P3 - 360º Company Dashboard

FINAL REPORT

Master in Informatics and Computing Engineering

Information Systems

Group H:
Daniela Quintas Fernandes de Sá - up201405457
Hélder Manuel Mouro Antunes - up201406163
Inês Filipa Noronha Meneses Gomes Proença - up201404228
João Filipe Pereira da Costa - up201403967

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

December 17, 2017

Contents

1 Overview

The purpose of this report is to demonstrate and explain the implementation of the 360th Company Dashboard project developed during the semester.

A website has been developed that provides complete information about the state of a business (sales, purchases, inventory and finances). The dashboard is communicates with Primavera ERP via web-services and the SAFT structure. Lines and pie charts and other tables were used to show the information in a more user friendly way.

This website will be based on known business intelligence tools like SAP Business Objects, QlikView and Power BI, which will serve as a reference for what statistics are relevant for a business.

In the next chapters the functionalities implemented, interoperability with Primavera and other details related to project implementation will be shown.

2 Core Views

For the purpose of providing a dashboard with complete information about the state of a company, the group prepared six core views. This are the core features of the project and are vital to the usability of the website.

2.1 Overview (V_01)

This dashboard, accessible at all times through the top navigation bar, provides a general idea of the state of my company with key performance indicators in the sales, purchases, inventory and financial area. All the information shown pertain to the time-frame selected.

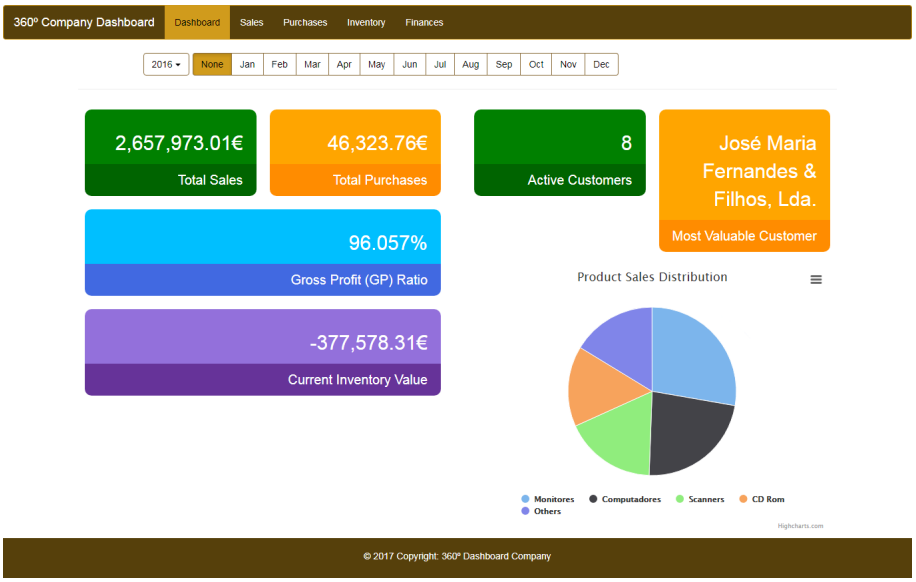

Figure 1: Overview dashboard

In the sales department, this dashboard shows the total amount spent on sales, the number of active customers, the customer who spent the most money on the business and a pie chart showing the distribution of sales per product category.

In terms of purchases, the overview only informs the total amount spent on purchases in the period selected.

For financial information, the gross profit ratio, which is computed by dividing the gross profit figure by net sales, is shown.

Similarly for inventory, there is a current inventory value key performance indicator, which illustrates the amount of money stored as inventory.

This page does not provide any specific drill down. However, the top navigation lets the user access any of the other area specific dashboards, which have more detailed information.

2.1.1 Primavera Webservices used:

- getPurchasesGroupedBySupplier

- getInventory

2.1.2 Webservices of our server used:

- /getSalesByYear
- /getSalesByMonth
- /getCustomers
- /getSalesByProductGroup

2.2 Purchases Dashboard (V_02)

The purchases dashboard, which can be reached at all times through the top navigation bar, provides an overview of the purchases department in the time-frame selected.

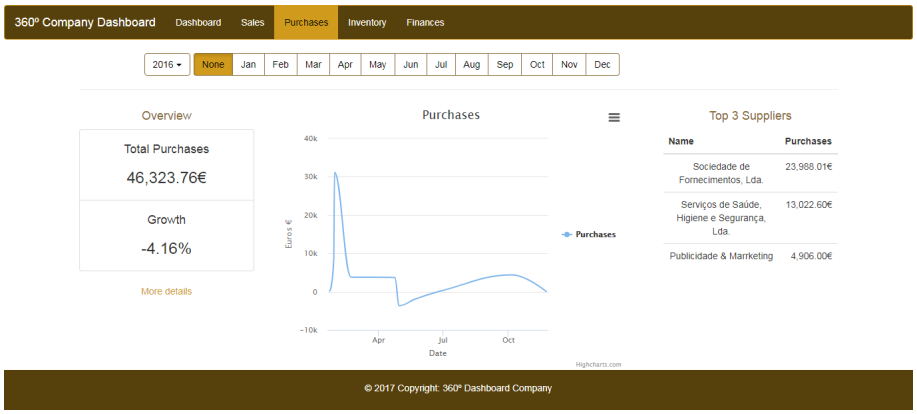

Figure 2: Purchases dashboard

On the left side of this web page, two KPI can be seen illustrating the total amount spent in purchases and the growth in volume of purchases compared to the same period last year. In the middle, a line chart displays the evolution of purchases per day. On the right side, a table showing the top three suppliers with whom the company spends more money and the amount of money sent which of them.

