



FINAL REPORT

INFORMATION SYSTEMS
2018/2019

INTEGRATED MASTER IN INFORMATICS AND COMPUTER
ENGINEERING

360° Company Dashboard

Group M

André Cruz	up201503776@fe.up.pt
Bárbara Sofia Silva	up201505628@fe.up.pt
Edgar Carneiro	up201503784@fe.up.pt
Xavier Fontes	up201503145@fe.up.pt

23rd December 2018

Contents

1 Project Overview	2
2 Core Views	3
2.1 Overview	3
2.2 Sales	4
2.3 Financial	5
2.4 Procurement	6
2.5 Inventory	7
2.6 Product Drill Down	8
2.7 Client Drill Down	8
2.8 Supplier Drill Down	9
3 Functionalities	10
3.1 Overview	10
3.2 Sales	11
3.3 Financial	12
3.4 Procurement	13
3.5 Inventory	14
3.6 Product Drill Down	15
3.7 Client Drill Down	16
3.8 Supplier Drill Down	16
3.9 Other Functionalities	17
4 Interoperability with SAF-T	18
5 Interoperability with Primavera	21
6 Paths	24
7 System Architecture	26
7.1 Services	26
7.2 Pages	28
7.3 Components	28
8 Project Specifications vs Delivered Project	29
9 Lessons Learned	30

1 Project Overview

This project consists of the development of a web application to provide data and key performance indicators (KPIs) relevant to the business.

The use of comprehensive visualizations aims to simplify the analysis of the large stream of data inherent to the business, thus easing the decision making process.

The dashboard presents information regarding the financial performance of the company, its sales, its purchases from suppliers, its inventory, and its products' specifications. The app relies on and inter-operates with the *Primavera ERP*, but all authentication data and KPIs calculations are handled by our own backend service.

This WebApp is for the company's internal use and thus should not have pages indexed in search engines, safe for the login page.

