FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Sunday, March 20 2020

Scope: 360º Company Dashboard

Information Systems
2020/2021

GROUP M:

Diogo Machado	up201706832@fe.up.pt
Gonçalo Marantes	up201706917@fe.up.pt
João Cardoso	up201705149@fe.up.pt
Paulo Marques	up201705615@fe.up.pt
Raul Viana	up201208089@fe.up.pt

Index

Introduction

Scope is a web application that provides our hardware and home improvement retail company Scope Bricolage with a thorough report, complete with information about sales, inventory status, and overall balance over time. Besides this it also offers detailed information about individual clients, suppliers and products.

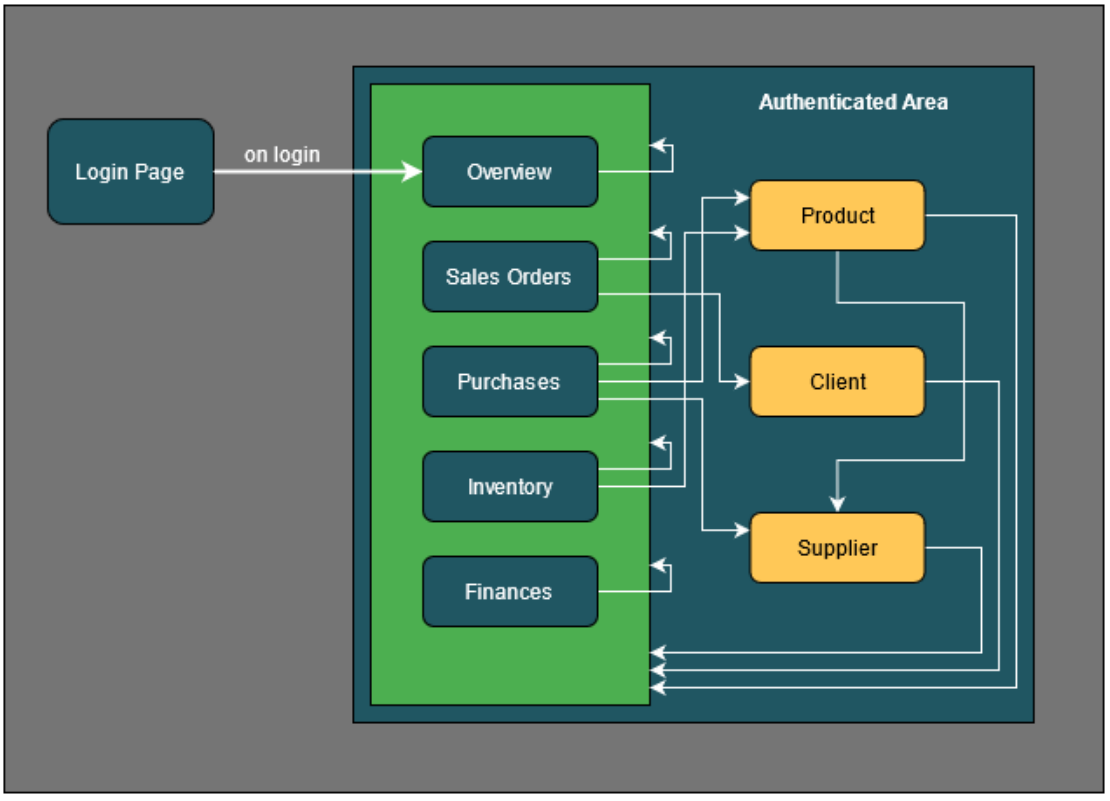
Project Overview

For this project, the goal was the development of a dashboard to provide data and key performance indicators (KPIs) relevant to our client's business.

To achieve this, we built a web application which has a great focus on the comprehensive visualization of the large amount of data inherent to the business, in order to assist our client by making their decision process easier.

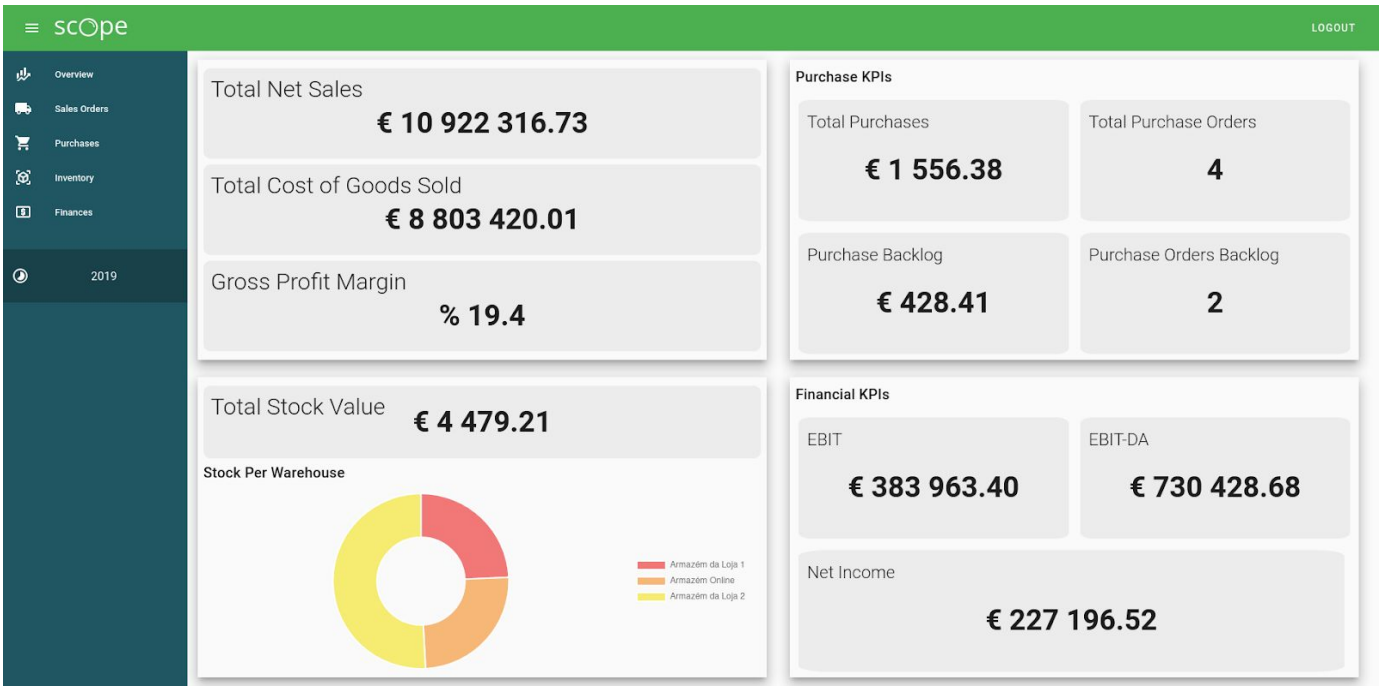
The dashboard displays information regarding the financial performance of the company in several areas, such as Sales Orders, Purchases, Inventory, and Financial information, as well as an Overview of the main KPIs which allow us to get a broader picture of the company's financial status. For this, it relies on and inter-operates with the *Primavera ERP*, while all KPI calculations are handled by our own backend.