From this page, a drill down with a detailed list of all purchases (V_08) in the time frame can be accessed.

2.2.1 Primavera Webservices used:

- getPurchasesGroupedByDate
- getPurchasesGroupedBySupplier

2.3 Sales Dashboard (V_03)

Similarly to the purchases dashboard, this dashboard is also accessible through the top navigation bar, and provides an overview of the sales department in the time-frame selected.

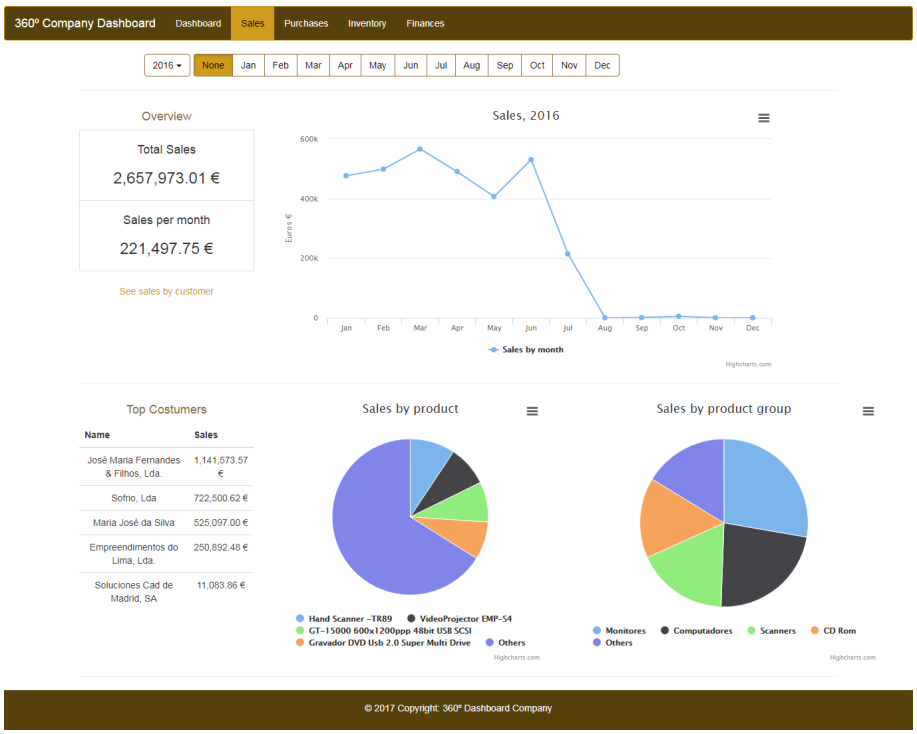

Figure 3: Sales dashboard

In this core view, two KPI can be seen showing the total amount of sales and an average sale per month or day, depending on the period selected. In addition, a line chart illustrates the sales evolution and two pie charts show the distribution of sales per product and per category in order to inform the user of which product are best selling. Also, a table with the top customers can be seen so the user knows who are the customers who spent money with the business.

Finally, from this dashboard the user can access a list of sales per customer (V_09) which provides more detailed information about sales. In alternative, all the other dashboards are accessible through the top navigation bar.

2.3.1 Webservices of our server used:

- /getSalesByYear
- /getSalesByMonth
- /getCustomers
- /getSalesByProductGroup
- /getSalesByProduct

2.4 Inventory Dashboard (V_04)

This dashboard can be reached at all times through the top navigation bar and provides an overview of the state of the inventory in the time-frame selected.

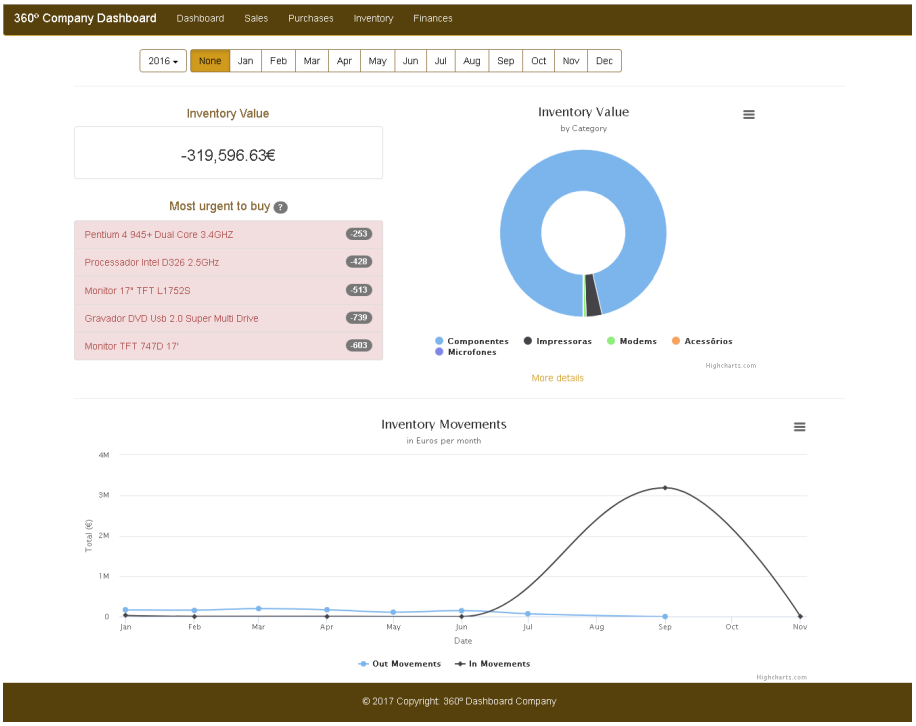

Figure 4: Inventory dashboard

In the the top-left corner of this page, the inventory value in the beginning of the time-frame selected. Below that, a list of products whose stock is below the minimum specified in Primavera ERP is shown so that the user knows that this are the products which need to be bought more urgently. In the top-right corner, a pie chart shows the distribution of inventory value by category/family. In the bottom the page, a line chart shows the evolution of the movements of stock.