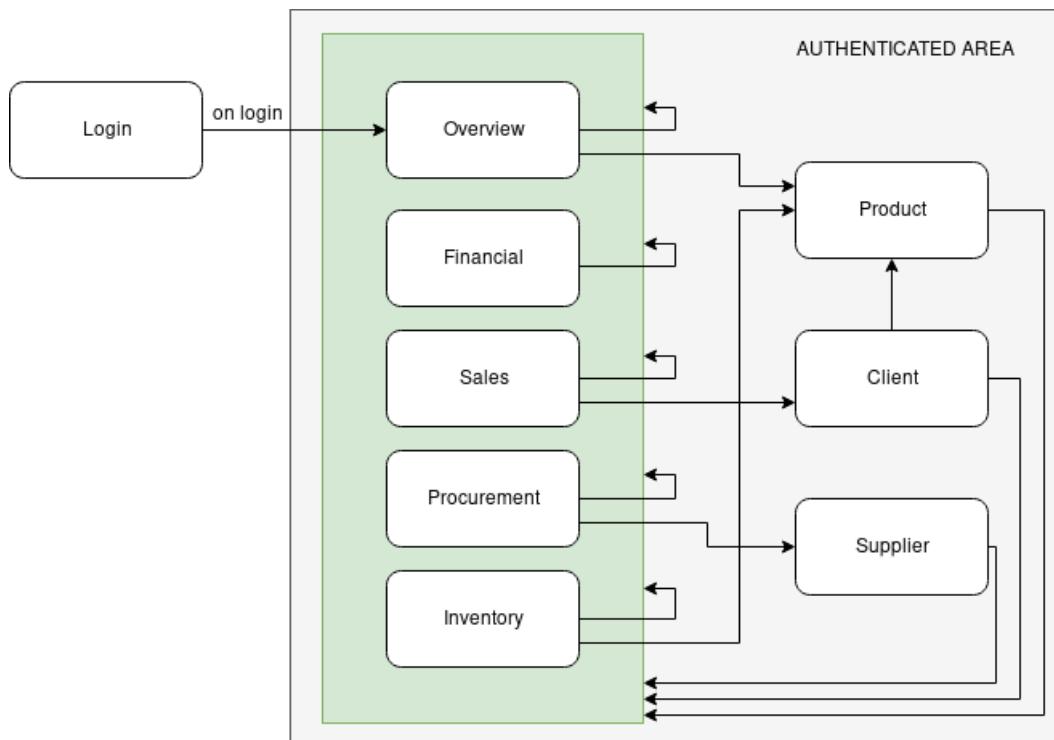


Figure 1: Site map of the project, explained in detail in Section 6

2 Core Views

2.1 Overview

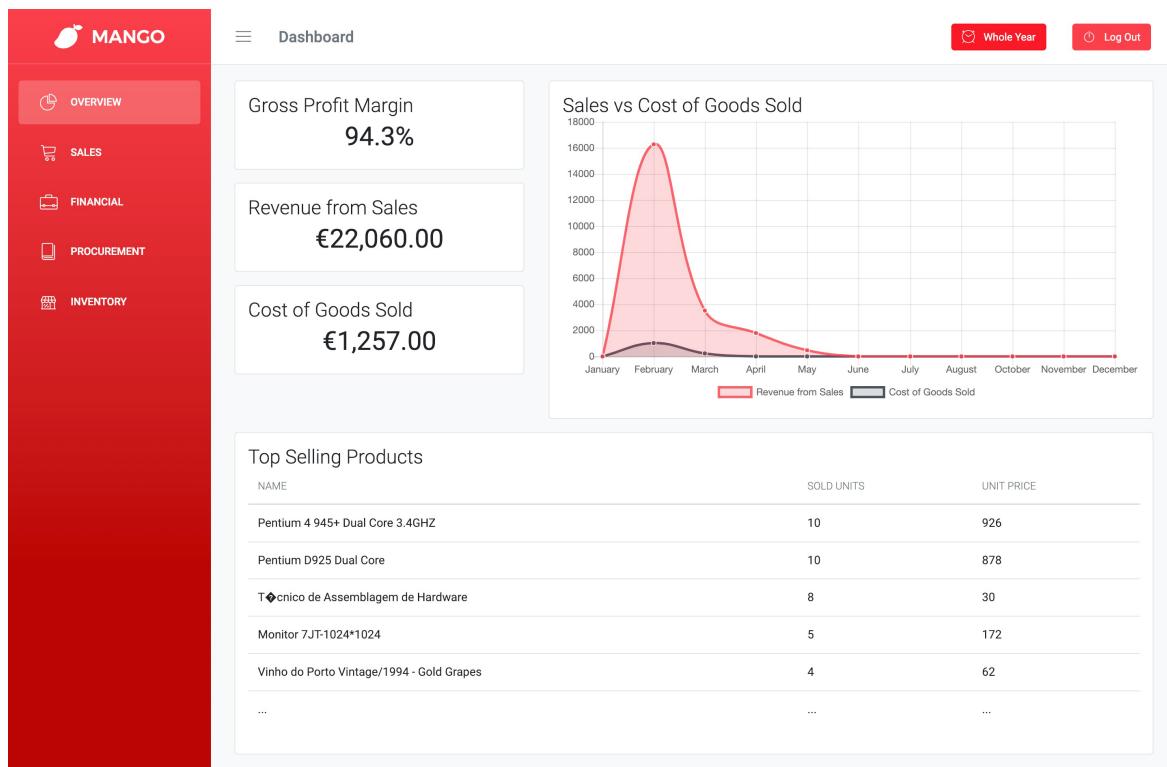


Figure 2: Overview core view

2.2 Sales