In addition, we implemented a secure authentication layer, also handled by our backend service, which ensures that this sensitive information cannot be accessed by outsiders.



Information Architecture

Overview



User and Business Goals

Company Status Overview (Inventory flow, Sales and Cash Flow overview)

Inward Paths/Trigger Words

Sign In
Register
Navigation Menu

Outward Paths/Call to Action

- user has a company overall view
- change the information time interval
- dive in details about all domains

Elements of the Core

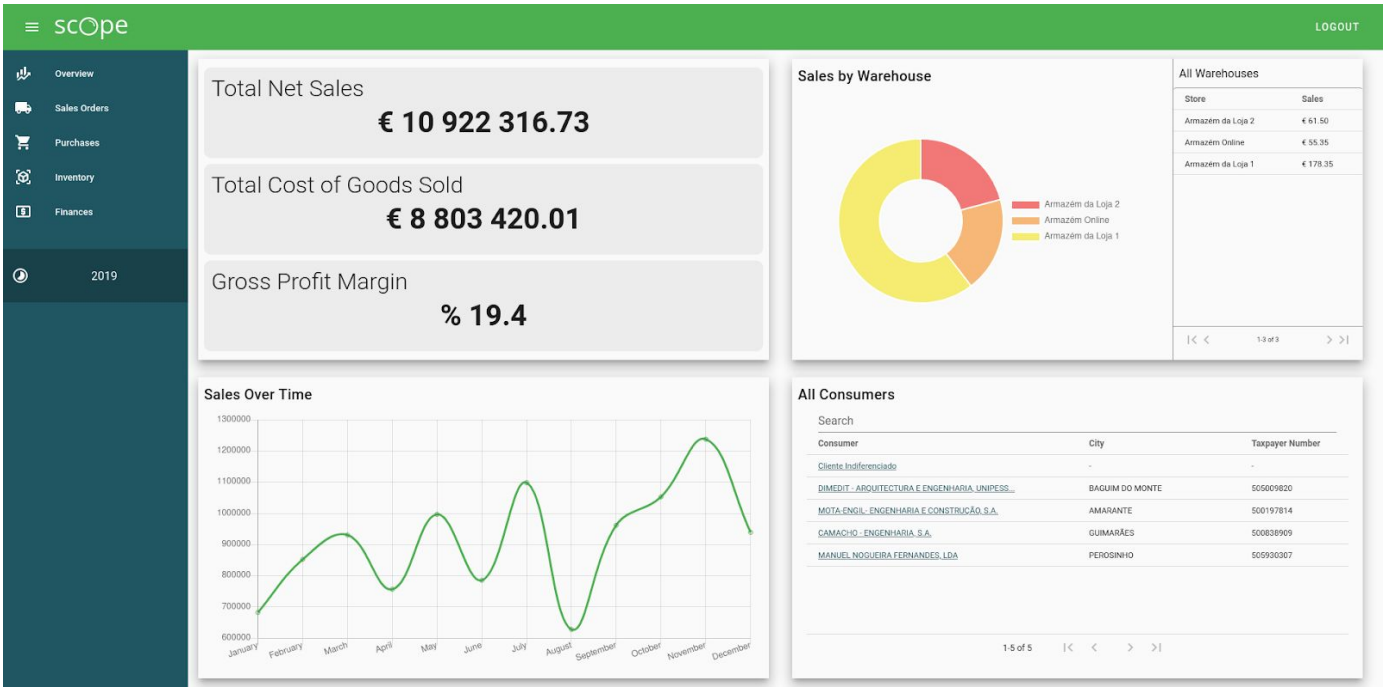
KPI_01(KPI) - Total Net Sales
KPI_02(KPI) - Total Cost of Goods Sold
KPI_03(KPI) - Gross Profit Margin

KPI_04(KPI) - Total Purchases
KPI_05(KPI) - Total Purchase Orders
KPI_06(KPI) - Purchase Backlog
KPI_07(KPI) - Purchase Orders Backlog

KPI_08(KPI) - Total Stock Value
DOUGHNUT_01(Doughnut Chart) - Stock per Warehouse

KPI_09(KPI) - EBIT
KPI_10(KPI) - EBITDA
KPI_11(KPI) - Net Income

Sales Orders



User and Business Goals

Company Sales (profit from sales, segmented sales by store, clients and time period)

Inward Paths/Trigger Words

Navigation Menu

Outward Paths/Call to Action

- detailed view about sales
- change information time interval
- Client drilldown
- Product drilldown

