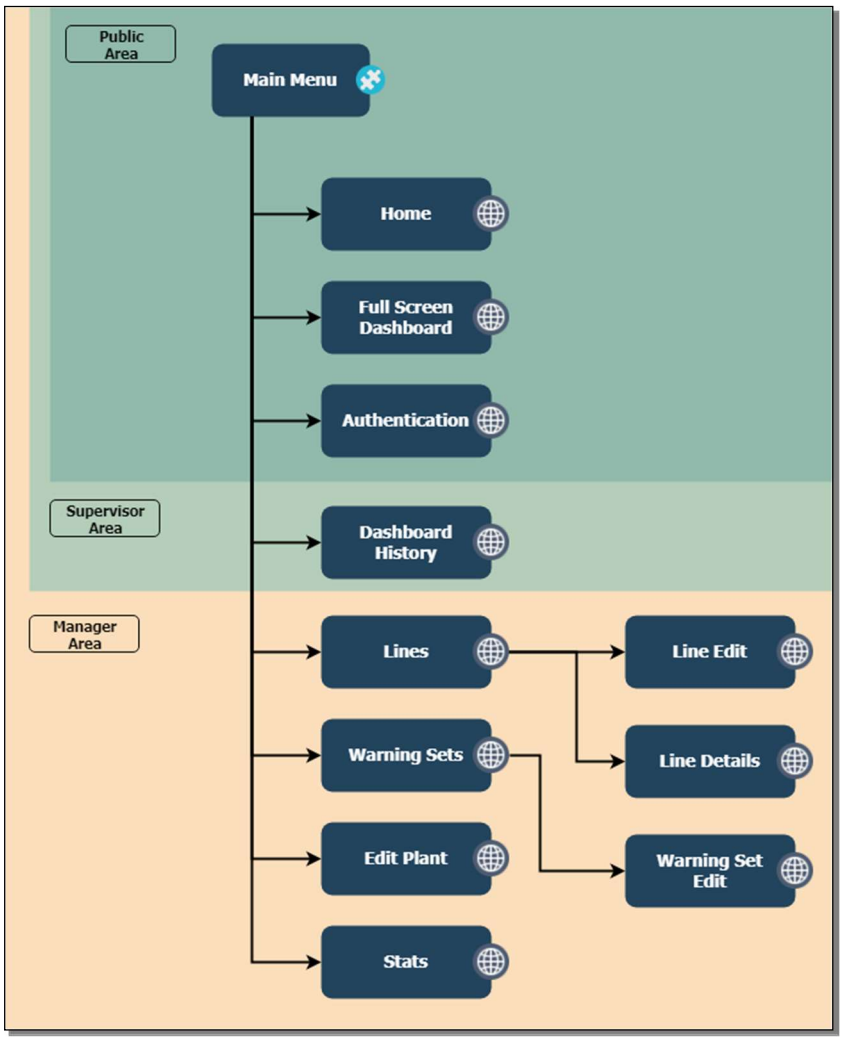

Fig. 5 – Sitemap

In Fig. 5 we can find a picture that describes the pages available in the backoffice

created in the context of this project. All their features and specifications will be detailed in the chapter “Developed System”.

3.4. Modeling and Processing Data

3.4.1. Data Model

In Fig. 6 we can find the Entity Relationship Diagram that supports this project. We can see two distinct groups of tables, “Bulk Import Group“ that aims to support all the data upload management, prior to its processing, and a second group “Warning System” that supports all the system either with processed data, or data created by users in Back Office.

In this model we can see, among others, the following main tables:

Incidents – all incidents occurred in lines, reported by line operators.

Codes – List of possible codes, previously defined, that represents one type of incident.

Lines – represents every production line in factory.

Blocks – each line is composed by a set of blocks, Incidents may occur in any of these blocks.

LineBlock – relates a Line with its blocks.

CodeBlock – An incident does not have the information of the block where it occurred, so this table is defined previously in order to relate a code with a line block. This way we know which was the block generating an incident from a production line.

WarningSet – set of rules that defines when a group of incidents should generate a warning.

Warnings – When a group of incidents match the criteria defined in WarningSet, a warning is generated, that will be displayed in dashboards.

WarningIncidents – relates a warning with the group of incidents that generated it.