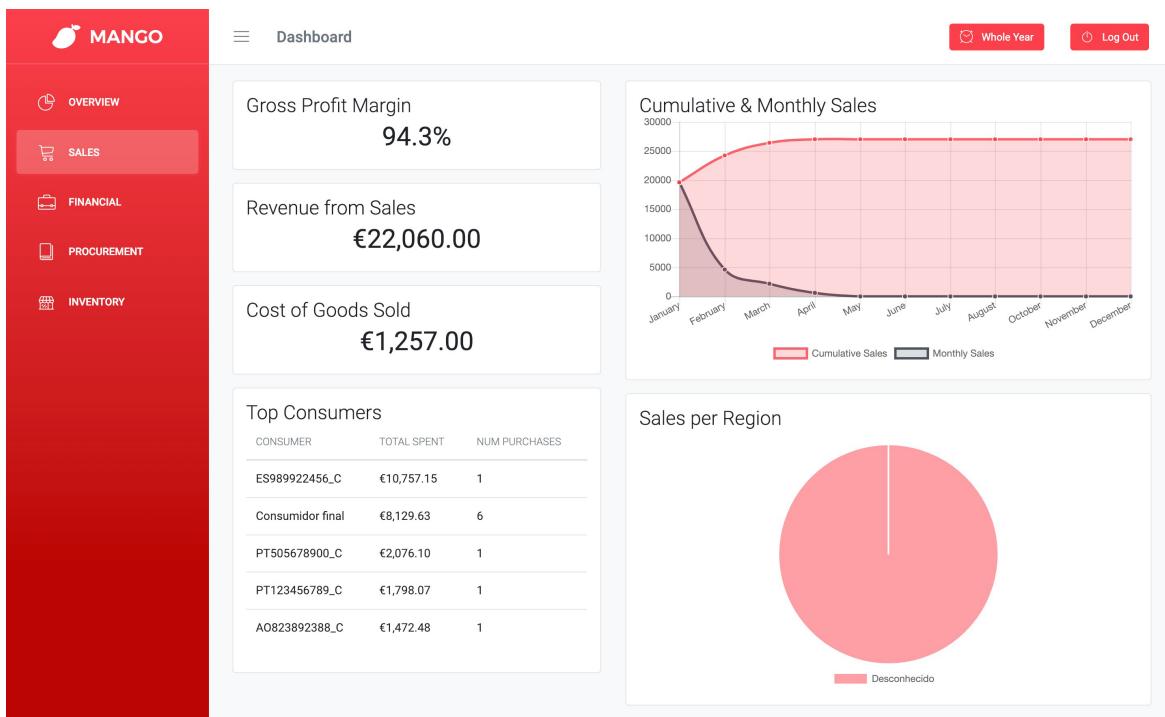


Figure 3: Sales core view

2.3 Financial

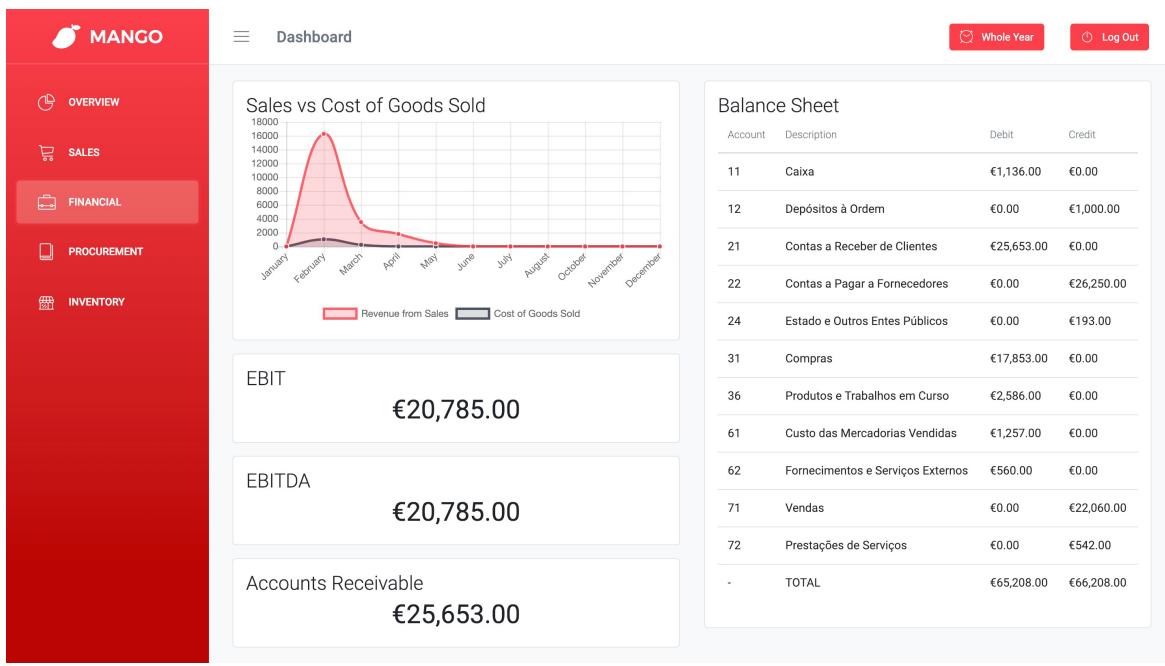


Figure 4: Financial core view

2.4 Procurement

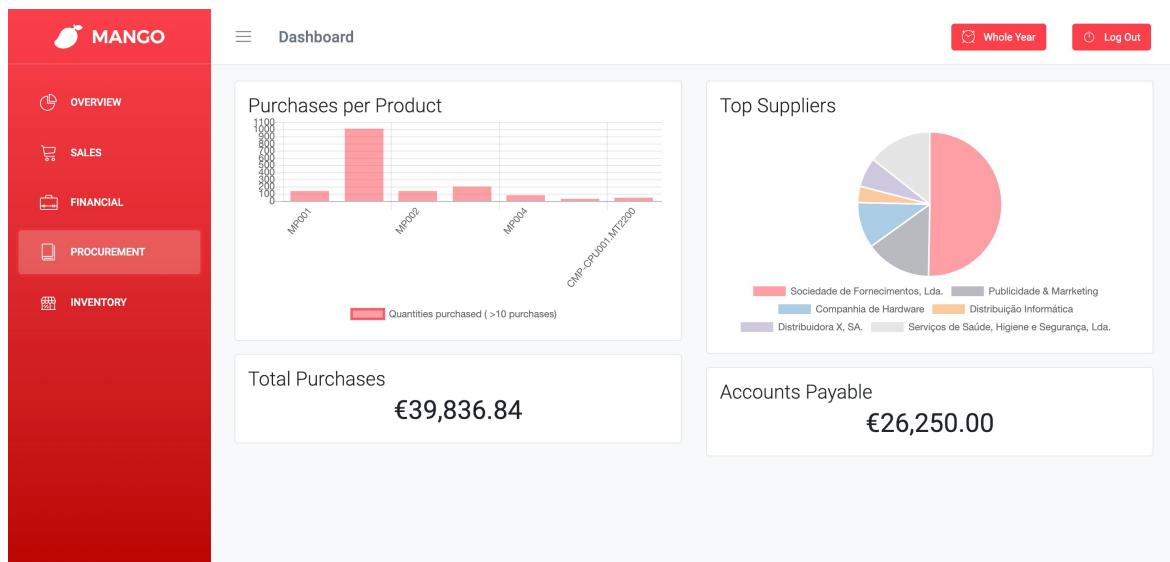
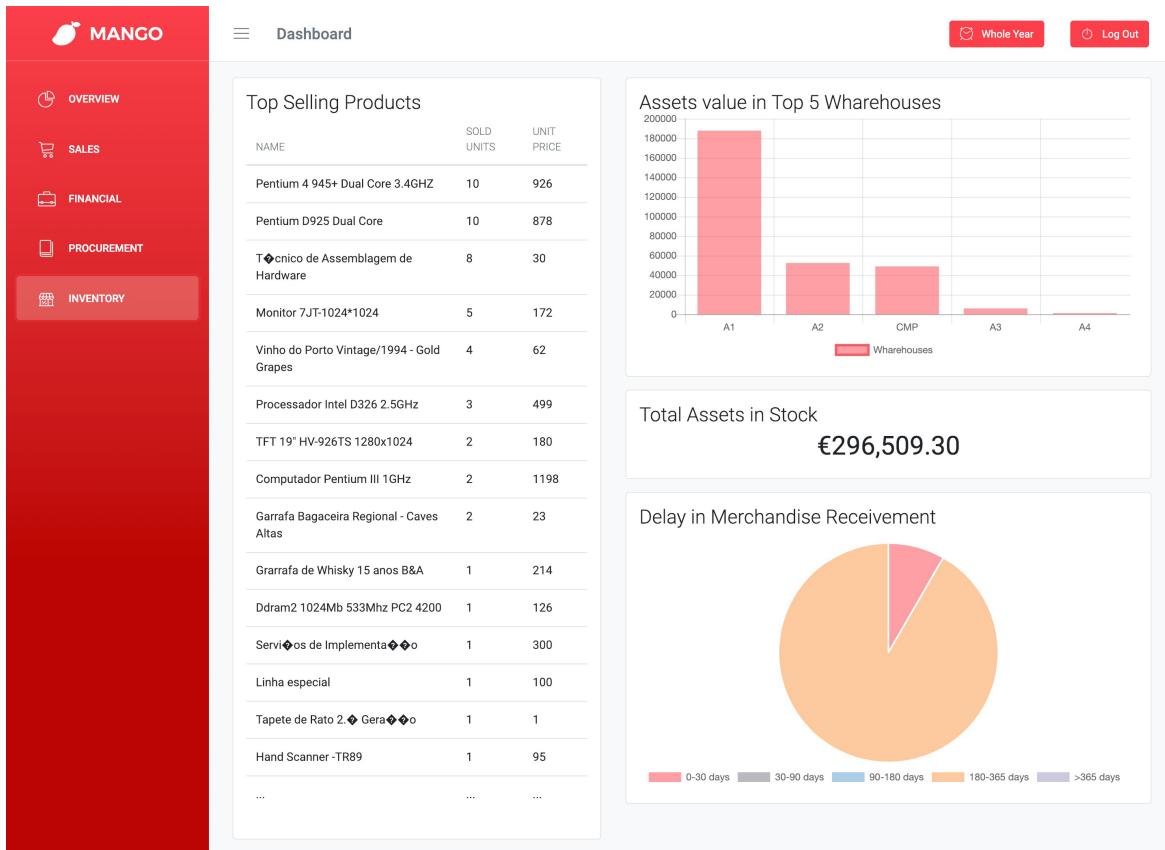