Elements of the Core

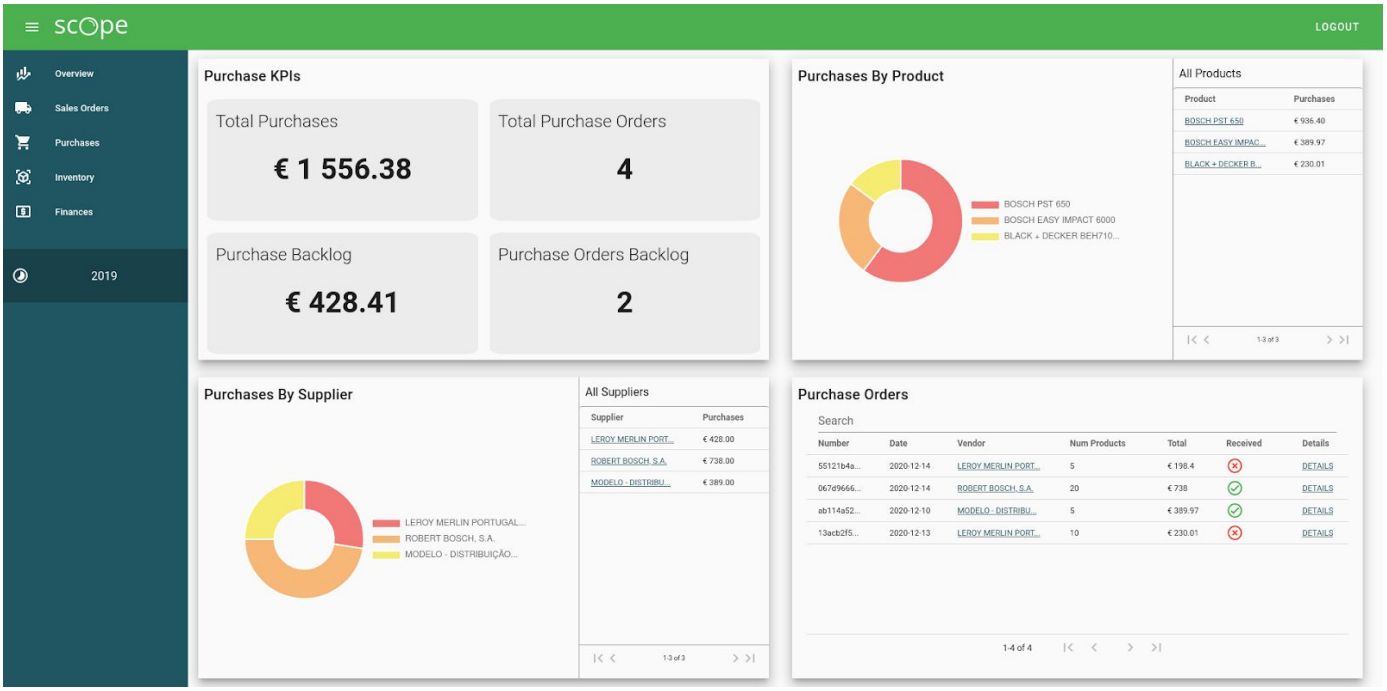
KPI_01(KPI) - Total Net Sales
KPI_02(KPI) - Total Cost of Goods Sold
KPI_03(KPI) - Gross Profit Margin

DOUGHNUT_01(Doughnut Chart) - Sales by Warehouse
TABLE_01(Table) - Sales by Warehouse

LINE_01(Line Chart) - Sales Over Time

TABLE_02(Table) - All Consumers

Purchases



User and Business Goals

Company Purchases (Purchases in numbers, by product, by supplier and most recent)

Inward Paths/Trigger Words

Navigation Menu

Outward Paths/Call to Action

- detailed view about purchases
- change information time interval
- Supplier drilldown
- Product drilldown

Elements of the Core

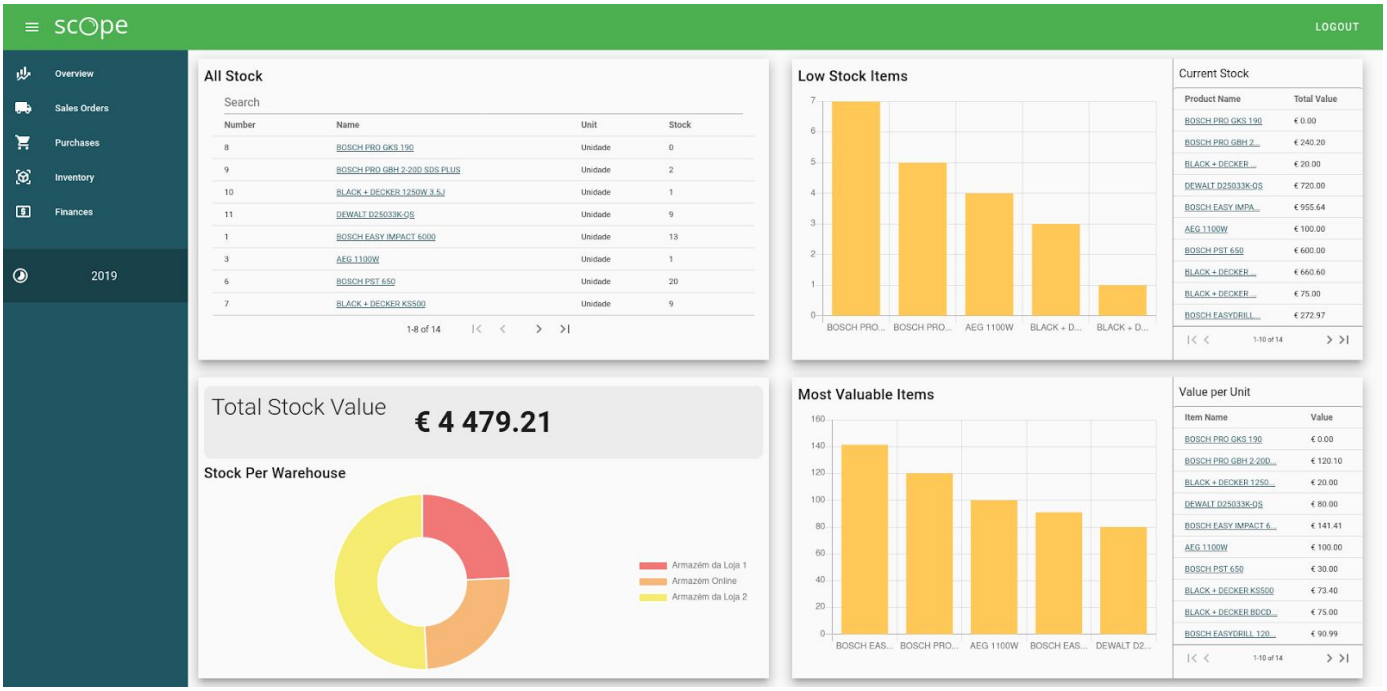
KPI_01(KPI) - Total Purchases
KPI_02(KPI) - Total Purchase Orders
KPI_03(KPI) - Purchase Backlog
KPI_04(KPI) - Purchase Orders Backlog

DOUGHNUT_01(Doughnut Chart) - Purchases By Product
TABLE_01(Table) - Purchases By Product

DOUGHNUT_02(Doughnut Chart) - Purchases By Supplier
TABLE_02(Table) - Purchases By Supplier

TABLE_03(Table) - Purchase Orders

Inventory



User and Business Goals

Company Inventory (products stock variation, stock value, low stock items)

Inward Paths/Trigger Words

Navigation Menu

Outward Paths/Call to Action

- detailed view about inventory
- change information time interval
- Product drilldown