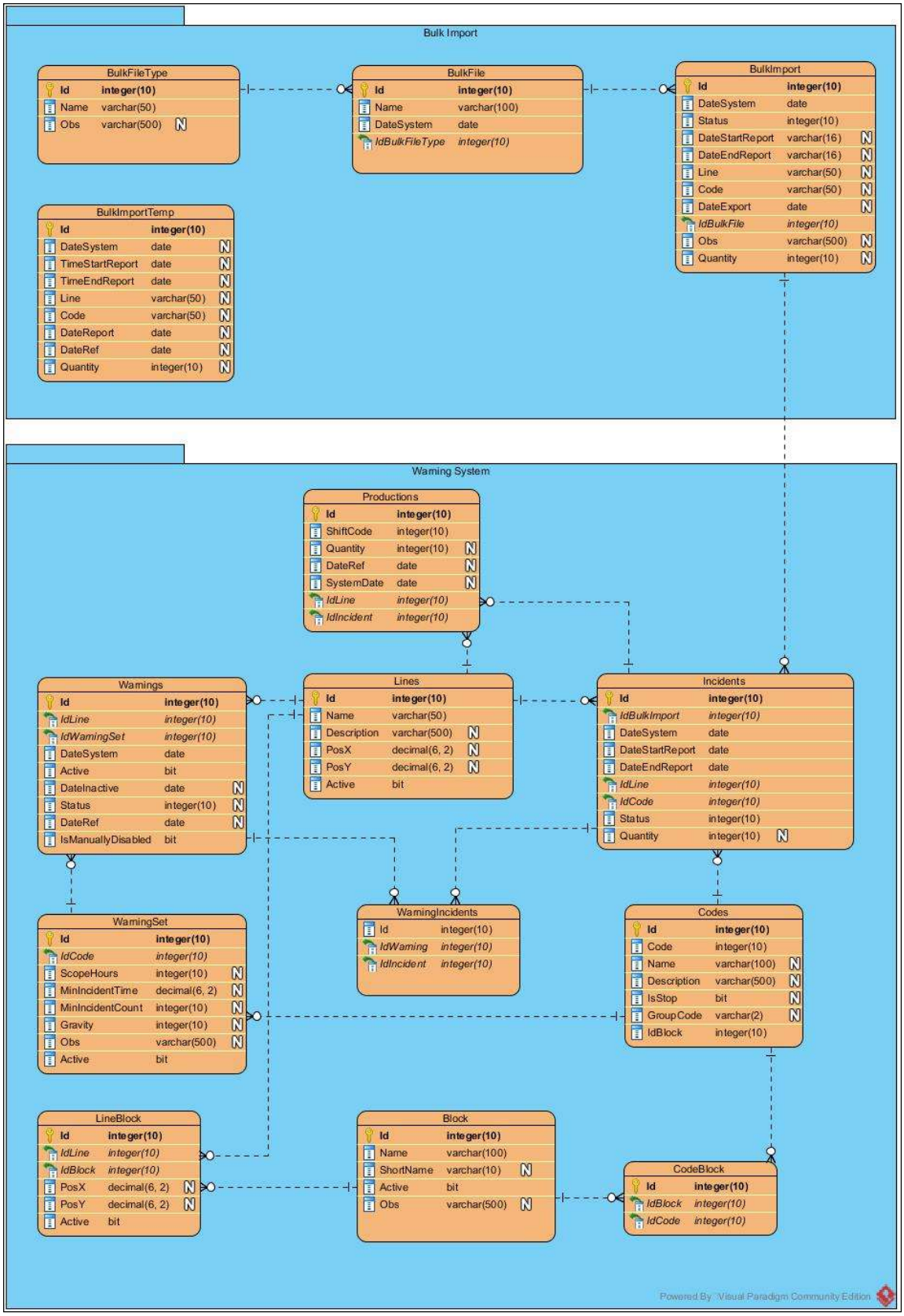

Fig. 6 - Entity Relationship Diagram

3.4.2. Data Processing (ETL)

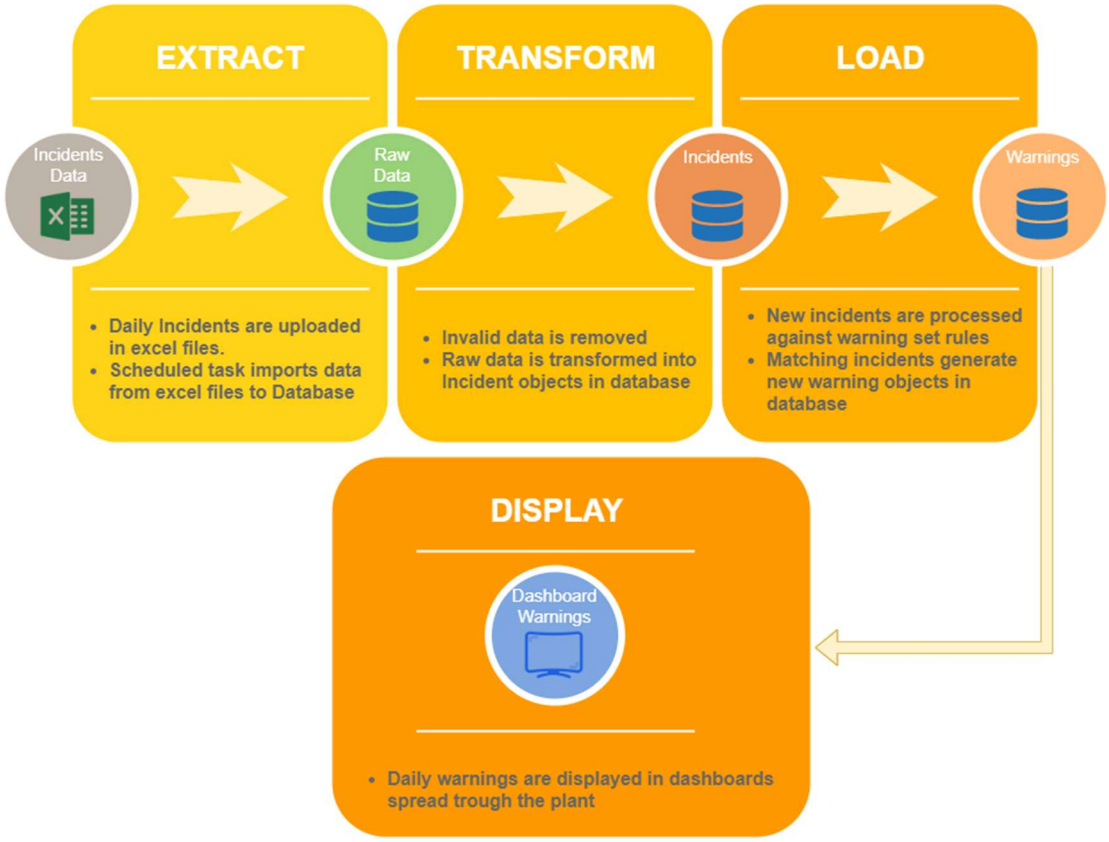

Fig. 7 - ETL Workflow

Below are the described steps that make part of the ETL Process, which run every time the task Excel Processor is triggered.

1. **Extract**

1.1 If there is any excel file to be processed, the file is read, according to the predefined excel structure, and all data is saved in database table [**BulkImportTemp**].

1.2 After all data is saved, it is invoked the stored procedure [**ProcessTempIncidents**] that will **delete** all invalid or duplicate data, **copy** all remaining data to table [**BulkImport**] and finally **delete** all data from [**BulkImportTemp**].

2. **Transform**

2.1 Store Procedure [**ProcessTempIncidentsFinish**] is invoked, in order to start

the transforming data, that was not yet transformed, from [**BulkImport**] into [**Incidents**].

3. Load

3.1 Stored Procedure [**WarningsProcessByDate**] is executed, which will create a cursor, iterating for every active Warning Set, and invoking, in each iteration, Stored Procedure [**WarningsProcessByWarningSet**].

3.2 The final step of this process is executed in the Stored Procedure [**WarningsProcessByWarningSet**], where for each active warning set, rules are matched against the data inserted in table [**Incidents**]. In case of positive matching criteria, are then created new data tuples in table [**Warnings**].

Chapter 4
Developed System

In this chapter it will be presented the implemented solution, showing all relevant screens and its features.

4.1. Home Page

Start page, available for anyone inside the network.

Fig. 8 - Home Page

Any user with access to the network is able to access Home Page, Authentication Page and Full Screen Page.

4.2. Authentication Page

In this page it is possible to sign in, and therefore, be recognized as a profile that

has access to features in the application, that are not allowed to public access.