Figure 5: Procurement core view

2.5 Inventory



2.6 Product Drill Down

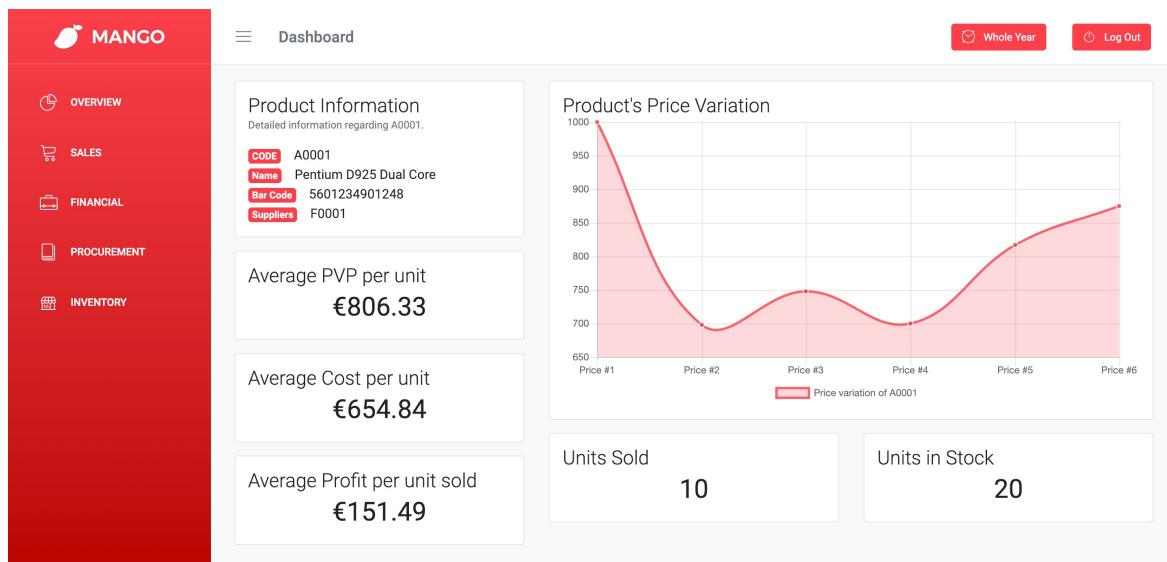


Figure 6: Product drill down core view

2.7 Client Drill Down

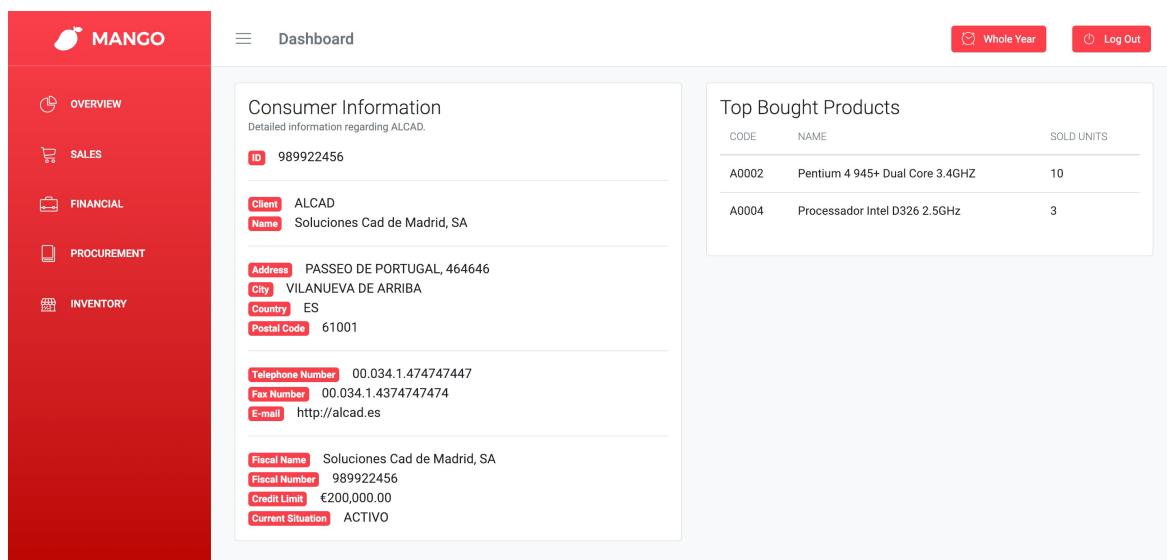


Figure 7: Client drill down core view

2.8 Supplier Drill Down

The screenshot shows the MANGO 360° Company Dashboard interface. On the left is a red sidebar with navigation links: Overview (selected), Sales, Financial, Procurement, and Inventory. The main content area has a white header bar with 'Dashboard' and two buttons: 'Whole Year' and 'Log Out'. Below this is a section titled 'Supplier Information' with a sub-section 'Supplier Details'. It lists the supplier's name as 'F0001' and 'Sociedade de Fornecimentos, Lda.'. It also provides physical address details: 'Av. do Sul, 387383', 'Lisboa', 'PT', and '1000-001'. Contact information includes telephone number '21383833838', fax number '21383833838', and email 'http://socfornecimento.pt'. Fiscal details include 'Fiscal Name' 'Sociedade de Fornecimentos, Lda.', 'Fiscal Number' '503144703', and 'Payment Method' 'CHQ'. Below this, there are two summary boxes: 'Total Purchased from Supplier' showing a value of '€48,394.46' and 'Pending Purchases Value' showing a value of '€6,547.78'.

Figure 8: Supplier drill down core view

3 Functionalities

3.1 Overview

Functionality	Description	Value	Associated Dashboard
Sales vs Cost of Goods Sold	Graph with the two values against each other.	An overview of financial operation.	CORE_OVER
Revenue From Sales	Value of the total sales volume.	An overview of how well the sales are.	CORE_OVER
Cost of Goods Sold	Cost of the products sold.	An overview of how much we spent on what we sold.	CORE_OVER
Top Products	List of the top selling products.	Understand which products the clients buy the most.	CORE_OVER
Gross Profit Margin	Indicator of how well we are doing in terms of turning profit.	Understand how much of we earn for every euro sold.	CORE_OVER

3.2 Sales

Functionality	Description	Value	Associated Dashboard
Gross Profit Margin	Indicator of how well we are doing in terms of turning profit.	Understand how much of we earn for every euro sold.	CORE_SALES
Revenue From Sales	Value of the total sales volume.	An overview of how well the sales are.	CORE_SALES
Cost of Goods Sold	Cost of the products sold.	An overview of how much we spent on what we sold.	CORE_SALES
Top Consumers	List of the top clients.	Present information regarding the entities that are buying more from us.	CORE_SALES
Sales per Region	List associating sales volume to its region, as a percentage.	Understand where our market is concentrated.	CORE_SALES
Cumulative and Monthly Sales	A cumulative graph about the company's sales volume in the past together with monthly information.	Understand the discrepancy in the sales volume, peaks and stagnation periods between time intervals.	CORE_SALES

3.3 Financial

Functionality	Description	Value	Associated Dashboard
Sales vs Cost of Goods Sold	Graph with the two values against each other.	An overview of financial operation.	CORE_FINANCIAL
EBIT/EBITDA	Value of the earnings before interest and taxes and considering depreciation and amortizations.	A common metric to evaluate the performance of a company's core operations.	CORE_FINANCIAL
Accounts Receivable	Credit we supplied to clients.	Understand how much money we are owed, understand our liquidity situation.	CORE_FINANCIAL
Balance Sheet	A table where all the items in the company's balance sheet are presented.	Presents an overview of the company's financial status.	CORE_FINANCIAL