Elements of the Core

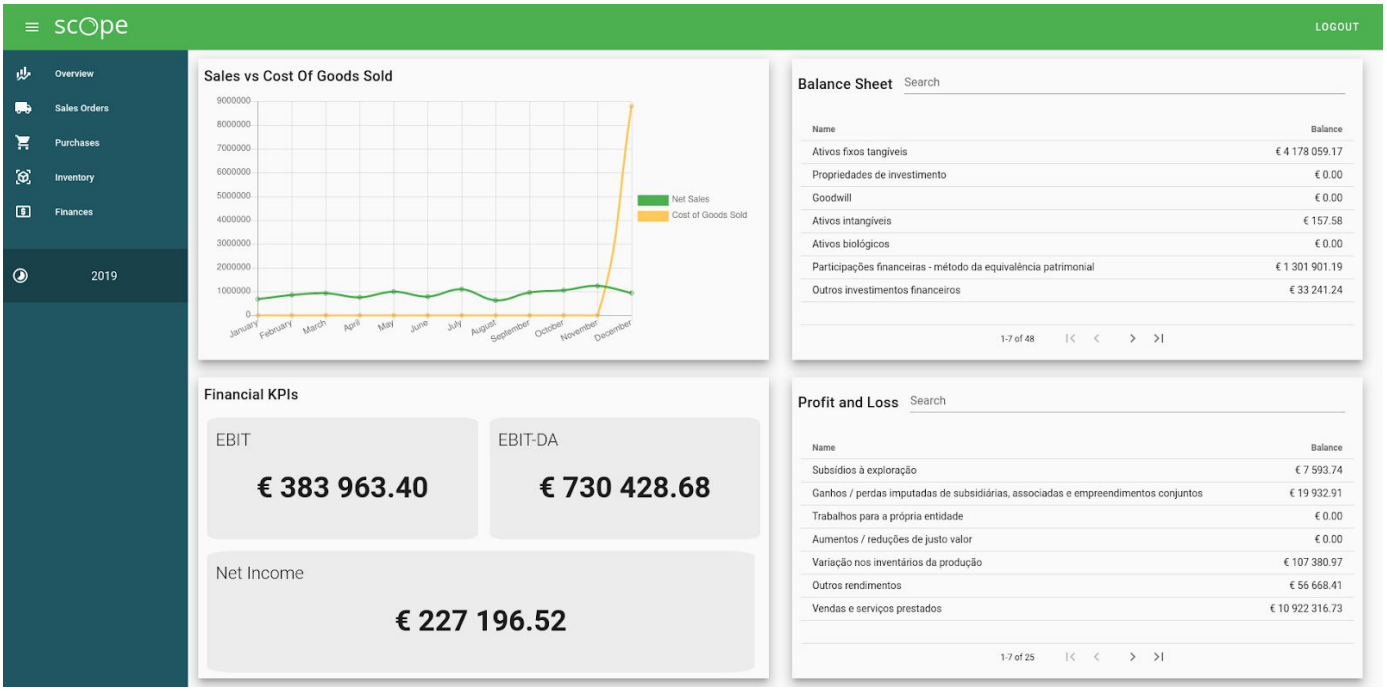
TABLE_01(Table) - All Items Stock

BAR_01(Bar Chart) - Low Stock Items
TABLE_01(Table) - Low Stock Items

KPI_01(KPI) - Total Stock Value
DOUGHNUT_01(Doughnut Chart) - Stock Per Warehouse

BAR_02(Bar Chart) - Most Valuable Items
TABLE_02(Table) - Most Valuable Items

Finances



User and Business Goals

Company financial health (EBIT, EBITDA, Accounts receivable and payable)

Inward Paths/Trigger Words

Navigation Menu

Outward Paths/Call to Action

- detailed view about financial details
- change information time interval
- balance details drilldown