From this dashboard, a list of information about the stock of all the products sold by the company can be reached. Also by clicking, in the rows of the table, the user is redirected to the page of product selected for more information on that specific product.

2.4.1 Primavera Webservices used:

- getInventory
- getInventoryOutOfStock
- getInventoryOutMovements
- getInventoryInMovements

2.5 Finances Dashboard (V_05)

The finances dashboard, always reachable through the top navigation bar, provides as the names suggests an overview of the financial state of the company in the time-frame selected.

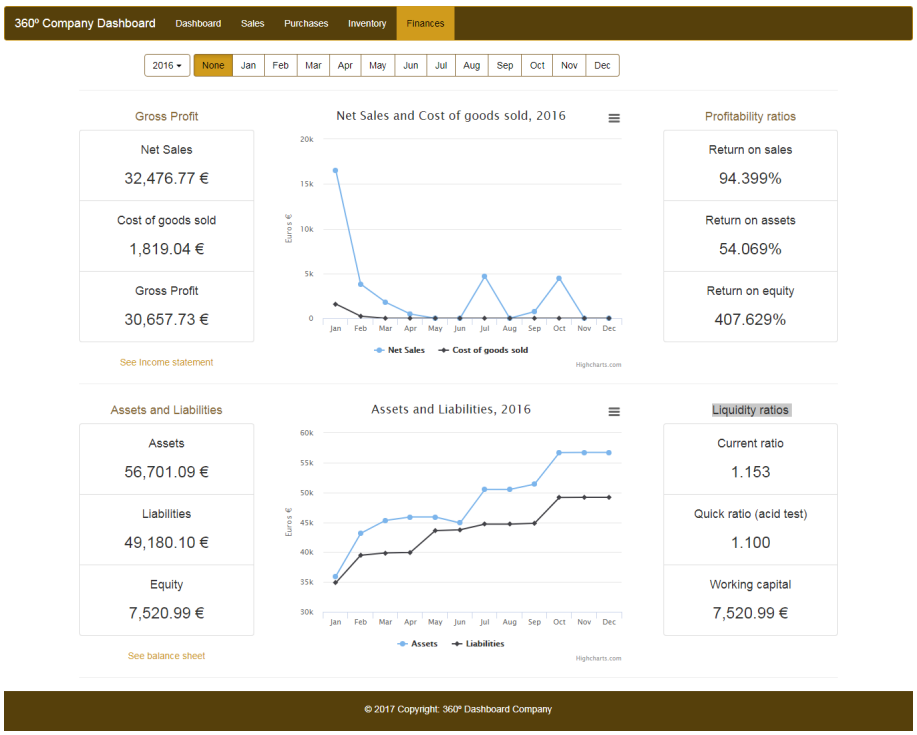

Figure 5: Finances dashboard

In this page, in the left side two tables can be seen showing some values relevant to the financial health of the business, such as net sales, cost of goods sold, assets and liabilities. In the middle section, two line charts show the evolution of net sales, cost of goods sold, assets and liabilities, which compare the first two and the later two. Ideally, the net sales should always be greater than the cost of goods sold. Also, the liabilities being less than the assets is a good indicator for the company. Finally, information about profitability and liquidity ratios can be seen in key performance indicators in the right side of the page.

A Income Statements (V_06) and Balance Sheet (V_07) pages can be accessed from this dashboard for more information.

2.5.1 Webservices of our server used:

- /getBalancos
- /getDemonstracaoResultados

3 Other Features

3.1 Income statement (V_06)

This page, accessible from the Finances Dashboard (V_05), shows a complete income statement of the company in a given point in time. For this purpose, we use a web service from our server called "/getDemonstracaoResultados"

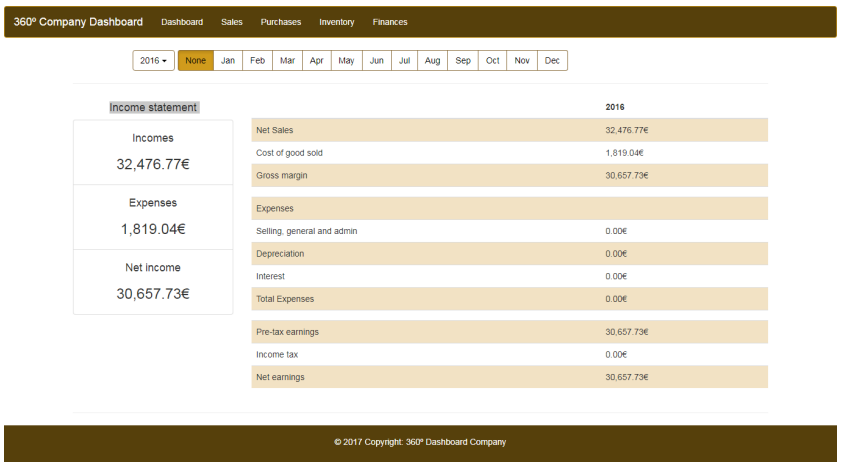

Figure 6: Income statement

3.2 Balance sheet (V_07)

This page, accessible from the Finances Dashboard (V_05), provides a balance sheet of the company in a given point in time. For this purpose, we use a web service from our server called "/getBalanco"