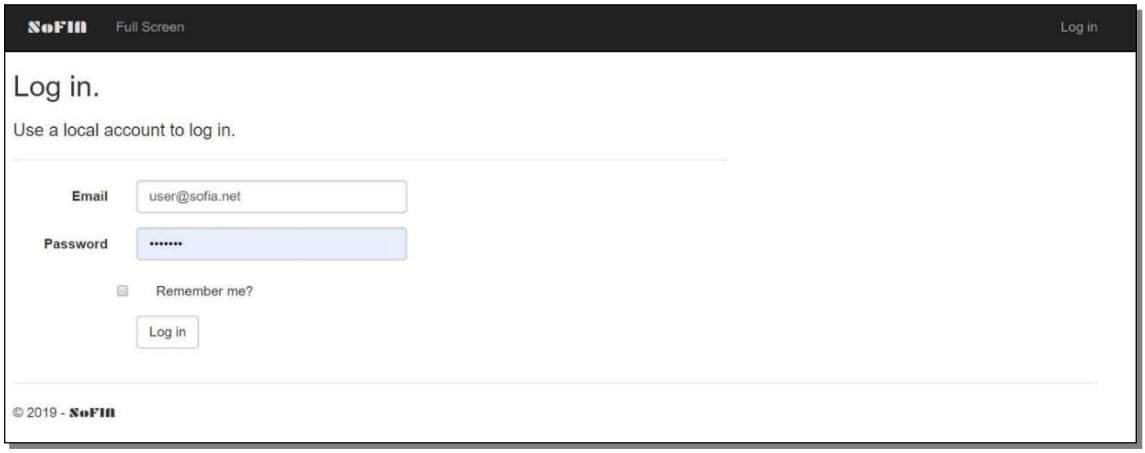

Fig. 9 - Sign in Page

4.3. Full Screen Dashboard

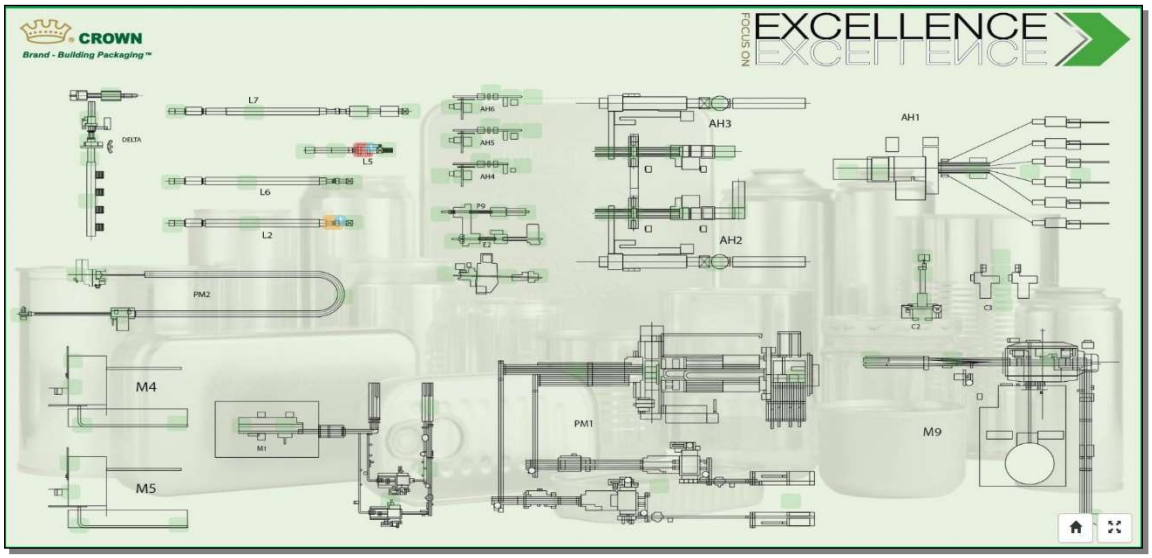

Fig. 10 - Full Screen Page

Any network user will be able to access this page. It will be generically used in big screens spread among the plant, where it is possible to see the warnings generated by the last data processing. This page automatically refreshes each 2 minutes, in order to reflect eventual changes originated by a data processing occurred at any time

In the right low corner there are available options to enter full screen (ou leave, in case it is already in full screen mode) and navigate to Home Page (as menu is not available in this page in order to optimize space).

4.4. Dashboard History

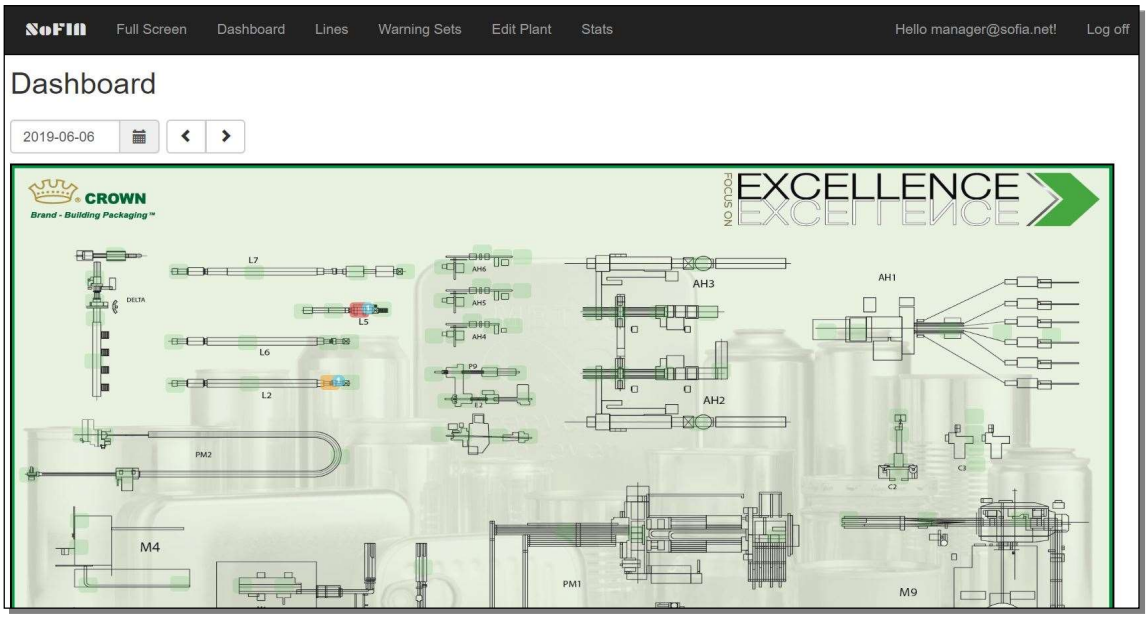

Fig. 11 - Dashboard Page

Page available only for **Supervisor** and **Manager** profiles. Allows access to warnings history, by selecting a date in the upper left corner. Warnings are presented in red or orange (depending on gravity defined for that kind of warning) on top of line blocks where they occurred. It is also shown how many warnings were generated for that line block. By selecting a block with active warnings it is also possible to see its details (list and type of warnings and incidents that generated them).