Elements of the Core

LINE_01(Line Chart) - Sales vs Cost Of Goods Sold

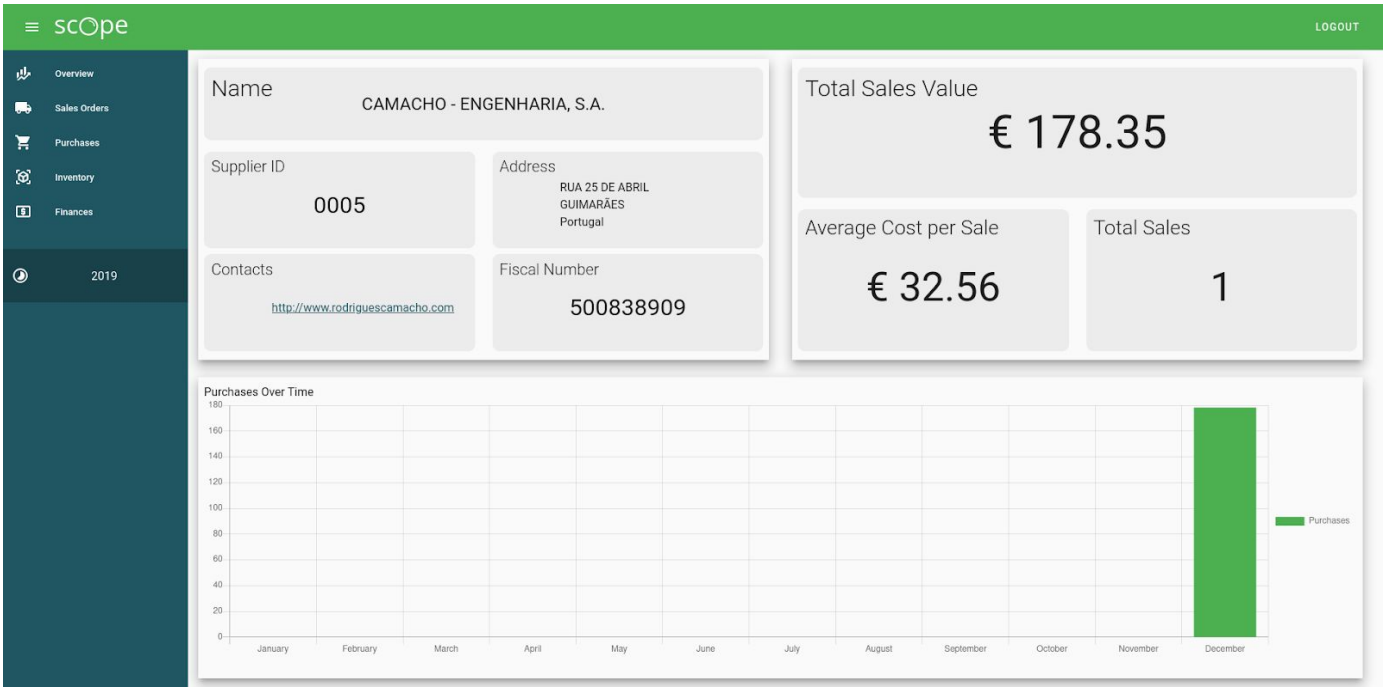
TABLE_01(Table) - Balance Sheet

KPI_01(KPI) - EBIT
KPI_02(KPI) - EBITDA
KPI_03(KPI) - Net Income

TABLE_02(Table) - Profit and Loss

Drill Downs

Client



User and Business Goals

Client contact information and financial indicators

Inward Paths/Trigger Words

Sales Orders

Outward Paths/Call to Action

- change information time interval

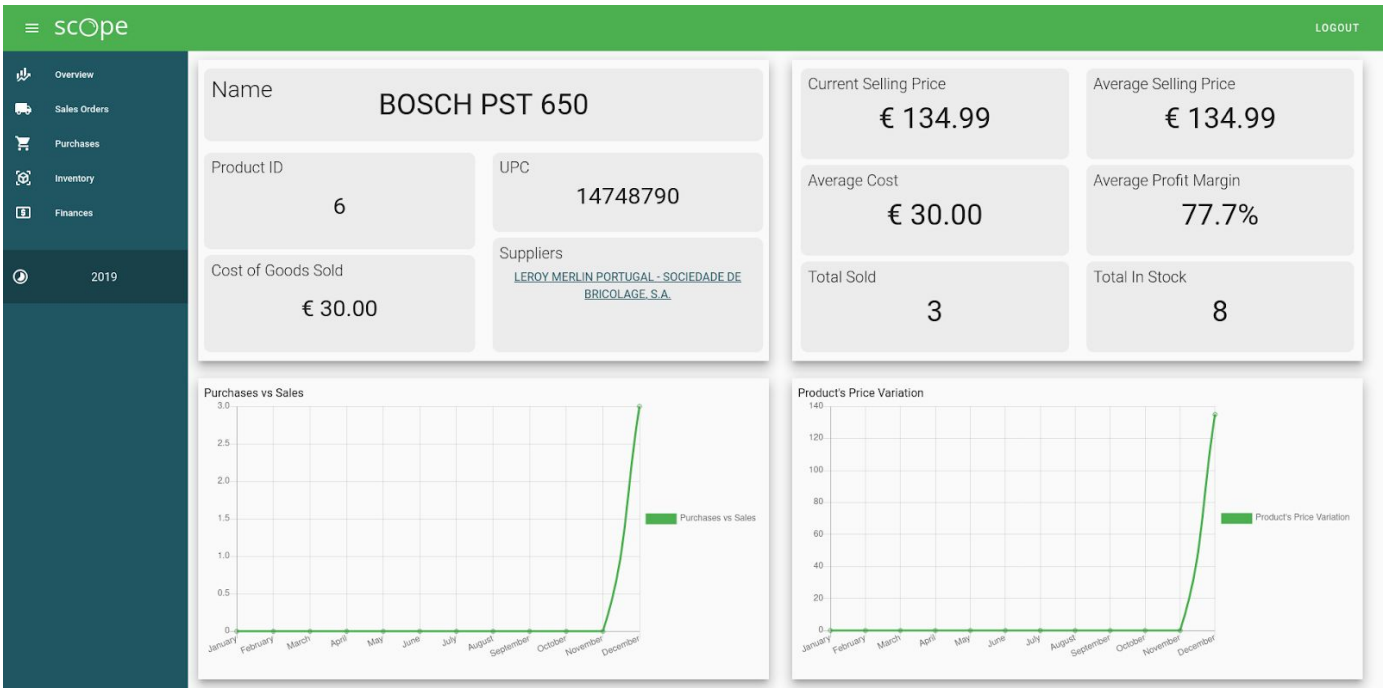
Elements of the Core

INFO_01(Information Box) - Name
INFO_02(Information Box) - Supplier ID
INFO_03(Information Box) - Address
INFO_04(Information Box) - Contacts
INFO_05(Information Box) - Fiscal Number

KPI_01(KPI) - Total Sales Value
KPI_02(KPI) - Average Cost per Sale
KPI_03(KPI) - Total Sales

BAR_01(Bar Chart) - Purchases Over Time

Product



User and Business Goals

Product information and financial indicators

Inward Paths/Trigger Words

Purchases
Inventory

Outward Paths/Call to Action

- change information time interval
- Supplier drilldown

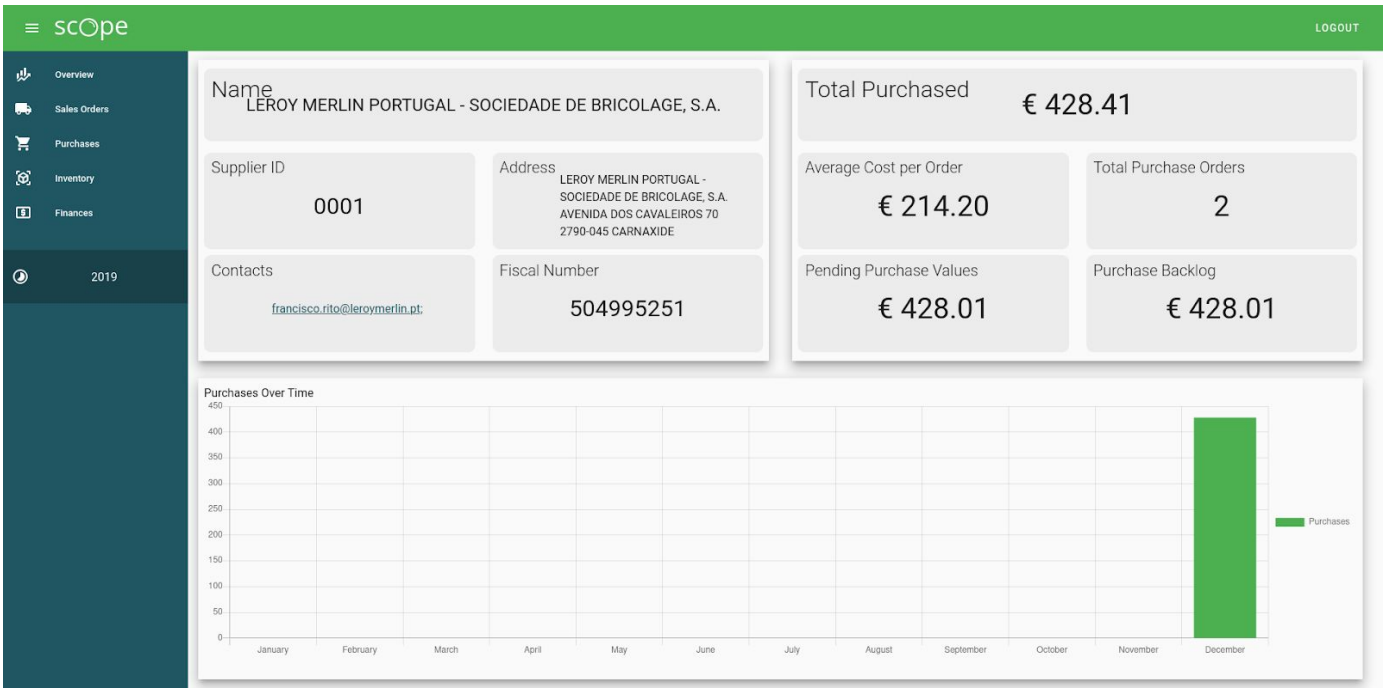
Elements of the Core

INFO_01(Information Box) - Name
INFO_02(Information Box) - Product ID
INFO_03(Information Box) - UPC
INFO_04(Information Box) - Cost of Goods Sold
INFO_05(Information Box) - Suppliers