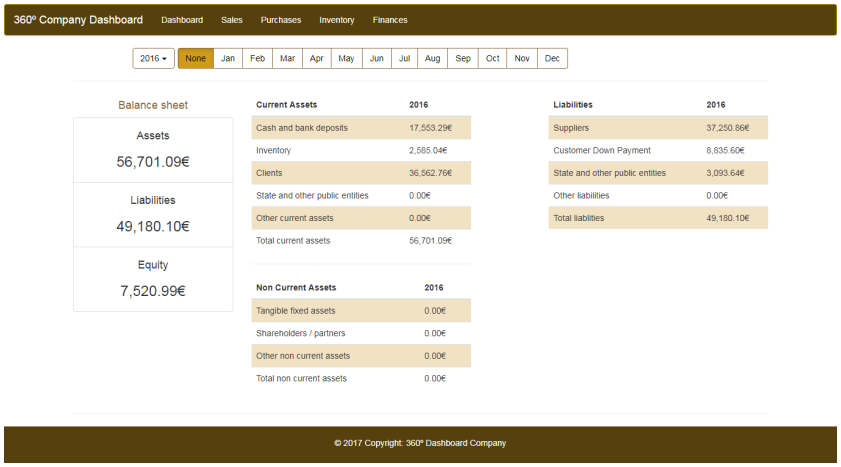

Figure 7: Balance sheet

3.3 Detailed Purchases (V_08)

This page, accessible from the Purchases Dashboard (V_02), shows a list of documents representing the business' purchases. For this purpose, we use a web service from Primavera called "getPurchases".

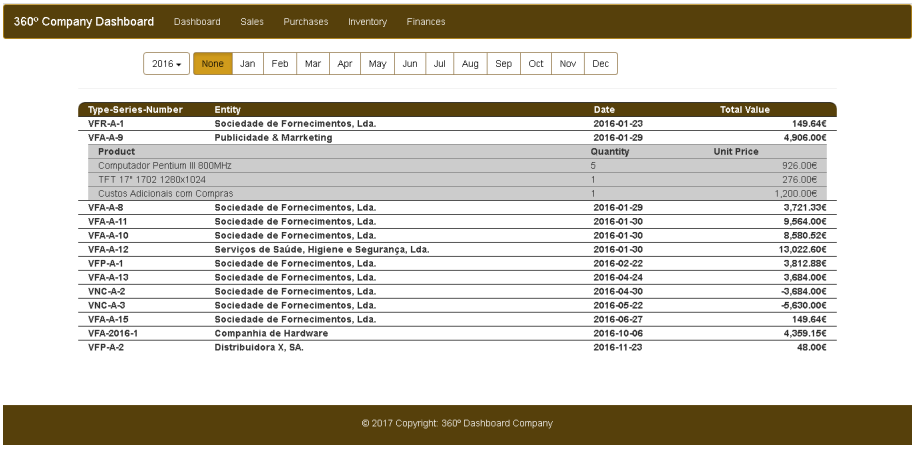

Figure 8: Detailed Purchases page

Furthermore, the user is redirect to product's page (V_11) for more information by clicking in any it's name.

3.4 Sales by Customer (V_09)

This page, accessible from the Sales Dashboard (V_03), shows a list of the sales made by each customer. For this purpose, we use one of two web services from our server called "/getSalesByYear" and "/getSalesByMonth", depending on the time frame selected.

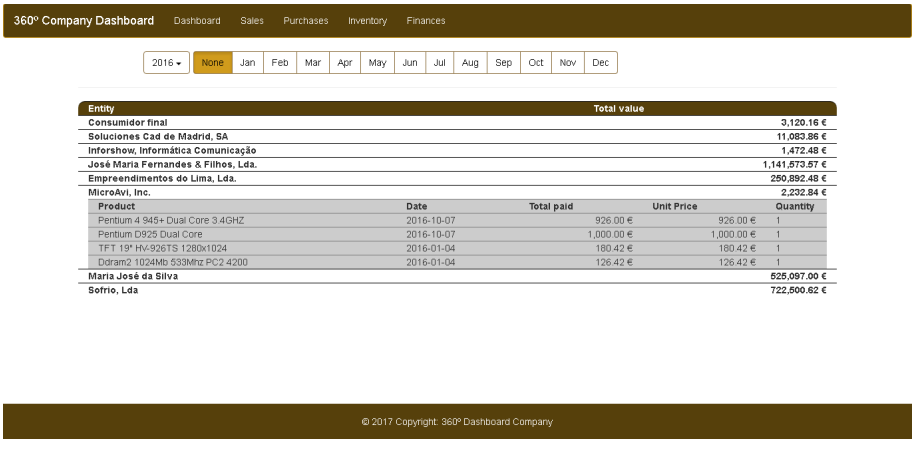

Figure 9: Sales by Customer page

Furthermore, the user is redirect to product's page (V_11) for more information by clicking in any it's name.

3.5 Detailed Inventory (V_10)

This page, accessible from the Inventory Dashboard (V_04), shows a list of product and it respective stock. For this purpose, we use a web service from Primavera called "getInventory".

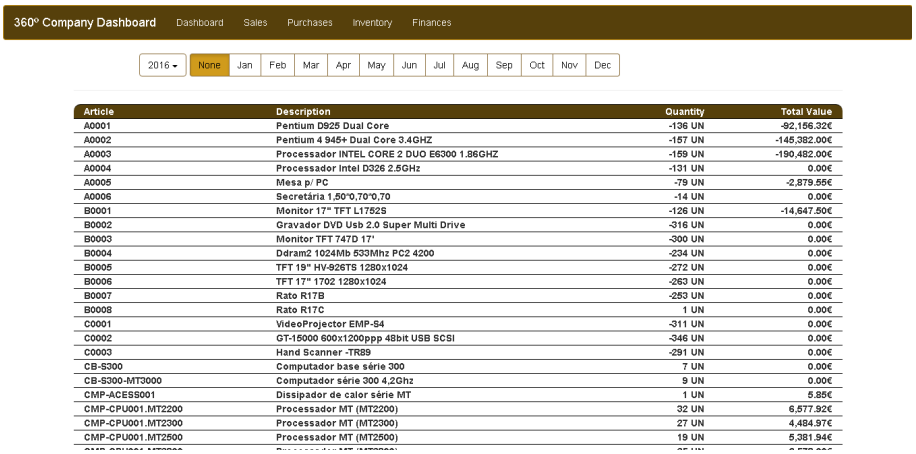

Figure 10: Detailed Inventory page

Furthermore, the user is redirect to product's page (V_11) for more information by clicking in a table row.