3.4 Procurement

Functionality	Description	Value	Associated Dashboard
Accounts Payable	Indicator of how much the company owes its suppliers.	Understand how much liabilities the company has to its suppliers.	CORE_PROCUR
Top Suppliers	Pie chart of sales percentage grouped by supplier and filtered by the top suppliers.	Understanding about which supplier is contributing more to our sales margin.	CORE_PROCUR
Total Purchases	Indicator of how much the company bought to suppliers.	Understand how much money was spent in procurement of products.	CORE_PROCUR
Purchases per Product	A bar chart indicating how much a given product was bought.	How much money is spent through orders of inventory.	CORE_PROCUR

3.5 Inventory

Functionality	Description	Value	Associated Dashboard
Top Products	List of the top selling products.	Understand which products the clients buy the most.	CORE_INVENTORY
Total Assets in Stock	Indicator of the total value of the inventory's products.	Understand how much money is sitting in inventory	CORE_INVENTORY
Assets Value in Top Warehouses	Bar chart with the top warehouses based on value of inventory stored there.	Understand how much money is sitting in each warehouse.	CORE_INVENTORY
Delay in Merchandise Receivement	Pie chart indicating how many orders are delayed and by how much.	Comprehend if the company's good are spending too much time without even getting to inventory.	CORE_INVENTORY

3.6 Product Drill Down

Functionality	Description	Value	Associated Dashboard
Product Information	Small product card where the general characteristics can be found.	Easy to understand what kind of product is being reviewed.	DRILL_PROD
Units Sold	The number of units sold in a given time period.	Conclude about life-time of a given product.	DRILL_PROD
Units in Stock	Indicator of how many units of the selected product are still in stock.	It's possible to see if a given product represents a lot of cash just sitting tight.	DRILL_PROD
Average Profit	An indicator of the average profit.	Understand how rentable a product is.	DRILL_PROD
Average PVP and Variation	An average of the product's PVP and its fluctuation as a graph.	Understand how differently the product has been priced.	DRILL_PROD
Average Cost	Indicator of the average unit cost of a given product.	Understanding how much we are spending for each unit we buy.	DRILL_PROD

3.7 Client Drill Down

Functionality	Description	Value	Associated Dashboard
Consumer Information	A detailed list of several characteristics of the consumer.	Understand what type of buyer we are looking at.	DRILL_CLIENT
Top Bought Products	A list of the entity's most bought products.	Understand what type of products this consumer buys.	DRILL_CLIENT

3.8 Supplier Drill Down

Functionality	Description	Value	Associated Dashboard
Supplier Information	A detailed list of several characteristics of the supplier.	Understand what type of supplier we are looking at.	DRILL_SUPPLIER
Total Purchase from Supplier	An indicator of the money spent on orders to this supplier.	Understand how much money we have already spent with this supplier.	DRILL_SUPPLIER
Pending Purchase from Supplier	An indicator of the money spent on orders that haven't arrived yet.	Understand how much money in inventory we have coming from this supplier.	DRILL_SUPPLIER

3.9 Other Functionalities

Functionality	Description	Value
Login/Logout	The user is able to login and logout.	Provide access to the dashboard.
Drilldown Top Products	Go to the product page when a product is clicked in a Top Products page.	Allows the user to investigate more about a product of interest.
Drilldown Consumers	Go to the client page when a client is clicked.	Allows the user to investigate more about a potential marketing target.
Drilldown Suppliers	Go to the supplier page when a supplier is clicked.	Allows the user to investigate more about a current partnership with a supplier.

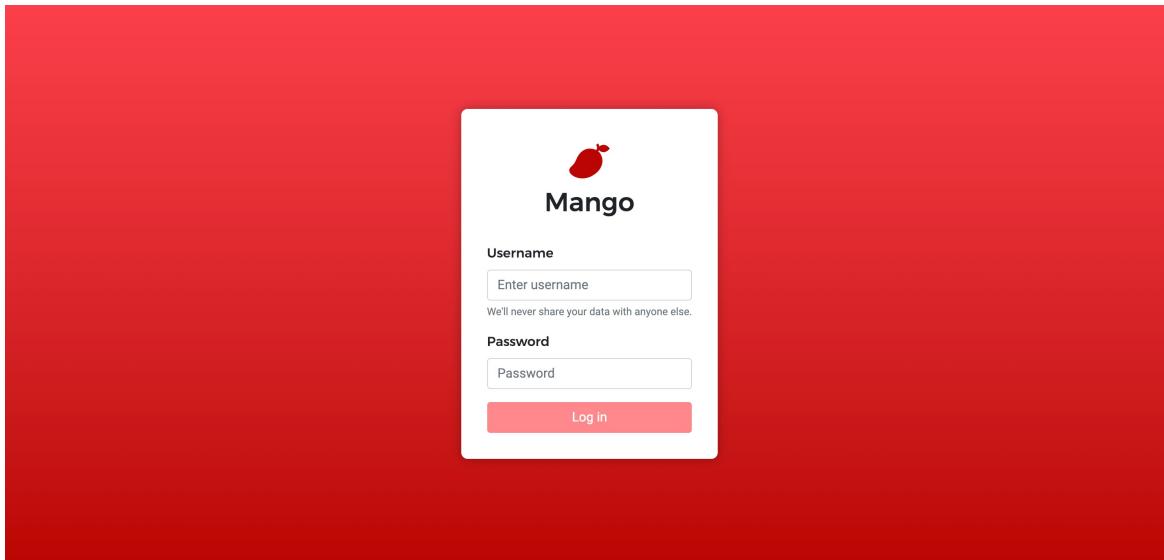


Figure 9: Login printscreencapture

4 Interoperability with SAF-T

SAF-T Route	Service Description	Related Core View	Input Example	Output Example
Fiscal Year	Retrieves the fiscal year of the SAF-T file.	<i>Used in all core views.</i>	/header/fiscal-year	"2018"
Top Selling Products	Retrieves the top selling products between the two given dates.	CORE_PROCUR	/sales/top-selling-products ?start-date= <i>date</i> &end-date= <i>date</i>	[{ "ProductCode": "A0002", "ProductDescription": "Pentium 4 945+ Dual Core 3.4GHZ", "UnitPrice": 926.015, "Quantity": 10 },]
Accounts Payable	Retrieves the current accounts payable to suppliers.	CORE_PROCUR	/AccountSum/22	{ "totalCredit": 26773, "totalDebit": 523 }
Accounts Receivable	Retrieves the current accounts receivable from clients.	CORE_FINANCIAL	/AccountSum/21	{ "totalCredit": 1325, "totalDebit": 26978 }