KPI_01(KPI) - Current Selling Price
KPI_02(KPI) - Average Selling Price
KPI_03(KPI) - Average Cost
KPI_04(KPI) - Average Profit Margin
KPI_05(KPI) - Total Sold
KPI_06(KPI) - Total In Stock

LINE_01(Line Chart) - Purchases vs Sales
LINE_02(Line Chart) - Product’s Price Variation

Supplier



User and Business Goals

Supplier contact information and financial indicators

Inward Paths/Trigger Words

Purchases
Product

Outward Paths/Call to Action

- change information time interval

Elements of the Core

INFO_01(Information Box) - Name
INFO_02(Information Box) - Supplier ID
INFO_03(Information Box) - Address
INFO_04(Information Box) - Contacts
INFO_05(Information Box) - Fiscal Number

KPI_01(KPI) - Total Purchased
KPI_02(KPI) - Average Cost per Order
KPI_03(KPI) - Total Purchase Orders
KPI_04(KPI) - Pending Purchase Values
KPI_05(KPI) - Purchase Backlog

BAR_01(Bar Chart) - Purchases Over Time

Features and Functionalities

SAF-T Complexity and Relational Database

The Standard Audit File for Tax purposes is used for documenting accounting records, and by using the XML syntax it is easily exported and used across multiple platforms. Although this file still needs some degree of manipulation, in order to find specific information.

Since we are going to be using this file a lot, we decided to transcribe all the information present into a relational database, where data access and modification is a lot easier and faster.

Financial Analysis and KPIs

In order to calculate our company's balance sheet and income statement, we need to use account taxonomies to identify transaction types, i.e where our money is being spent. In order to simplify calculations we have created a Taxonomy Table in our database. Each row in this table uniquely identifies a Taxonomy Code, and stores information about its opening debit and credit balance as well as the sum of all credit and debit transactions for all accounts with that particular Taxonomy Code.

Data Visualization

We can clearly see that complex data manipulation is a recurring issue in our project. Hence, it is necessary to show that data in a human friendly way. Our main strategy was developed around using charts to showcase data relation and over time events, such as sales or purchases, as well as simple searchable tables. To implement this strategy we have decided to use tables from Vuetify UI component library and charts from Chart.js library.

Authorization and Security

It is important to note that all the data shown in the client's Dashboard is sensitive and should not be accessible to outsiders. For this reason, we have used the JSON Web Token protocol and implemented a session based authentication. Since our server acts as a middleman between the client and the data, it is the server's responsibility to authenticate the client.

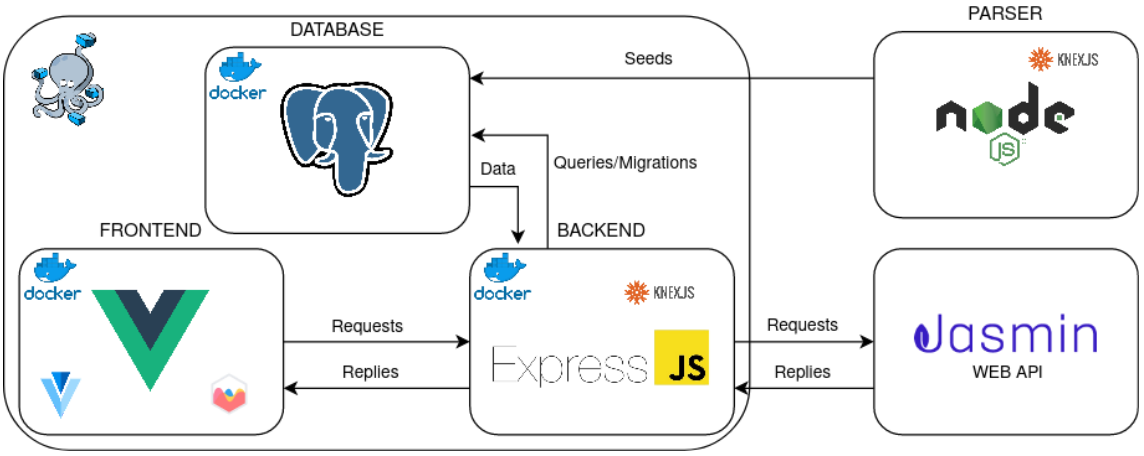
Fluid User Experience

When dealing with large amounts of data, it is important not to tax the client. In order to offer a fluid User Experience for our users all requests and responses are handled in the backend. Since Jasmin's JSON response is complex and has a lot of information with no interest for a particular request, it acts like a middleware between the client and the

Jasmin API. The server parses Jasmin response and delivers just the needed information to the client. This allows a more fluid client usability.

System Architecture and Technologies

Our system is divided into 4 core parts: client (frontend), server (backend), database and SAF-T file parser. In order to correctly replicate the development environment in any system we have decided to containerize our application using Docker and Docker-Compose, not counting the parser due to it only being run once for database population.



Frontend

For the client side, our system uses Vue.js for the main User Interface building framework. Besides Vue, we also make use of Vuetify as the UI component library and ChartJS to show data graphically.

Backend

Our server is running an Express.js server, and using Knex.js as its Object-relational mapping service for Database migrations, seeding and queries. The server is also responsible for authentication and authorization and serves as a middleware between the client and the Jasmin WEB API.

