

# Sales Order Picking

## Final Report

### **Class 1 Group B**

Francisca Leão Cerquinho Ribeiro da Fonseca | 201505791 | up201505791@fe.up.pt

Luís Miguel Santos Monteiro Saraiva | 201404302 | up201404302@fe.up.pt

Maria Eduarda Santos Cunha | 201506524 | up201506524@fe.up.pt

Maria Francisca Azevedo Paupério | 201403785 | up201403785@fe.up.pt

Mariana Lopes da Silva | 201506197 | up201506197@fe.up.pt

23rd December 2018

## **Index**

<b>Overview</b>	<b>3</b>
<b>Core Views</b>	<b>4</b>
CV_Picker	5
CV_Manager	10
<b>Interoperability with Primavera</b>	<b>13</b>
Webservices	13
<b>System Architecture</b>	<b>27</b>
<b>Project Specifications vs Delivered Project</b>	<b>27</b>
<b>Lessons Learned</b>	<b>29</b>
<b>Anex</b>	<b>30</b>
Locations and Check Digits	30

## Overview

In general, this project consisted of creating a Web App that allowed users to make a picking process of a sales order.

We considered the Web App has two types of users, namely the picker and manager, in which the latter is responsible for assigning sales orders to the existing pickers and the former receives the appointed set of sales orders.

This product is based on the process that is implemented in Sonae's Modelo Continente, allowing the pickers to pick several products that are listed in a picking wave, as well as reduce the decision-making load of the picking process and increase productivity. Pickers are given a product to pick, where it is located and the quantity to pick, making the process much more automated.

More specifically, we opted for the PBS process, which stands for Picking by Store. In this process, each sales order corresponds to an order commissioned by a store, which includes several products in various quantities. Pickers are appointed to one or more sales orders and pick its products a sales order at a time. This means, once all the products in a sales order have been picked, the picker moves on to the next sales order and so on until all have been tackled.

## Core Views

In this section, we identify two core views, one for pickers and another one for managers. For each core view, we have a variety of pages and we will specify the user and business goals, inwards and outward paths and core elements for each page.

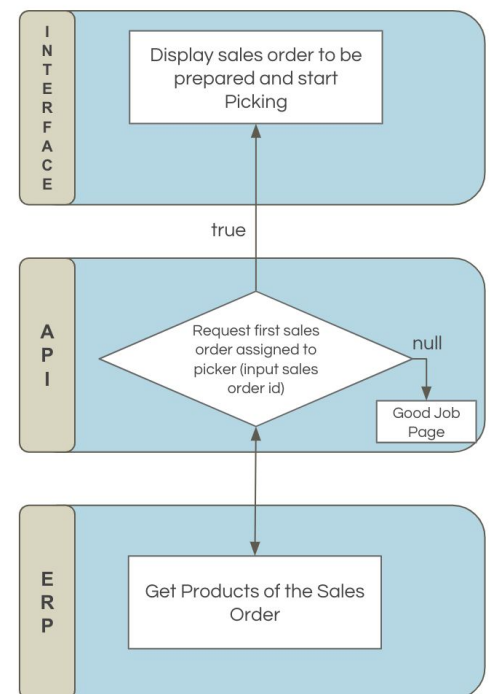
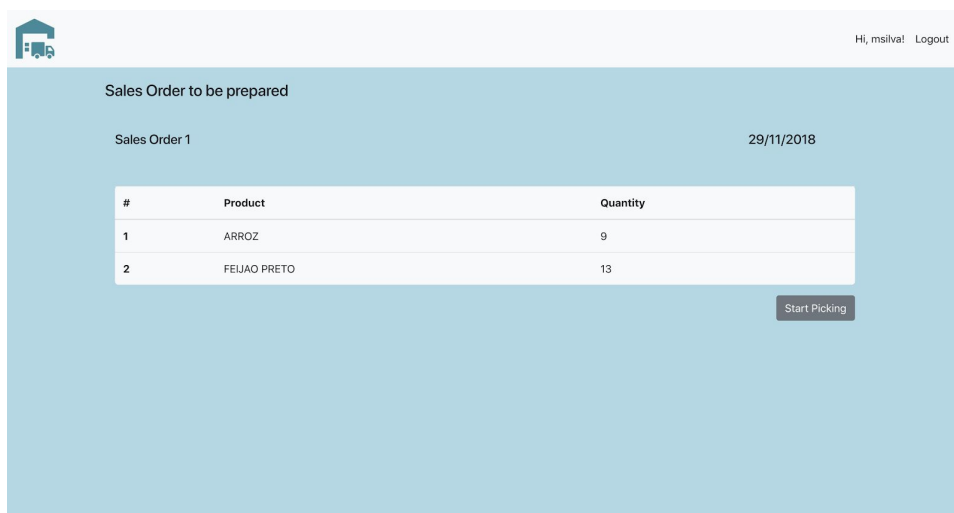