SAF-T Route	Service Description	Related Core View	Input Example	Output Example
Cost of Goods Sold (with time pivot)	Retrieves the overall cost of goods sold, between the given start and end dates.	CORE_PROCUR	/AccountSum/61 ?start-date= <i>date</i> &end-date= <i>date</i>	{ "totalCredit": 0, "totalDebit": 1257 }
Expenses with external services (with time pivot)	Retrieves the overall expenses with external services.	CORE_FINANCIAL	/AccountSum/62 ?start-date= <i>date</i> &end-date= <i>date</i>	{ "totalCredit": 0, "totalDebit": 560 }
Earnings from sales (with time pivot)	Retrieves the earnings from sales between the two given dates.	CORE_SALES	/AccountSum/71 ?start-date= <i>date</i> &end-date= <i>date</i>	{ "totalCredit": 22060, "totalDebit": 0 }
Earnings from provided services (with time pivot)	Retrieves the earnings from provided services between the two given dates.	CORE_FINANCIAL	/AccountSum/72 ?start-date= <i>date</i> &end-date= <i>date</i>	{ "totalCredit": 542, "totalDebit": 0 }

SAF-T Route	Service Description	Related Core View	Input Example	Output Example
Top Clients (with time pivot)	Retrieves the clients who bought most items (in monetary value), between the two given dates.	CORE_SALES	/sales/top-clients ?start-date= <i>date</i> &end-date= <i>date</i>	[{ "client": "ES989922456_C", "totalPurchased": 10757.15, "nPurchases": 1 }, ...]
Top Selling Products (with time pivot)	Retrieves the products who sold the most, in descending order of earnings from sales, between the two given dates.	CORE_INVENTORY	/sales/top-selling-products ?start-date= <i>date</i> &end-date= <i>date</i>	[{ "ProductCode": "A0002", "ProductDescription": "Pentium 4 945+ Dual Core 3.4GHZ", "UnitPrice": 926.015, "Quantity": 10 }, ...]
Sales per Region (with time pivot)	Retrieves revenue from sales for each region, between the two given dates.	CORE_SALES	/sales/sales-by-region ?start-date= <i>date</i> &end-date= <i>date</i>	[{ "id": "PT", "value": 566, "netTotal": 2050074 }, { "id": "ES", "value": 1, "netTotal": 326 }, ...]
Balance Sheet	Retrieves all ledger entries, in order to list/calculate a balance sheet.	CORE_FINANCIAL	/General LedgerEntries	[{ "TransactionDate": "2018-01-31", "Description": "Caixa A - Pagamentos", "TransactionType": "N", "Lines": { "CreditLine": ..., "DebitLine": ... } ... }]

5 Interoperability with Primavera

Webservice ID	Webservice Description	Related Core View	Input Example WebApi	Body	Output Example
Top Clients	Returning relevant information about clients	CORE_SALES	<code>{{apiUrl}}</code> Administrador/ Consulta	SELECT Cliente, Nome, TotalDeb FROM V_Clientes	{ "Cliente" : "ALCAD", "Nome" : "Solucoes Cad de Madrid", "TotalDeb" : 20318.25, }
Product Information	Returning relevant information about products	CORE_PRODUCT	<code>{{apiUrl}}</code> Administrador/ Consulta	SELECT Artigo, Descricao, PCMedio as PrecoMedio, PCUltimo as PrecoUltimo FROM Artigo	{ "Artigo" : "A001", "Descricao" : "RAM 4GB", "PrecoMedio" : 30.25, "PrecoUltimo" : 25.40 }

Webservice ID	Webservice Description	Related Core View	Input Example WebApi	Body	Output Example
Product Stock	Returns the stock of products and its warehouse	CORE_PRODUCT	{{apiUrl}} Administrador/ Consulta	SELECT Artigo, Armazem , SUM(StkActual) as StkArmazem FROM V_INV_Artigo Armazem Resumido GROUP BY Artigo, Armazem	{ "Artigo" : "A001", "StkArmazem" : 45, "Armazem" : "A1" }
Top Suppliers	Returning relevant information about suppliers	CORE_PROCUR	{{apiUrl}} Administrador/ Consulta	SELECT Fornecedor, Nome, TotalDeb FROM V_Fornecedores	{ "Fornecedor" : "F001", "Nome" : "Papalaco e Papeis", "TotalDeb" : 25004.50, }
Product PVPs	Returning PVPs of a Product	CORE_PRODUCT	{{apiUrl}} Administrador/ Consulta	SELECT Artigo ,PVP1 ,PVP2 ,PVP3 ,PVP4 ,PVP5 ,PVP6 FROM ArtigoMoeda	{ "Artigo" : "A001", "PVP1" : 40.56, "PVP2" : 50.00, "PVP3" : 27.50, "PVP4" : 75.44, "PVP5" : 27.80, "PVP6" : 25.60, }

Webservice ID	Webservice Description	Related Core View	Input Example WebApi	Body	Output Example
Purchases	Returns purchases to suppliers	CORE_PROCUR	<pre>{ { apiUrl } } Administrador/ Consulta</pre>	<pre>SELECT A.Artigo ,B.Entidade ,A.DataEntrega ,A.Quantidade ,A.PrecUnit ,A.Armazem ,A.Descricao FROM LinhasCompras AS A JOIN CabecCompras AS B on A.IdCabecCompras = B.Id WHERE A.Artigo = 'A0001' AND B.TipoDoc = 'ECF'</pre>	<pre>{ "Artigo" : "A001", "Entidade" : "F001", "DataEntrega" : "12/10/2018", "Quantidade" : 10, "PrecUnit" : 20.45, "Armazem" : "A1", "Descricao" : "Produto XYZ" }</pre>