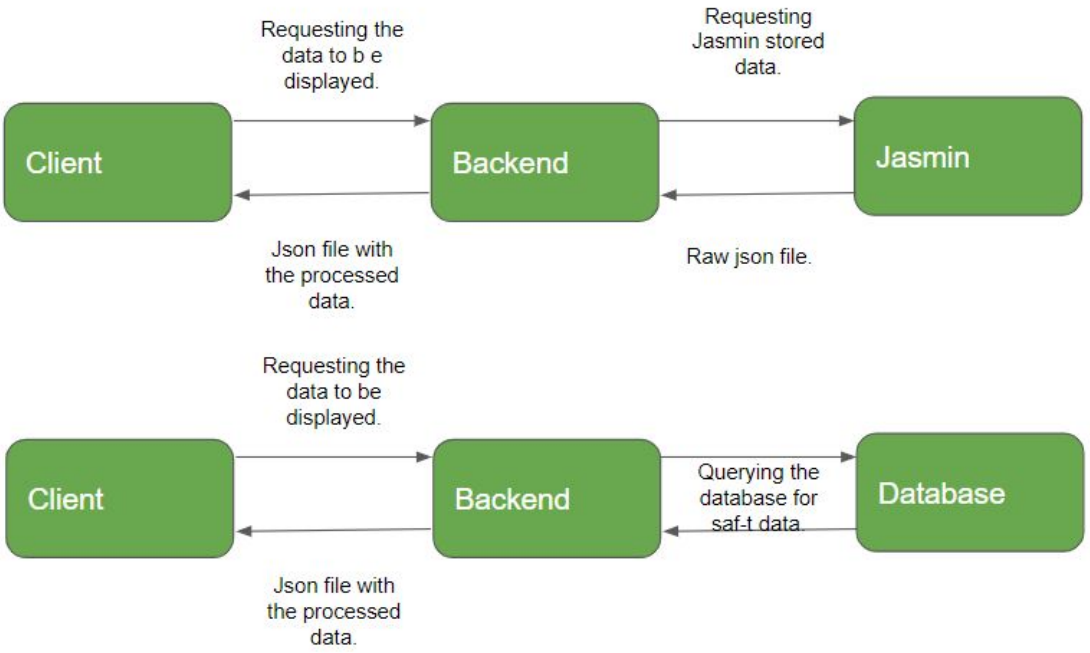
Database

For the database we have decided to use a relational database, and for that reason PostgreSQL was chosen.

Parser

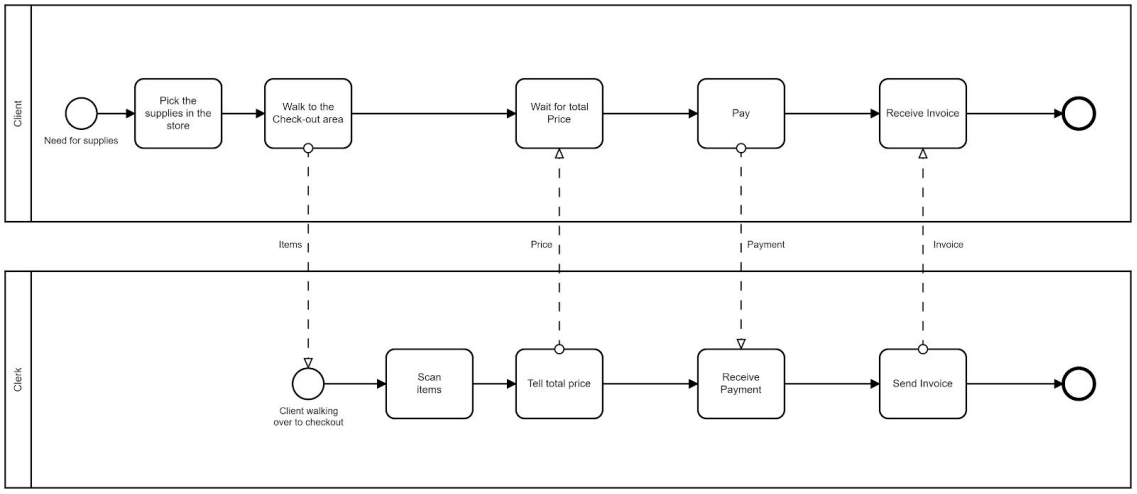
Since all our other services are using JavaScript as the main programming language, and Knex.js is already set up for the server, we have decided to follow a similar approach for our SAF-T file parser. This service is responsible for reading the provided file and seeding the database accordingly.