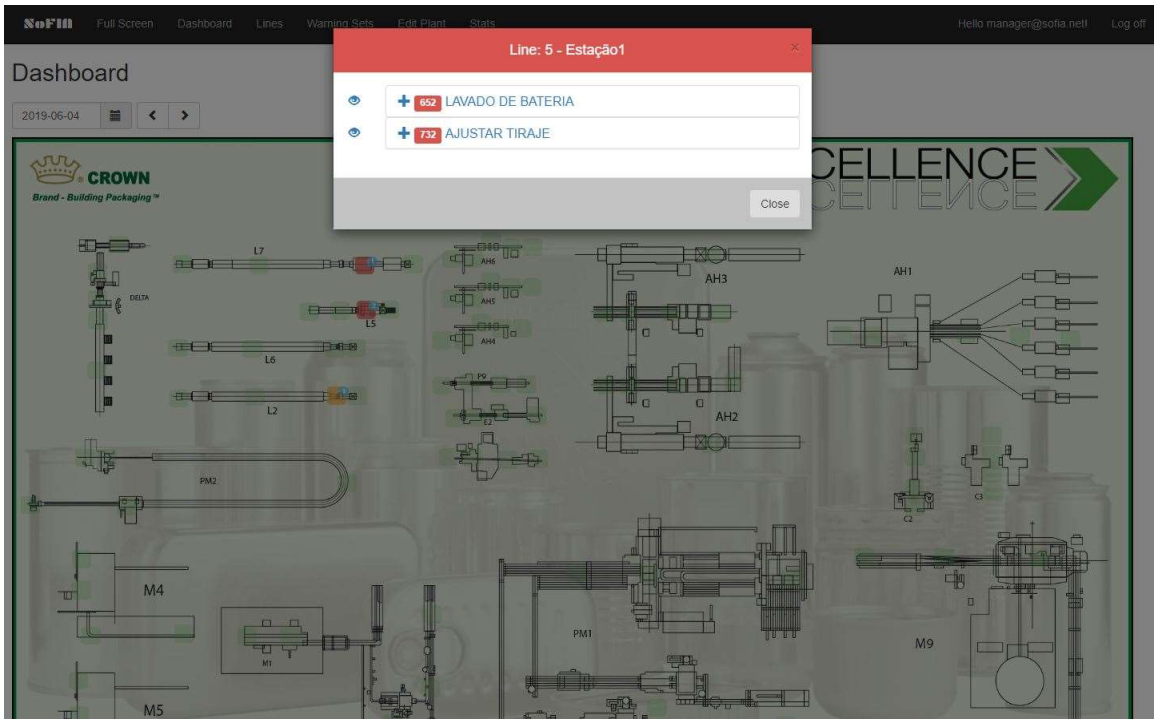

Fig. 12 - Dashboard Warning

After selecting a block with active warnings, a pop up is shown, having the list of all warnings that occurred in that block. It is possible to see warning details by pressing icon "plus" ✚.

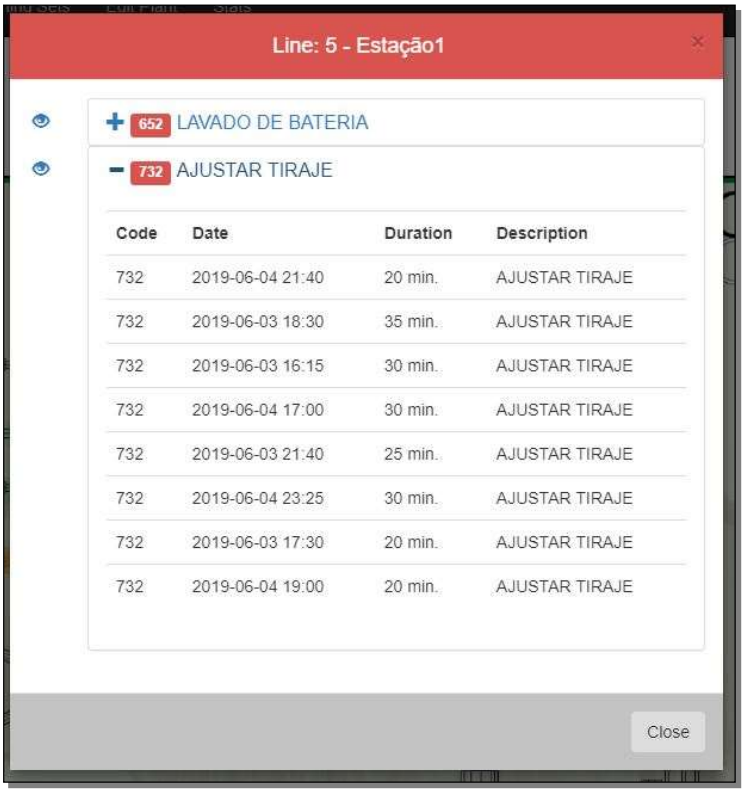

Fig. 13 - Dashboard Warning Expanded

It is also possible to manually disable a warning, by pressing the icon “eye” immediately at the left of the warning that we wish to disable.