3.6 Product Page (V_11)

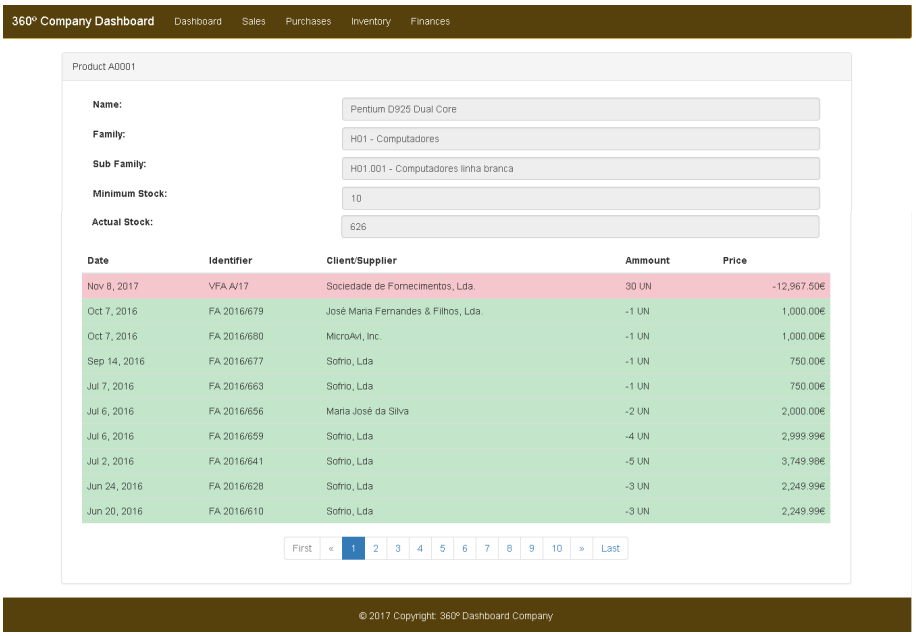

Figure 11: Product Page

This page, accessible from the Detailed Purchases (V_08), Sales by Customer (V_09) and Detailed Inventory (V_10) pages, shows information about a specific product such as, it's name, family, sub-family, current stock an minimum stock acceptable. Also, a list of transactions (sales and purchases) involving the the selected product. For this purpose, we use two web service from Primavera called "getProductInfo" and "getPurchasesOfProduct". In addtion, a web service from our server called "getSalesOfProduct" is also used to get information about the sales of this product from the SAF-T file.

4 Interoperability with Primavera

Webservice ID	getPurchases
Webservice Description	Fetches a list with all purchases in specified year and month
Related Core View(s)	Purchases dashboard
Route+Verbs	host/api/Purchases?year=Y&month=M
Method	GET
Input Example	host/api/Purchases?year=2016&month=5

Expected Output	{[
	{
	"Id": "{96F11A5A–E8B3–11DD–A8B6–000C2989A062}",
	"DocumentDate": "2016–05–22T00:00:00",
	"PaymentDate": "2016–06–21T00:00:00",
	"Entity": "F0001",
	"EntityName": "Sociedade de Fornecimentos, Lda.",
	"DocumentNumber": 14,
	"Notes": "",
	"TotalValue": 3640,
	"DocumentType": "VFA",
	"DocumentSeries": "A",
	"Items": [
	{
	"Id": "{7DD7B8E9–E8B3–11DD–A8B6–000C2989A062}",
	"Product": "A0001",
	"Description": "Computador Celeron 800MHz",
	"Quantity": –8,
	"UnitPrice": 455,
	"Value": 3640
	}
	],
	}
	...]}

Webservice ID	getPurchasesGroupedByDate
Webservice Description	Fetches a list with the total spent in purchases per day in specified year and month
Related Core View(s)	Purchases dashboard
Route+Verbs	host/api/Purchases/groupByDate?year=Y&month=M
Method	GET
Input Example	host/api/Purchases/groupByDate?year=2015&month=06
Expected Output	{[{ "DocumentDate": "2015-06-01T00:00:00", "TotalValue": 981, }, ...]}

Webservice ID	getPurchasesGroupedBySupplier
Webservice Description	Fetches a list with the total spent in purchases per supplier in specified year and month
Related Core View(s)	Purchases dashboard
Route+Verbs	host/api/Purchases/groupBySupplier?year=Y&month=M
Method	GET
Input Example	host/api/Purchases/groupBySupplier?year=2015&month=06
Expected Output	{[{ "Entity": "F0003", "EntityName": "Companhia de Hardware", "TotalValue": 10203.8 }, ...]}

Webservice ID	getPurchasesFromSupplier
Webservice Description	Fetches a list with all purchases from a supplier in time interval specified
Related Core View(s)	Purchases dashboard
Route+Verbs	host/api/Purchases?initial=X&final=Y&supplier=Z
Method	GET
Input Example	host/api/Purchases?initial=2015-01-01&final=2016-12-30&supplier=F0001