KPI Information Flow Example

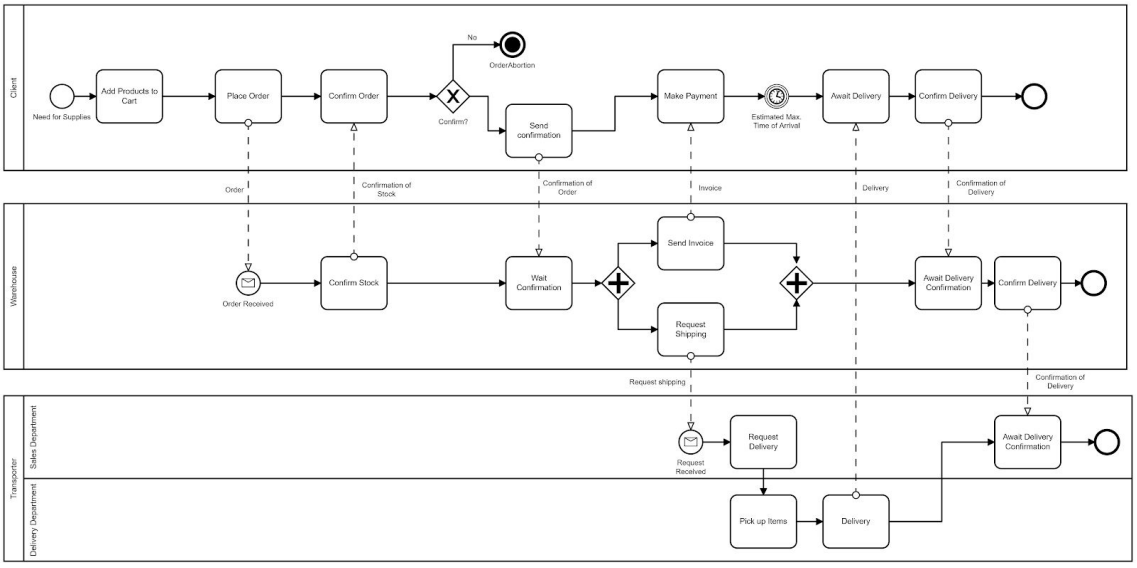


BPMN Flows

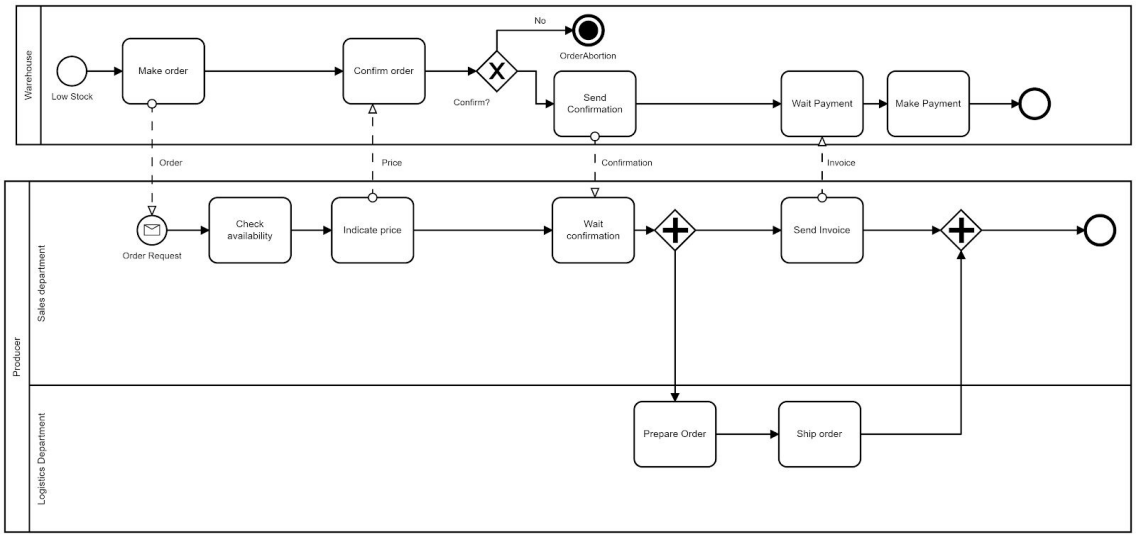
Purchase in-store



Purchase online

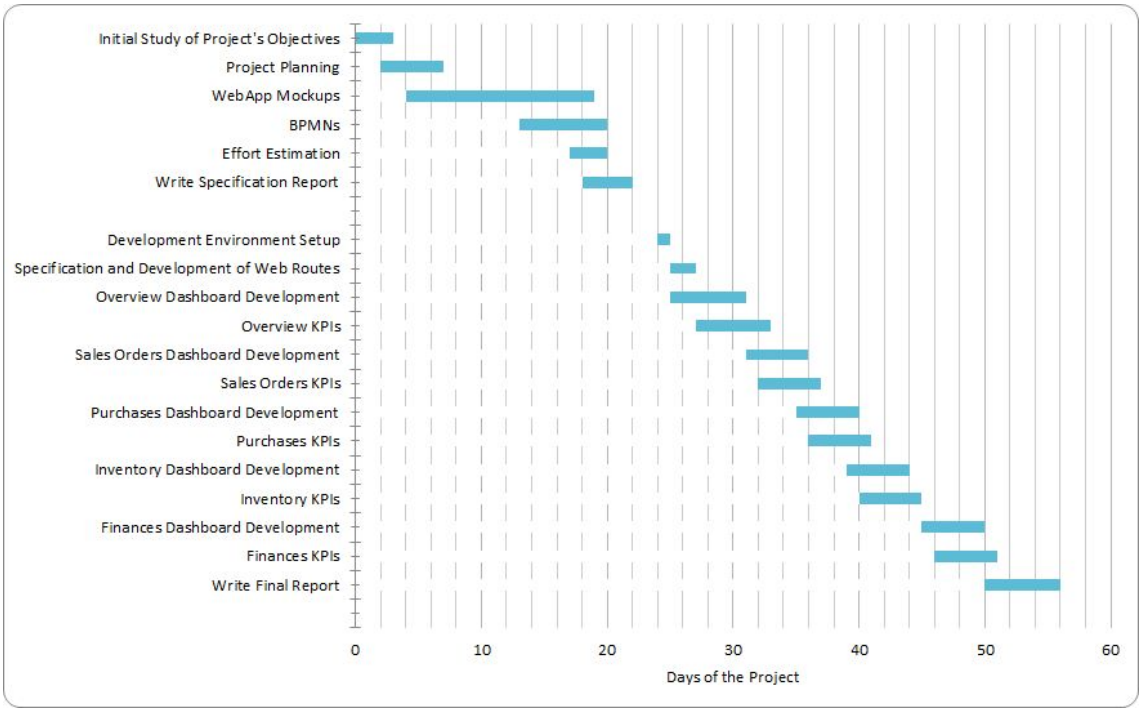


Restock


Warehouse
Low Stock
Make order
Confirm order
Confirm?
No
OrderAbortion
Send Confirmation
Wait Payment
Make Payment
Order
Price
Confirmation
Invoice
Producer
Sales department
Order Request
Check availability
Indicate price
Wait confirmation
Send Invoice
Logistics Department
Prepare Order
Ship order

Planning (Gantt Project)

TASK NAME	START DATE	END DATE	START ON DAY*	DURATION* (WORK DAYS)	TEAM MEMBER	PERCENT COMPLETE
Functional Specification						
Initial Study of Project's Objectives	10/18	10/20	0	3		0%
Project Planning	10/20	10/24	2	5		0%
WebApp Mockups	10/22	11/5	4	15		0%
BPMNs	10/31	11/6	13	7		0%
Effort Estimation	11/4	11/6	17	3		0%
Write Specification Report	11/5	11/8	18	4		0%
App/Product Deliver						
Development Environment Setup	11/11	11/11	24	1		0%
Specification and Development of Web Routes	11/12	11/13	25	2		0%
Overview Dashboard Development	11/12	11/17	25	6		0%
Overview KPIs	11/14	11/19	27	6		0%
Sales Orders Dashboard Development	11/18	11/22	31	5		0%
Sales Orders KPIs	11/19	11/23	32	5		0%
Purchases Dashboard Development	11/22	11/26	35	5		0%
Purchases KPIs	11/23	11/27	36	5		0%
Inventory Dashboard Development	11/26	11/30	39	5		0%
Inventory KPIs	11/27	12/1	40	5		0%
Finances Dashboard Development	12/2	12/6	45	5		0%
Finances KPIs	12/3	12/7	46	5		0%
Write Final Report	12/7	12/12	50	6		0%



Conclusion

This project was definitely a challenge to complete, we were introduced to completely new concepts that are not so easy to comprehend at first glance. Nonetheless we rose up to the challenge and delivered a working application that adds value to any company wanting to get a birds eye view of their financial situation.

We have also used this opportunity to develop our application using new tools we haven't used before, such as VueJS, which in the long run proved itself to be a useful tool. One of the major difficulties we had was understanding the account taxonomy system and how to calculate KPIs based on account movements. But with the help of our teacher and an SQL Database with accounting data, this task was made simple.

Summing up, we believe that we have finished this project successfully and achieved all the established goals, leading to the delivery of a tool that would be useful in any company.