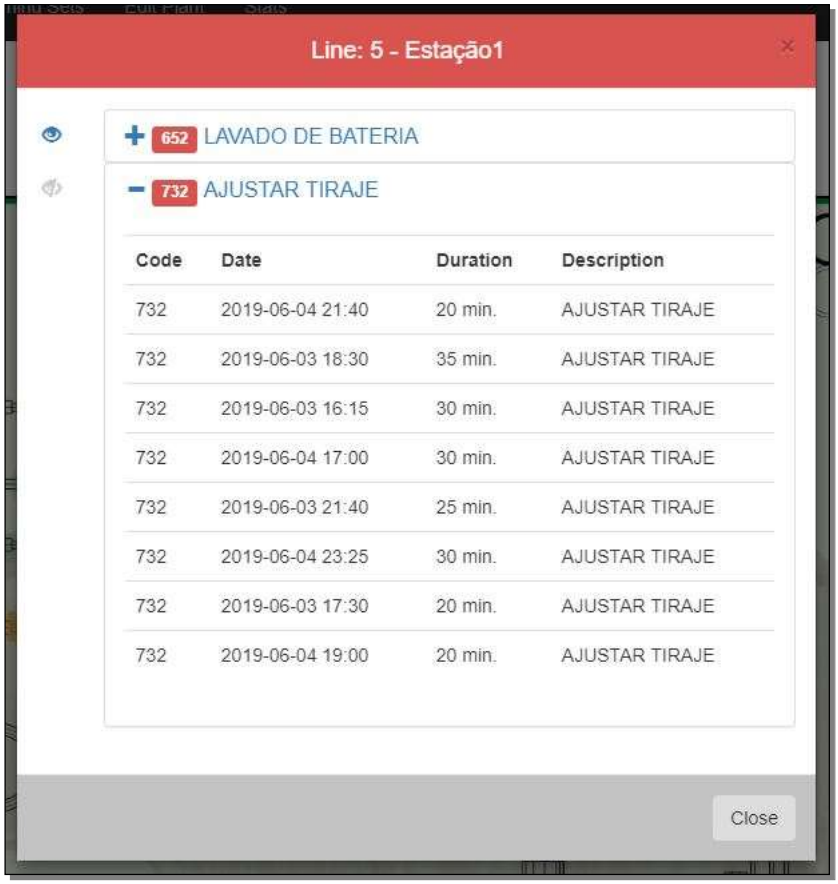

Fig. 14 - Dashboard Warning Disabled

4.5. Lines

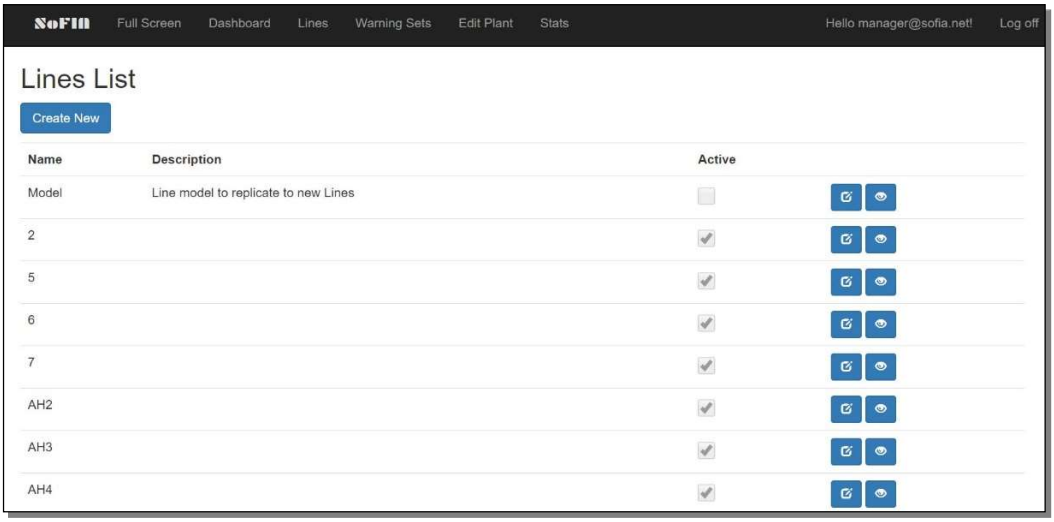

Fig. 15 - Lines Page

When accessing menu option Lines, it will be presented the list of production lines generated so far.

In this page it is possible to create a new line, by pressing button “Create New” in the top left corner, as well as see or edit existing lines by pressing respective edit  and view  buttons.

When selecting edit or create new, we navigate to a page with a form that allows us to edit line information.

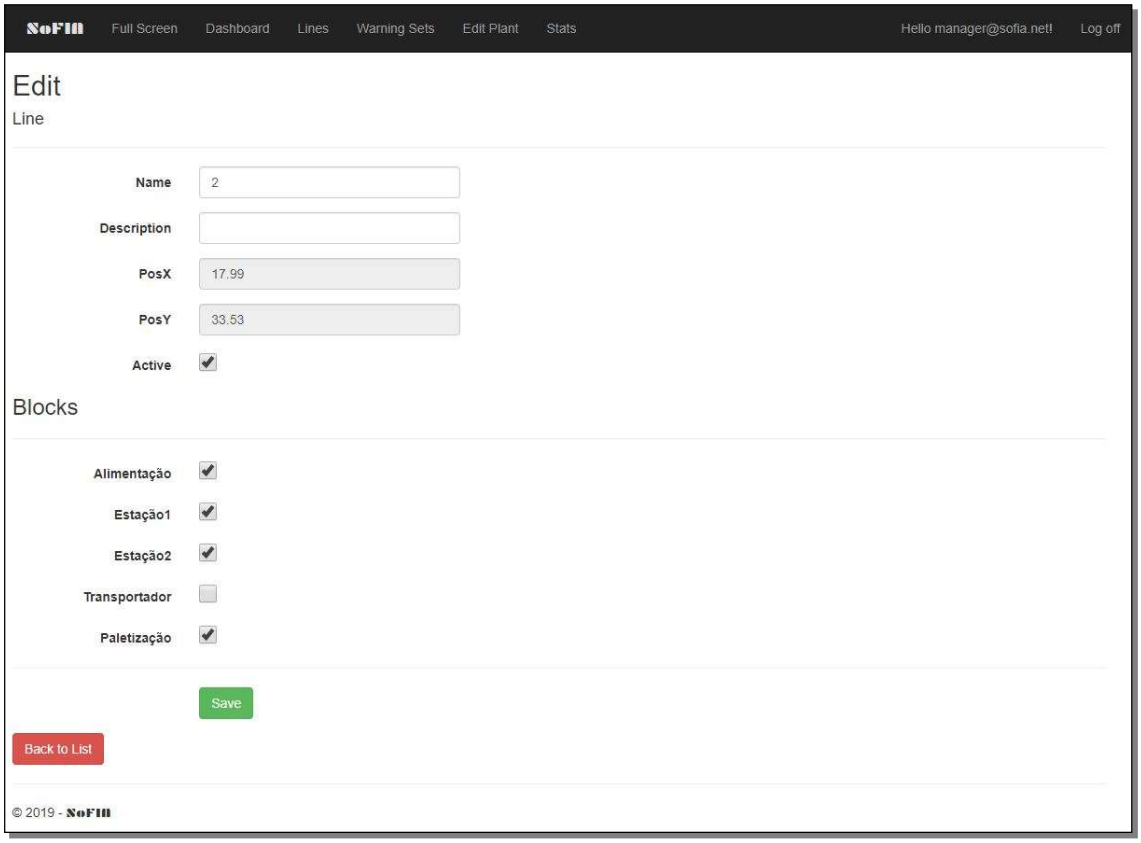

Fig. 16 - Line Edit

In this form we can change line name, line description, see cartesian coordinates (editing geographic coordinates is only possible in Edit Plant page). We can also enable or disable a line and set which blocks make part of it.