Expected Output	{
	"purchases" : [
	{
	"Id": "{B8D74796-8F02-431C-A174-CF737D863497}",
	"DocumentDate": "2015-06-06T00:00:00",
	"PaymentDate": "2015-07-06T00:00:00",
	"Entity": "F0003",
	"EntityName": "Companhia de Hardware",
	"DocumentNumber": 5,
	"Notes": "",
	"TotalValue": -10203.8,
	"DocumentType": "VFA",
	"DocumentSeries": "A",
	"Items": [
	{
	"Id": "{6BF51D9E-05A4-47F1-AD87-ECE0E70F2D98}",
	"Product": "A0003",
	"Description": "Processador INTEL CORE 2 DUO E6300
	"Quantity": -3,
	"UnitPrice": -1198,
	"Value": -2875.2
	},
	...]
	}
	...]}

Webservice ID	getInventory
Webservice Description	Fetches a list with all products and theirs respective quantity at a given date(first day of the year/month)
Related Core View(s)	Inventory dashboard
Route+Verbs	host/api/Inventory/date/year/month
Method	GET
Input Example	host/api/Inventory/date/2016/2
Expected Output	{ [{ "Article": "A0001", "Family": "Computadores", "SubFamily": "H01.001", "Description": "Pentium D925 Dual Core", "CurrentStock": -136, "ReserveStock": 0, "UnitPrice": 677.62, "TotalValue": -92156.32 }, ...]}

Webservice ID	getInventoryOutOfStock
Webservice Description	Fetches a list with all products whose stock is below the minimum defined in Primavera at the moment
Related Core View(s)	Inventory dashboard
Route+Verbs	host/api/Inventory/outOfStock
Method	GET
Input Example	host/api/Inventory/outOfStock
Expected Output	{ [{ "Article": "A0002", "Family": "H01", "SubFamily": "H01.002", "Description": "Pentium 4 945+ Dual Core 3.4GHZ", "CurrentStock": -253, "MinStock": 20, "ReposStock": 25, }, ...]}

Webservice ID	getInventoryOutMovements
Webservice Description	Fetches a list with information about the date, monetary value and quantity of all movements of stock out of the company in a given time frame
Related Core View(s)	Inventory dashboard
Route+Verbs	host/api/Inventory/outMovements?year=year&month=month
Method	GET
Input Example	host/api/Inventory/outMovements?year=2016&month=1
Expected Output	{ [{ "data": "2016-01-01T00:00:00", "valor": 17936, "quantidade": 46 }, ...]}

Webservice ID	getInventoryInMovements
Webservice Description	Fetches a list with information about the date, monetary value and quantity of all movements of stock into of the company in a given time frame
Related Core View(s)	Inventory dashboard
Route+Verbs	host/api/Inventory/inMovements?year=year&month=month
Method	GET
Input Example	host/api/Inventory/inMovements?year=2016&month=1
Expected Output	{ [{ "data": "2016-01-30T00:00:00", "valor": 16363.96, "quantidade": 41 }, ...]}

Webservice ID	getProductInfo
Webservice Description	Fetches some information about a specific product
Related View(s)	Product Page
Route+Verbs	host/api/Inventory/id
Method	GET
Input Example	host/api/Inventory/A0001

Expected Output	{
	"Article": "A0001",
	"Family": "H01 - Computadores",
	"SubFamily": "H01.001 - Computadores linha branca",
	"Description": "Pentium D925 Dual Core",
	"CurrentStock": 626,
	"MinStock": 10,
	"ReposStock": 14,
	}

Webservice ID	getPurchasesOfProduct
Webservice Description	Fetches some information about a specific product
Related View(s)	Product Page
Route+Verbs	host/api/Purchases/id
Method	GET
Input Example	host/api/Purchases/A0001

Expected Output	{
	[{
	"Id": "VFA A/17",
	"DocumentDate": "2017-11-08T00:00:00",
	"PaymentDate": "2017-12-08T00:00:00",
	"Entity": "F0001",
	"EntityName": "Sociedade de Fornecimentos, Lda.",
	"DocumentNumber": 17,
	"Notes": "",
	"TotalValue": -56094,
	"DocumentType": "VFA",
	"DocumentSeries": "A",
	"Items": [
	{
	"Id": "{9037FCB8-D3A1-11E7-89D5-080027BB1857}",
	"Product": "A0001",
	"Description": "Pentium D925 Dual Core",
	"Quantity": 30,
	"UnitPrice": -455,
	"Value": -12967.5
	}
	],
	},
	...]
	}

4.1 Interoperability with own server and data obtained through the SAF-T

Webservice ID	getSalesByMonth
Webservice Description	Fetches a list with all sales invoices in specified month of an year
Related Core View(s)	Sales dashboard, Overview dashboard
Route+Verbs	host/getSalesByMonth?month=X&year=Y
Method	GET
Input Example	host/getSalesByMonth/?month=1&year=2016
Expected Output	[{ "InvoiceNo": "FA 2016/1", "InvoiceStatus": "N", "InvoiceDate": "2016-01-01T00:00:00.000Z", "InvoiceType": "FT", "CustomerID": "PT123456789_C", "TaxPayable": 54.51, "NetTotal": 272.54, "GrossTotal": 327.05, "Lines": [{ "lineNumber": "1", "productCode": "B0007", "productDescprition": "Rato R17B", "quantity": "5", "unitOfMeasure": "UN", "unitPrice": 0, "creditAmount": 0, "debitAmount": null, "taxType": "IVA", "taxPercentage": 20, }, ...] }, ...]