6 Paths

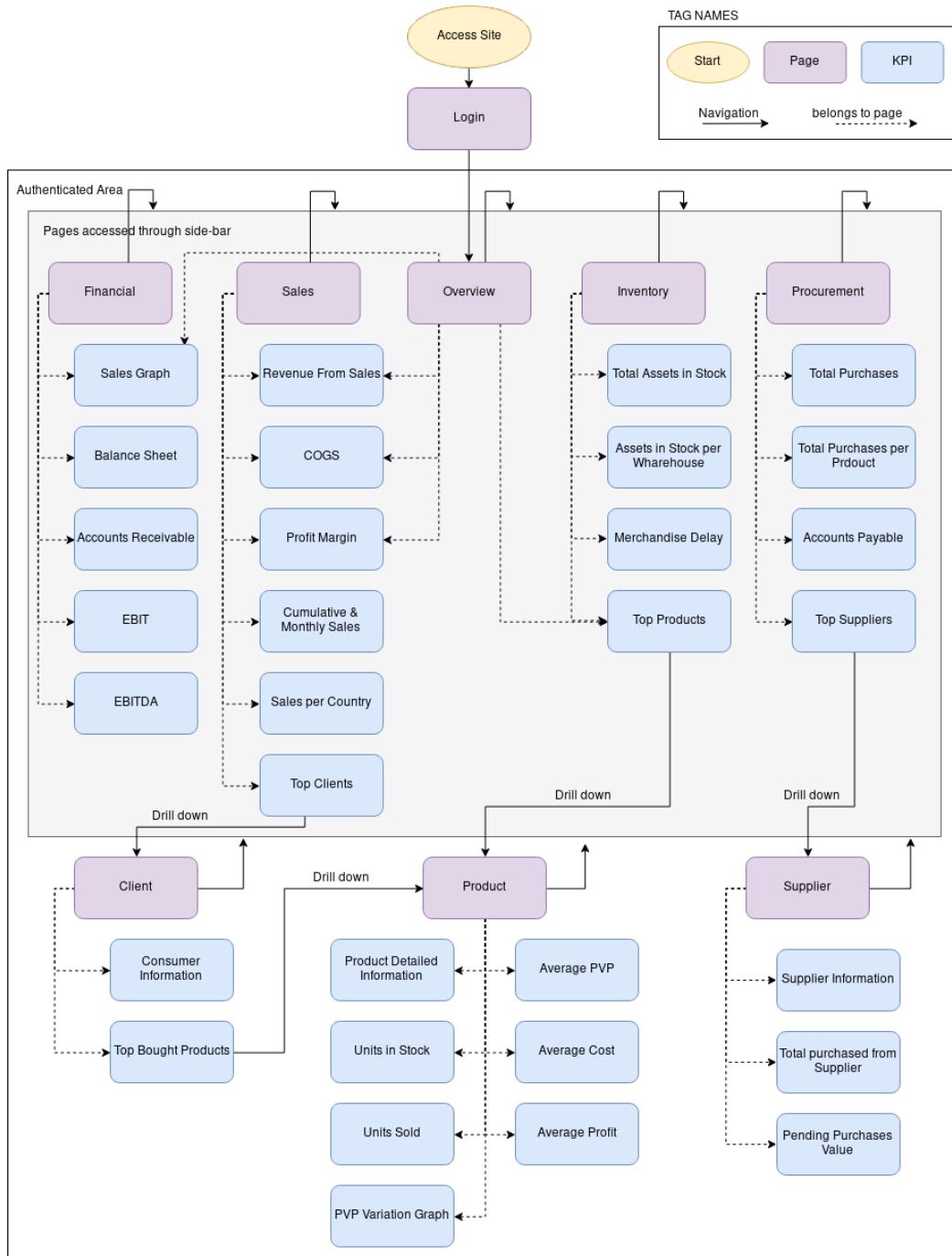


Figure 10: Site paths

Briefly explaining the site paths in figure 10, an unauthenticated user can only access the Login page. After authenticating, the user gets redirected to the Overview page. There, the User can access all the other core views, using the sidebar menu (represented through the grey box wrapping all the components accessible using the sidebar).

In the Inventory and Overview page, one can reach the Product page, using the **Top Products KPI drill down**. The Client page can be reached through the **Top Consumers KPI drill down** in the Sales page. In the Client page, the user can also access the Product page, using the **Top Bought Products KPI drill down**. Lastly, using the **Top Sales KPI drill down** in the Procurement page, one can reach the Supplier page.

All the pages are accessible through their respective *URL*, as well. As a developer you can check the application routing configuration in the *src/app/app-routing.module.ts*. The application pages' *URLs* are:

- Overview page: /
- Financial page: /financial
- Sales page: /sales
- Procurement page: /procurement
- Inventory page: /inventory
- Product page: /product/:id
- Client page: /consumer/:id
- Supplier page: /supplier/:id
- Login page: /login

To ensure an authenticated user can not access authenticated areas and vice-versa we use Angular Guards. This guards, and authentication in general, use the **token obtained from Primavera's Web API** to manage themselves.

7 System Architecture

In order to accomplish the proposed system, we developed a frontend application based on *Angular* and *TypeScript*, as well as a backend server and its corresponding Web API for supplying information from the SAF-T file. As such, the frontend application is stateless, and simply sends HTTP requests to either the SAF-T Web API or the Primavera Web API, as each page is loaded with information.

The backend SAF-T server was developed with *node.js*, using the SAF-T file as a JSON database. This server exposes several GET routes, which group information from the database in order to facilitate manipulation of information on the frontend application (as seen in Section 4).

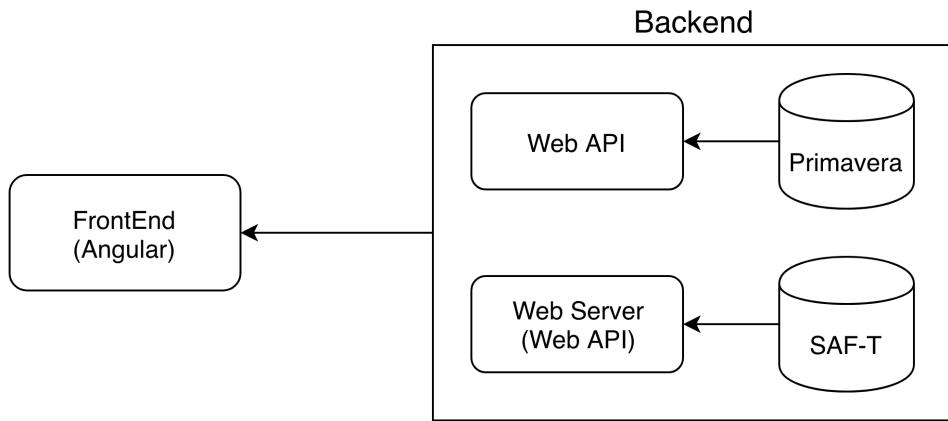


Figure 11: Simplified architecture

Doing an in-depth analysis of the frontend architecture, visible in figure 12, one can easily identify three types of structures: **services**, **components** and **pages**. Other structures were used, namely **guards** and **resolvers** but their impact in the overall structure can be disregarded.

7.1 Services

In the application were used 4 different services worth of highlight:

1. The **saftApi service**, responsible for all communication with the server serving the *SAF-T*. A component would only have to inject this service and call its method `get(url: string)` with the corresponding server route to fetch data from the *SAF-T*.