At the end of the form we have the options to save changes (Save Button) and navigate back to lines list without saving changes (Back to list button).

In the view page we can see not only the line properties but also an historic chart or warnings in the last 12 months.

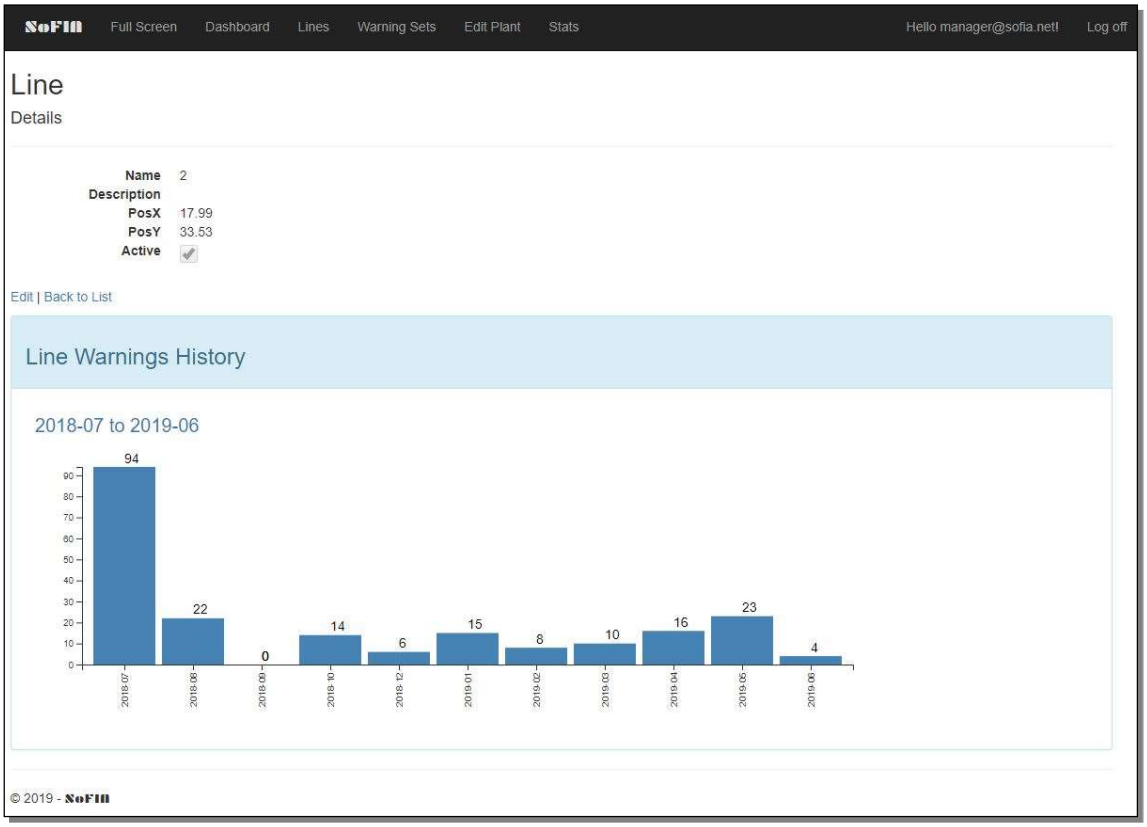

Fig. 17 - Line View

4.6. Warning Sets

When accessing Warning Sets menu option, we will access a page with a list of all warnings rules groups that define what triggers the warnings.

In this page we can navigate to a form that allows us to create a new warning, by pressing the **Create New** button on the top left corner.

We can also see the list of warnings created so far, and in each line we also have available the most significant properties of each warning set and access its edition.

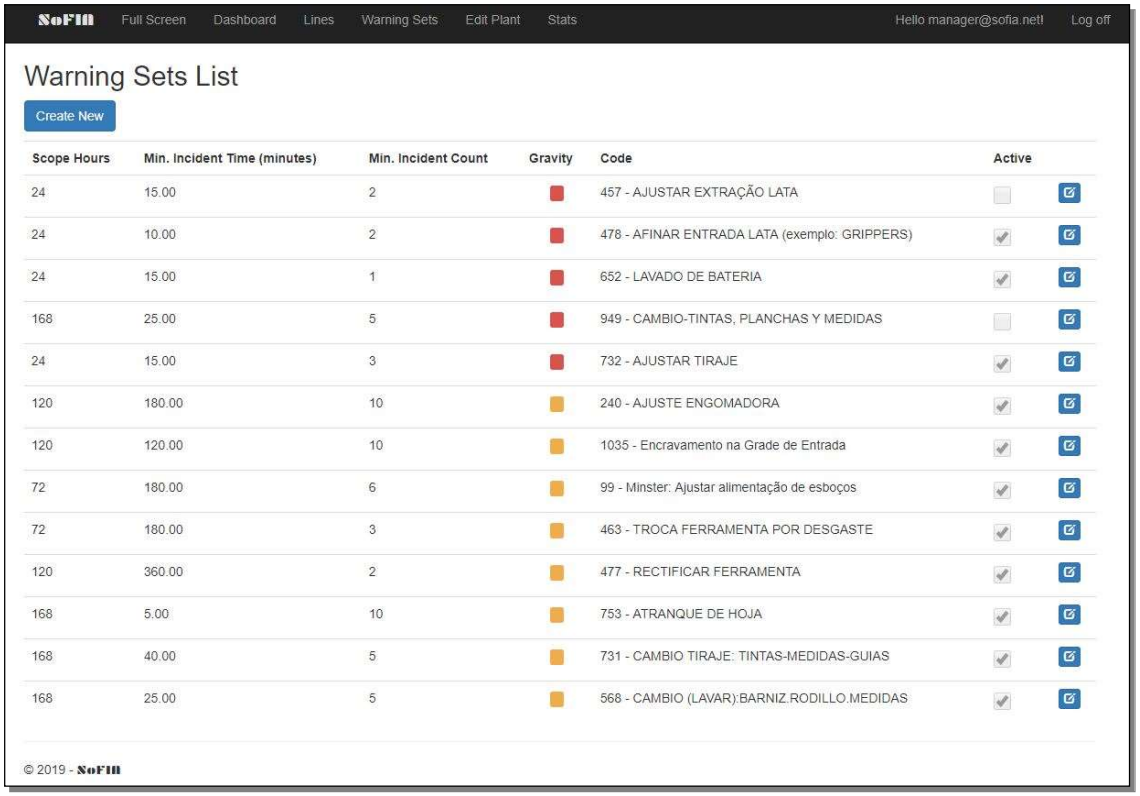

Fig. 18 - Warning Sets List Page

By pressing the edition button  at the end of the line we access the form to edit the context warning set.