Webservice ID	getSalesByYear
Webservice Description	Fetches a list with all sales invoices in specified year
Related Core View(s)	Sales dashboard, Overview dashboard
Route+Verbs	host/getSalesByMonth?year=Y
Method	GET
Input Example	host/getSalesByMonth/?year=2016
Expected Output	[{ "InvoiceNo": "FA 2016/1", "InvoiceStatus": "N", "InvoiceDate": "2016-01-01T00:00:00.000Z", "InvoiceType": "FT", "CustomerID": "PT123456789_C", "TaxPayable": 54.51, "NetTotal": 272.54, "GrossTotal": 327.05, "Lines": [{ "lineNumber": "1", "productCode": "B0007", "productDescprition": "Rato R17B", "quantity": "5", "unitOfMeasure": "UN", "unitPrice": 0, "creditAmount": 0, "debitAmount": null, "taxType": "IVA", "taxPercentage": 20, }, ...] }, ...]

Webservice ID	getCustomers
Webservice Description	Fetches a list with all customers in specified year or month
Related Core View(s)	Sales dashboard, Overview dashboard
Route+Verbs	host/getCustomers?year=Y[&month=M]
Method	GET
Input Example	host/getCustomers?year=2016&month=1
Expected Output	[{ "customer_id": "PT123456789_C", "account_id": "21111001", "customer_tax_id": "123456789", "company_name": "Sofrio, Lda", "telephone": "200267890", "fax": "200267899", "sales": 85411.18 }, ...]

Webservice ID	getSalesByProduct
Webservice Description	Fetches a list with all products sorted by decreasing order in the number of sales
Related Core View(s)	Sales dashboard
Route+Verbs	host/getSalesByProduct?year=Y[&month=M]
Method	GET
Input Example	host/getSalesByProduct?year=2016&month=1
Expected Output	[{ "_id": "5a2ff3527c1738104805b8c9", "ProductType": "P", "ProductCode": "C0003", "ProductGroup": "Scanners", "ProductDescription": "Hand Scanner - TR89", "ProductNumberCode": "C0003", "__v": 0, "itemsSelled": 91 }, ...]

Webservice ID	getSalesByProductGroup
Webservice Description	Fetches a list with all product groups sorted by decreasing order in the number of sales
Related Core View(s)	Sales dashboard, Overview dashboard
Route+Verbs	host/getSalesByProductGroup?year=Y[&month=M]
Method	GET
Input Example	host/getSalesByProductGroup?year=2016&month=1
Expected Output	[{ "name": "Monitores", "itemsSelled": 218 }, { "name": "Computadores", "itemsSelled": 183 }, ...]

Webservice ID	getBalanco
Webservice Description	Get balance sheet
Related Core View(s)	Finances dashboard, Balance sheet
Route+Verbs	host/getBalanco?year=Y[&month=M]
Method	GET
Input Example	host/getBalanco?year=2016&month=1

Expected Output	{
	"assets": {
	"value": 35965.12,
	"curr": {
	"cashAndBankDeposits": 18429.87,
	"inventory": 0,
	"clients": 17535.25,
	"stateAndOtherPubEntAssets": 0,
	"otherCurrAssets": 0,
	"totalCurrAssets": 35965.12
	},
	"nonCurr": {
	"shareholders_partners": 0,
	"tangibleFixedAssets": 0,
	"otherNonCurrAssets": 0,
	"totalNonCurrAssets": 0
	}
	},
	"liabilities": {
	"value": 34869.47,
	"suppliers": 23830.78,
	"stateAndOtherPubEntAssetsLiabilities": 2203.09,
	"otherLiabilities": 0,
	"customerDownPayment": 8835.6,
	"totalCurrLiabilities": 34869.47
	},
	"equity": {
	"value": 1095.65
	}
	}

Webservice ID	getBalancos
Webservice Description	Get a list of balance sheets over the specified period
Related Core View(s)	Finances dashboard
Route+Verbs	host/getBalancos?year=Y[&month=M]
Method	GET
Input Example	host/getBalancos?year=2016&month=1

Expected Output	[
	{
	"assets": {
	"value": 35965.119999999995,
	"curr": {
	"cashAndBankDeposits": 18429.87,
	"inventory": 0,
	"clients": 17535.25,
	"stateAndOtherPubEntAssets": 0,
	"otherCurrAssets": 0,
	"totalCurrAssets": 35965.12
	},
	"nonCurr": {
	"shareholders_partners": 0,
	"tangibleFixedAssets": 0,
	"otherNonCurrAssets": 0,
	"totalNonCurrAssets": 0
	}
	},
	"liabilities": {
	"value": 34869.47,
	"suppliers": 23830.78,
	"stateAndOtherPubEntAssetsLiabilities": 2203.09,
	"otherLiabilities": 0,
	"customerDownPayment": 8835.6,
	"totalCurrLiabilities": 34869.47
	},
	"equity": {
	"value": 1095.6499999999942
	}
	},
	...
	]

Webservice ID	getDemonstracaoResultados
Webservice Description	Get income statement
Related Core View(s)	Finances dashboard, Income Statement
Route+Verbs	host/getDemonstracaoResultados?year=Y[&month=M]
Method	GET
Input Example	host/getDemonstracaoResultados?year=2016&month=1

Expected Output	{
	"netIncome": 14926.79,
	"netExpense": 1588.64,
	"netResult": 14926.79,
	"netSales": 16515.43,
	"costOfGoodSold": 1588.64,
	"grossMargin": 14926.79,
	"sellingGeneralAdmin": 0,
	"depreciation": 0,
	"interest": 0,
	"totalExpenses": 0,
	"preTaxEarnings": 14926.79,
	"incomeTax": 0,
	"netEarnings": 14926.79,
	"incomes": 16515.43,
	"expenses": 1588.64
	}

Webservice ID	getSalesOfProduct
Webservice Description	Get a list with all the invoices of sales of a certain product
Related View(s)	ProductPage
Route+Verbs	host/getSalesOfProduct?code=X
Method	GET
Input Example	host/getSalesOfProduct?code=A0001

Expected Output	{
	[{
	"Id": "FA A/18",
	"DocumentDate": "2016-01-30T00:00:00.000Z",
	"Entity": "PT123456789_C",
	"EntityName": "Sofrio, Lda",
	"TotalValue": 2876.59,
	"Items": [
	{
	"Product": "A0001",
	"Quantity": -3,
	"UnitPrice": 878,
	"Value": 2634
	}
	]
	},
	...]
	}

5 System Architecture

In figure 12, we present the system architecture, a client-server model in which the front-end is made using AngularJS and comunicates with two servers.