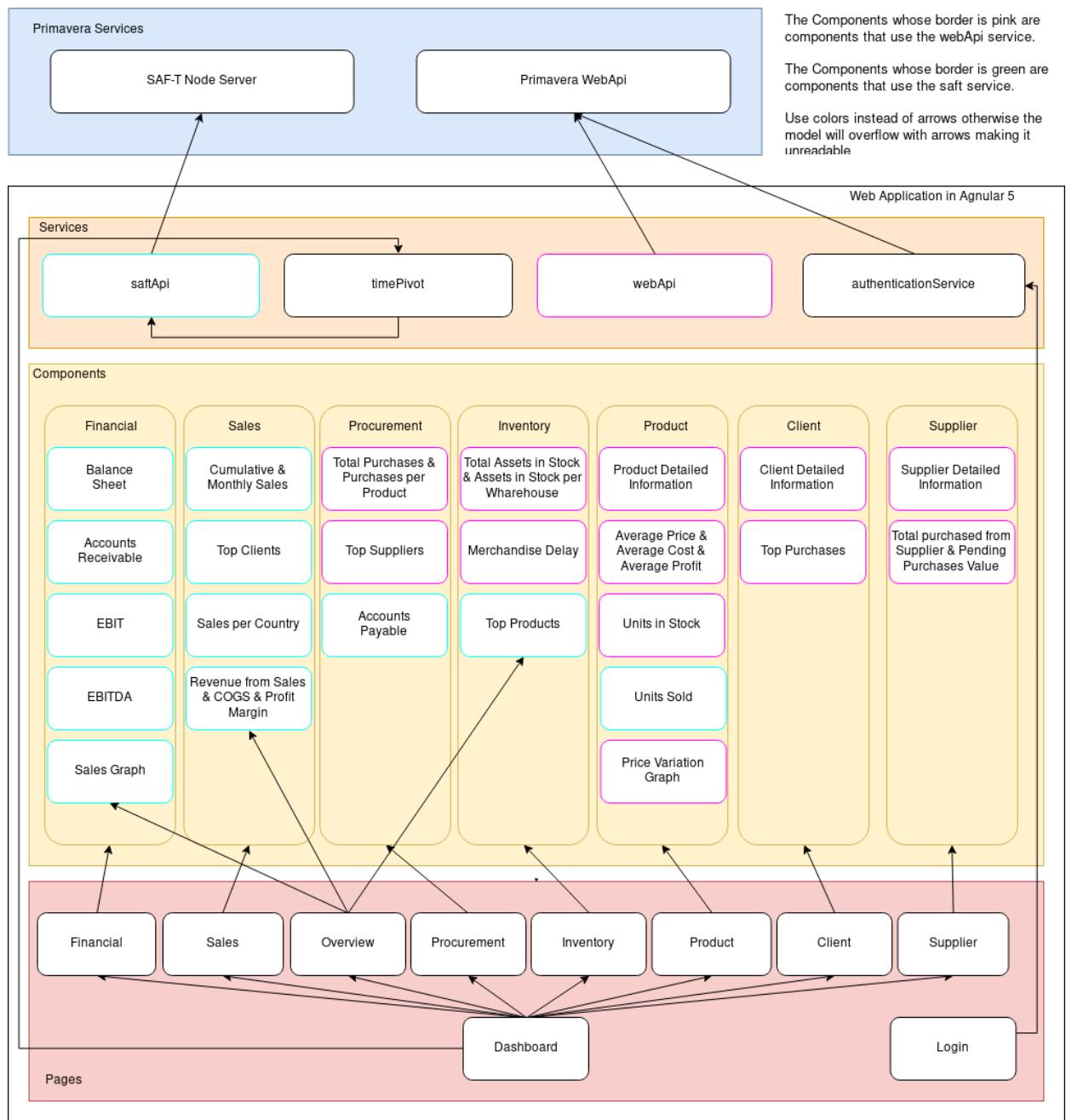


Figure 12: Frontend's detailed architecture

2. The **webApi service**, responsible for all communication with the server serving the Primavera's web API tool. Again, the component would have to inject this service and then call its methods, either `get(url: string)` or `post(url: string, body: object)` depending on the server route. This service is also responsible for in case of receiving a 401 status code as an answer to the request made, resend the authentication request to get a new token and then resending the initial request with the updated valid token.
3. The **timeFrame service**, responsible for managing the time pivot present in all the views, that would later be inserted as parameters in the saf-t requests.
4. The **authentication service**, responsible for handling all the authentication and requiring valid tokens to the web API.

7.2 Pages

Pages are Angular components that can be accessed through an url (as seen in section 6). Pages will contain components that represent their KPIs.

7.3 Components

Components are Angular components that feature in a page and wrap a KPI. Naturally, to wrap a KPI the component will have to be responsible for getting its data from either the web API or the SAF-T server. Hence, components will inject the respective service and use it's method to get the desired data.

In figure 12 components have different border colors to indicate where their data come from: pink border means the component gets its data from the `webApi` service; green border from the `safTApi` service. We chose to use border colors instead of arrows since arrows would overflow the model, thus making it unreadable.

8 Project Specifications vs Delivered Project

In comparison with the project specification, we achieved 100% of what we planned. Although, during the development of this project, our teacher suggested the addition or change of a few KPIs, leading to overall improvements of the delivered project. This way we have a more well-rounded approach with a focus not just on the financial view of the company, but also on procurement, inventory, and sales.

More specifically, there were improvements/changes to about 25% of the KPIs, as per the advice of our teacher. For instance, we now show the EBIT and EBITDA values, as these reflect different views on the financial health of the company (only the EBIT KPI was initially planned). Furthermore, the "top sales" component was scrapped in favour of the "top clients" component, as this shows information that we now see as more useful to a manager.

9 Lessons Learned

The most valuable lessons we learned was how a system like *Primavera* works, which types of data that are most valuable for a company and how to best represent them.

Throughout the project, doubts were raised by us about the value that the graphics and metrics presented in the application would add to the project. After long deliberation and with the help of our teacher, we believe that we learned to think in a more strategic way about how to use the values that an information system contains.

There were a few challenges during the development of this project. Mostly the **interoperability with Primavera**, which works on an endless amount of data and to which we had to adapt in order to obtain the relevant data for the application. The **interoperability with SAF-T** was easier because we developed a tool that converts a SAF-T XML file into a JSON database. Also, it makes some adjustments and serves it through json-server.

The fact that we used *Angular* as a frontend web application framework was a great choice. Not only did a few of us already know how to work with it, but also it provided us tools that improved the way we presented information.

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