When navigating to Create or edit a warning we are presented with a form to edit its properties (rules).

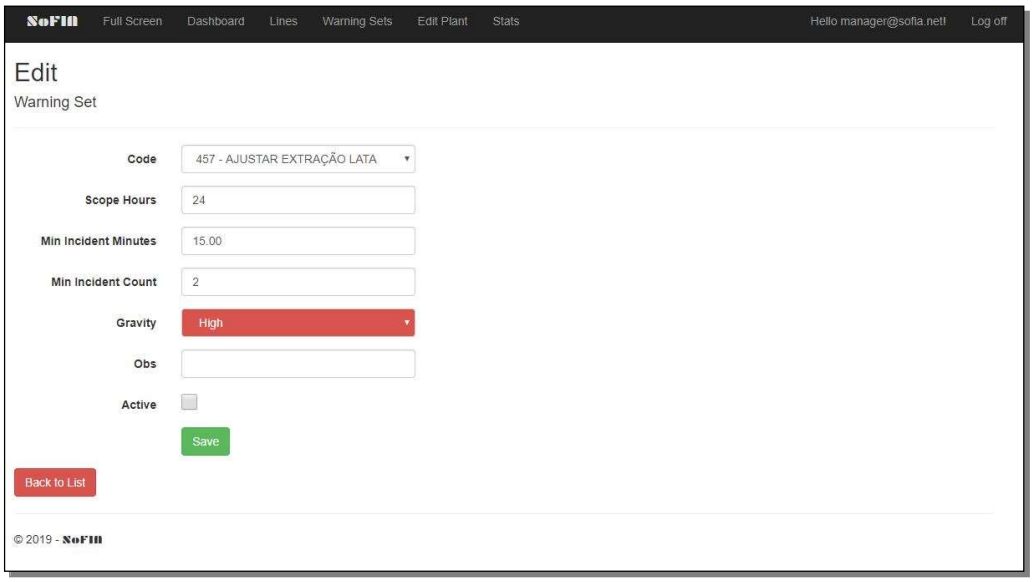

Fig. 19 - Warning Set Edit

In this form we will find the following fields:

- ***Code*** – incident code that will trigger the alarm;
- ***Scope Hours*** – time scope (in hours) in which it will be analyzed the presence of incidents defined in the previous field;
- ***Min. Incident Minutes*** – minimum stop time for an incident to be considered in this set of rules;
- ***Min. Incident Count*** – minimum incidents quantity to trigger a warning;
- ***Gravity*** – warning gravity, orange (high priority) or red (critical). The selected color will be applied in the dashboard when showing the warning;
- **O*bs*** – notes or other information useful to keep about this warning set;
- ***Active*** – only active warning sets will trigger new warnings.

At the end of the form we also have options to save changes (***Save*** button) or navigate back to warnings sets list without saving changes (***Back to list*** button).

The data saved in this page consists in the set of rules that are taken into consideration when processing daily incidents data. Every time the ETL process runs, each of this [Active] warning sets is checked against incidents data to determine if in the last [Scope Hours] hours before each incident, there were at least a number of incidents [Min. Incident Count] with a minimum duration [Min. Incident Minutes] with the same code [Code]. In case these rules find any matching criteria, a new warning is displayed in the corresponding color of its gravity [Gravity]. Only days where there were produced items are considered in what regards scope hours, meaning that for any day where there are no produced items the scope hours are extended backwards by 24h (e.g. Let's imagine we are processing incidents data gathered on December 2nd, checking a warning set with 48 scope hours. On the December 1st there were no produced items because it is holyday. So, instead of checking only for the last 48h, the scope hours are added by 24h, meaning that scope data will start on November 29th instead of November 30th) .

4.7. Edit Plant

Edit Plant menu option will lead us to the page where we can edit the warnings geographic positioning on the top of plant background. We can set positioning for lines and its blocks.

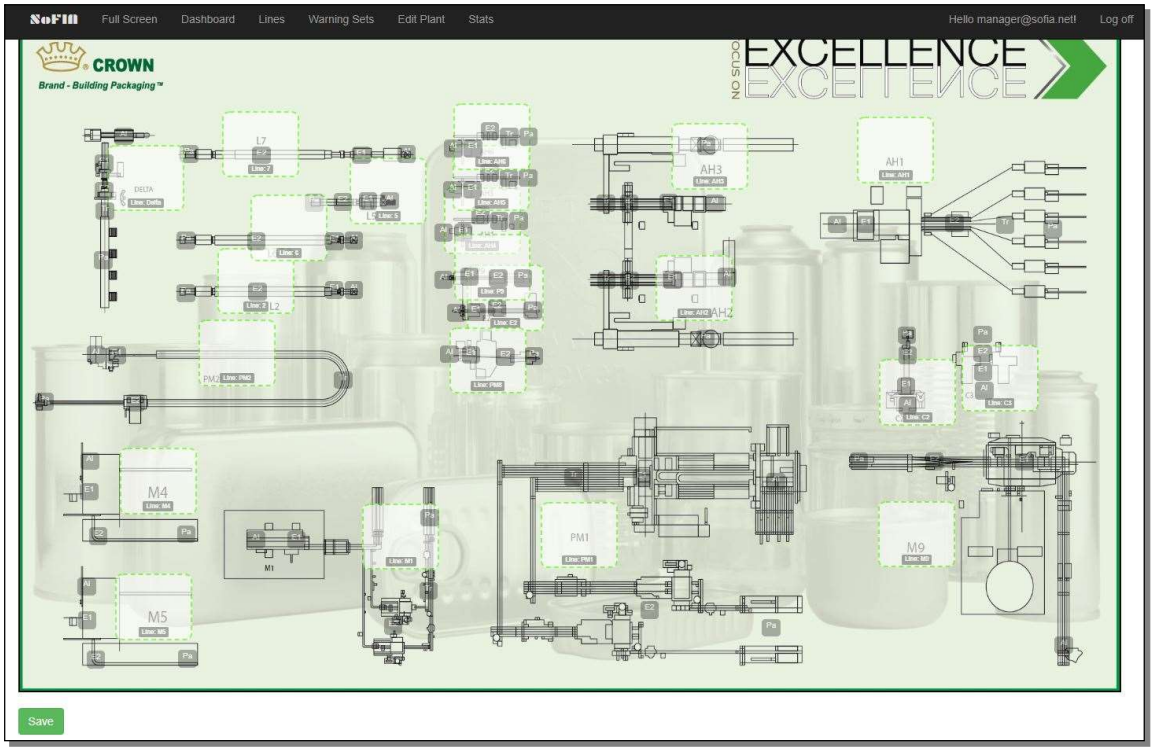

Fig. 20 - Edit Plant Page

As background image we will have the factory plant, it is possible to set production lines positioning by dragging them to where we want, when dragging a production line all its blocks will be dragged together. Each block is also draggable on its own if we only select a block to move instead of the production line.