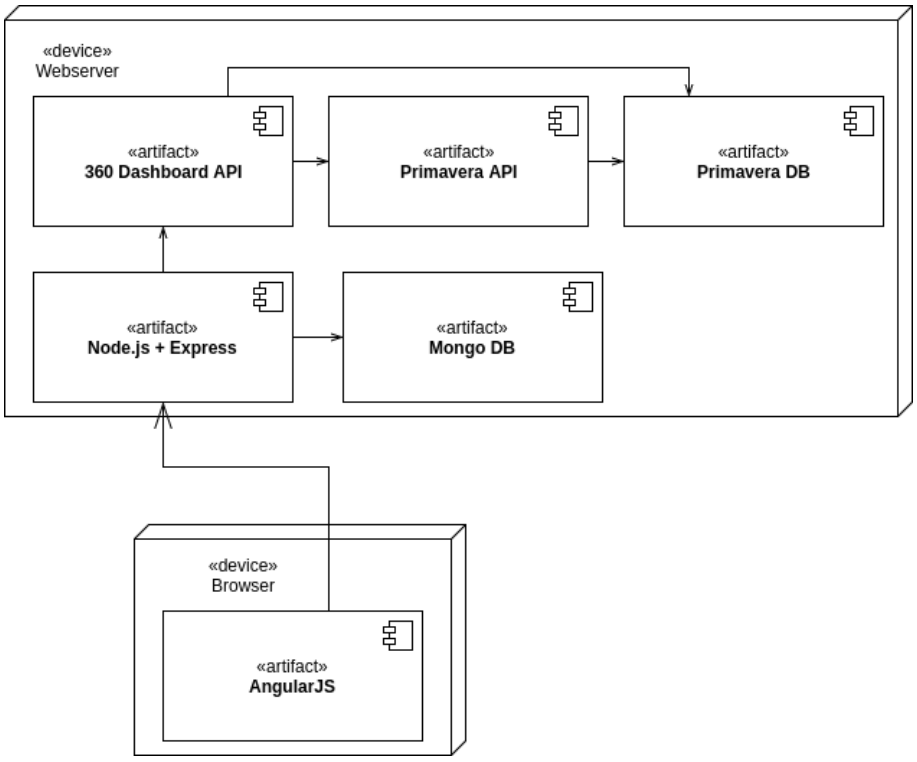

Figure 12: Deployment diagram

One server uses C# AND ASP.NET and communicates with Primavera ERP and in term to it's own database. As such it provides the endpoints for the Primavera interoperability web services. Consequently, it tries to best of the group's abilities to use the Primavera API. However, some SQL queries were also required to obtain the information need to implement the features agreed upon.

The other server made in Node.js using Express framework parses SAF-T file and ensures persistence the information in a MongoDB database. In addition, this server provides the endpoints for the pages of the site and the webservices which provide information from found in the SAF-T file.

6 Paths

In a normal flow, the user enter the site by the Overview Dashboard page. From that page, it can access any of other dashboards and respective drill down functionalities pages as shown in figure 13.

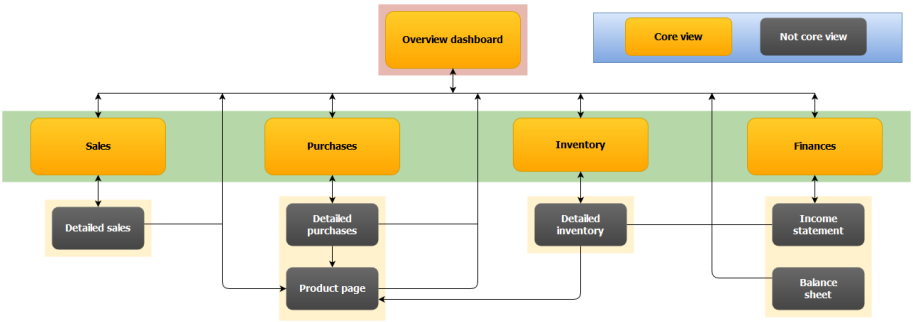

Figure 13: Site map

7 Project Specifications vs Delivered Project

All the initial specifications were implemented. The only exception being the functionality of viewing stock evolution due to the query required being inducing more latency than the value of the information it provided was deemed worth.

However, during the semester, new ideas emerged and some were implemented. The charts of top product sales, a specific page for a product, and finance ratios in the core of finance were added. Also, the drill downs of the income statement and balance sheet were added to the Finances Dashboard.

8 Lessons learned

This project introduced us to the operation of an ERP, in this case the Primavera. Also gave us to know details of the operation of a company in the level of management and finances. At the technological level it was also a very instructive project. We had to program in a language that we've never used before, C#. Also we were able to better understand how the information is stored in Primavera ERP and the complexities of having such a complex information stored in a coherent way.

In addition, we implemented our own server using node.js. For the part of the front end we used AngularJS, something that we had never used before either.