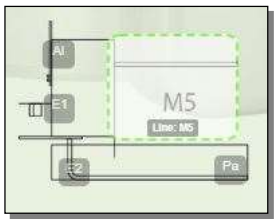

Fig. 21 - Edit Plant - Line and Blocks

In the image above it is possible to identify the production line (M5 white square) and the 4 blocks that gravitate around it (A1, E1, E2 e Pa).

The cartesian coordinate reference point for each line is the bottom left corner of the image (coordinates x=0, y=0), while the geographic reference point for each block is the center of the production line they belong to.

4.8. Stats

Fig. 22 - Stats Page

When accessing ***Stats*** menu option we will navigate to a page where we can see a date selector and a set of charts. These charts will show the following information, regarding the period selected:

- *Total Warnings* – Total warnings quantity generated in this period and the homologous period in the previous year.
- T*otal Warnings Per Production* – Average of warnings generated for each 100.000 items produced for the selected period and the homologous period in the previous year;
- *Line Warnings Top 10* – 10 lines that triggered more warnings in this period;
- *Block Warnings Top 10* – 10 blocks that triggered more warnings in this period.

Chapter 5

Conclusions and Limitations Analysis

The developed work ended up with a completely functional pilot, deployed and in permanent use, within Crown factory facilities. Crown acquired the required hardware (64” screen) and set up the software environment (servers) where the system is now deployed. All the requirements identified as **Must Have**, **Should Have** and **Could Have** were successfully implemented.

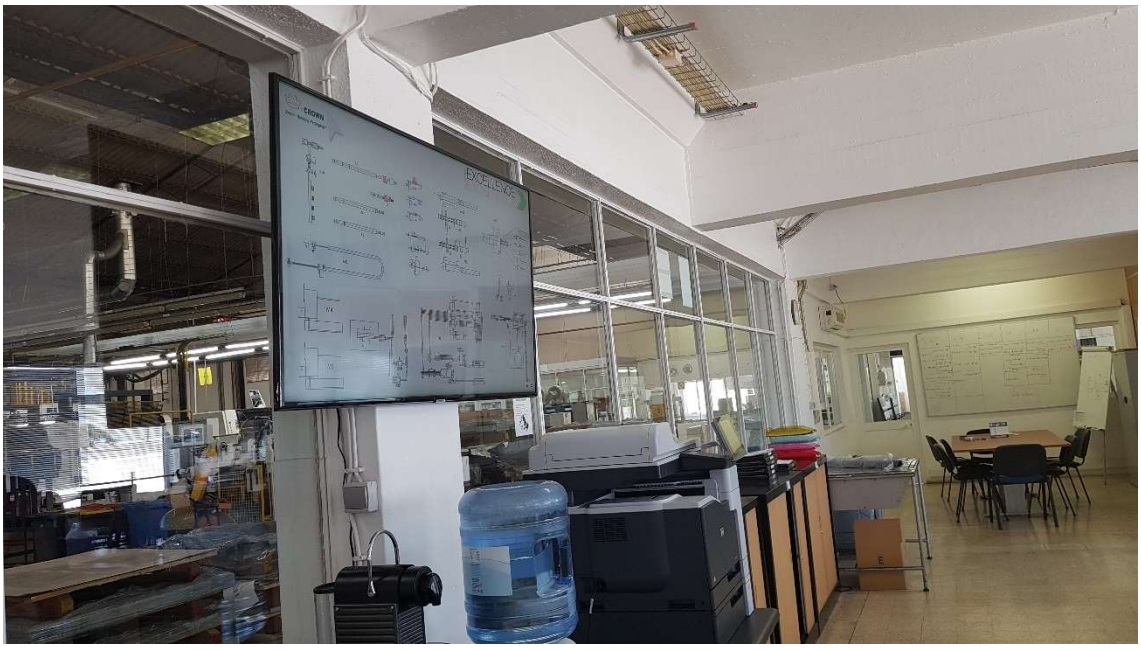

Fig. 23 - Project in production environment

Regarding limitations, it is important to mention that having the information available in real time is a requirement that makes sense, in the application context, right now is presented with a lag of 24 hours. However, considering the time gap between the moment when the data is collected (written by line operators on physical device, paper and gathered trough systems) and the moment when this data enters the system, after standardization and digitalization, achieved by supervision, prevents achieving that goal at the moment.

It was suggested a second phase, that focus mostly on physical device data dematerialization, passing through the initial data gathering being made on digital format,

supported by an application developed for that purpose, that can be executed for instance in a tablet device.

Additionally, it should be considered the adoption of sensors, connected to the production lines, that can identify incident codes that cause line stops, and make this data available for the system in real time, reducing the data quantity that has to be reported by operators, and consequently, making this data less prone to error.

Bibliography

Bootstrap. (2013). *Documentation.* Retrieved from https://getbootstrap.com/docs/3.3/

Bostock, M. (2016). *D3 Data-Driven Documents.* Retrieved from https://d3js.org/

Inmon, W. H., Imhoff, C., & Sousa, R. (2001). *Corporate Information Factory.* Wiley. Retrieved from https://books.google.pt/books?id=IzSOzX_7cr4C

Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (2nd ed.).* New York, USA: John Wiley & Sons, Inc.

Larson, D., & Chang, V. (2016). *A review and future direction of agile, business intelligence, analytics and data science.* International Journal of Information Management, 36(5), 700–710. Retrieved from http://doi.org/10.1016/J.IJINFOMGT.2016.04.013

Microsoft. (2016). *Entiity Framework documentation.* Retrieved from https://docs.microsoft.com/en-us/ef/ef6/

Microsoft. (2017). *LINQ documentation.* Retrieved from https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/

Microsoft. (2018). *MVC.net documentation.* Retrieved from https://docs.microsoft.com/pt-pt/aspnet/mvc/mvc5