## CV\_Picker

### Page 1

As a picker, I should be able to view the products of the sales order I will prepare and its quantities.

<b>User (Picker) Goals</b>	View sales order to be prepared (products and their quantities).
<b>Business Goals</b>	Start picking a sales order.
<b>Inward Paths/Trigger words</b>	Sign In Page
<b>Core elements</b>	Get_Sales_Order_ID
<b>Calls to Action/Outward Paths</b>	Start picking button -> Product Location Page Logout button -> Sign In Page



## Page 2

As a picker, I have to enter the location code of the product that I am being told to pick, because my manager wants to check if I am in the right position.

<b>User (Picker) Goals</b>	Enter the product location (check digit).
<b>Business Goals</b>	Do the picking of the product. Check product's location digit.
<b>Inward Paths/Trigger words</b>	Sales Order (to be prepared) Page
<b>Core elements</b>	Set_Check_Digit
<b>Calls to Action/Outward Paths</b>	OK button -> Product Units Page

Product Location A1.2.1

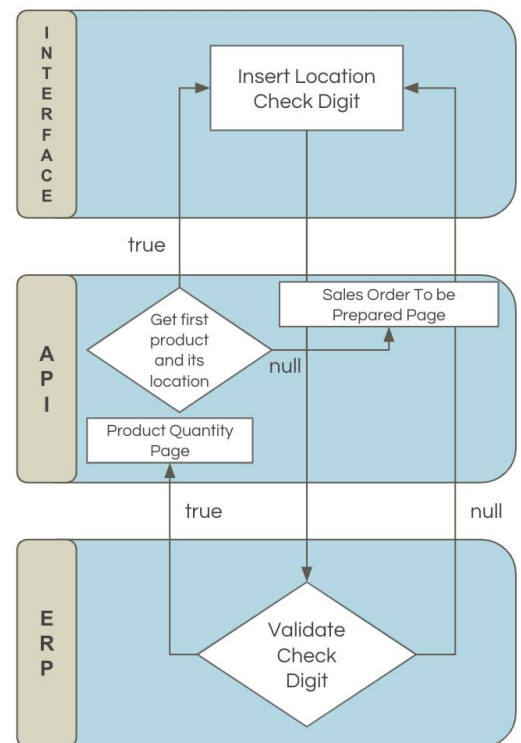
Product ARROZ

Date

Enter location's check digit:

Check Digit

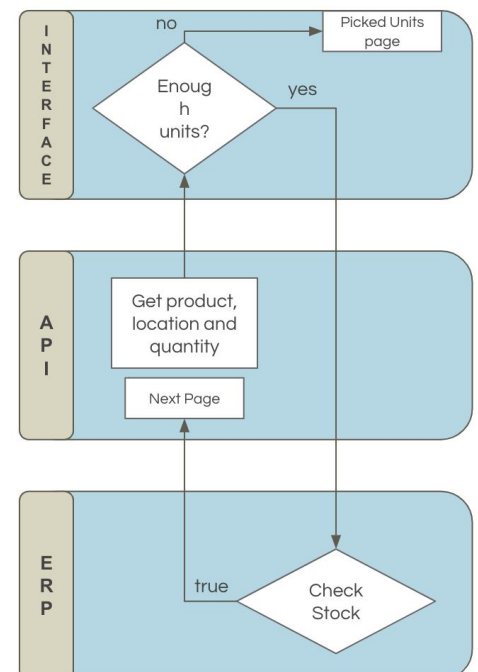
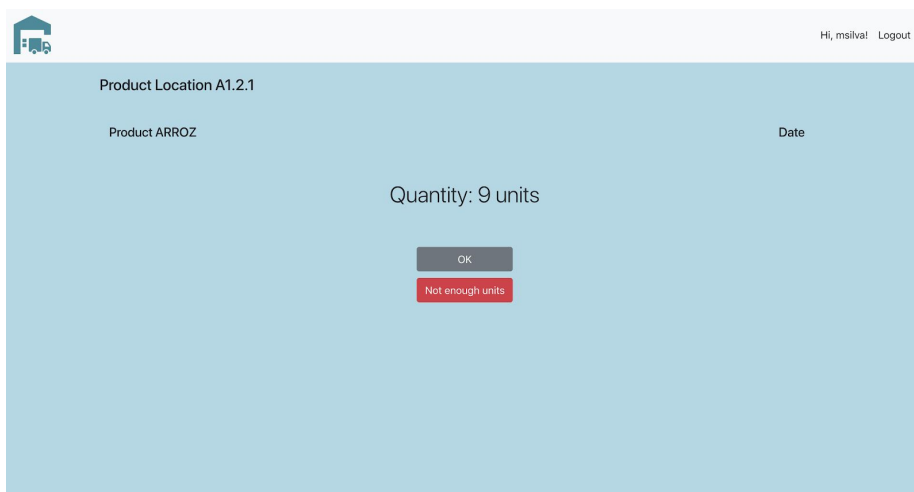
OK



## Page 3

As a picker, I should be able to view the product's needed quantity to be picked.

<b>User (Picker) Goals</b>	Picker views number of product units to pick.
<b>Business Goals</b>	Inform number of product units to pick.
<b>Inward Paths/Trigger words</b>	Product Location page
<b>Core elements</b>	Get_Product_Units
<b>Calls to Action/Outward Paths</b>	Not enough units button -> Picked Units Page OK button (finished) -> Success Page OK button (next) -> Sales Order (to be prepared) Page

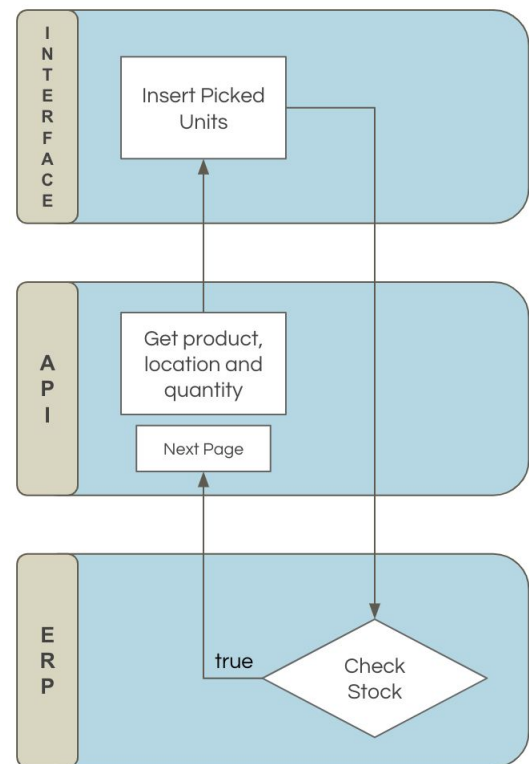


## Page 4

As a picker, I want to enter the number of picked units if there is not enough quantity to satisfy the sales order.

<b>User (Picker) Goals</b>	Picker inserts the available quantity to pick.
<b>Business Goals</b>	Enter product units to pick.
<b>Inward Paths/Trigger words</b>	Product Units page
<b>Core elements</b>	Set_Picked_Units
<b>Calls to Action/Outward Paths</b>	OK button (finished) -> Success Page OK button (next) -> Sales Order (to be prepared) Page

The screenshot shows a web application interface for a picker. At the top, there is a header with a home icon, the text "Product A1.2.1", and a user greeting "Hi, msilva! Logout". Below the header, the main content area has a light blue background. It displays "Product ARROZ" and a "Date" field. In the center, there is a form with the text "Quantity: 9 units" and "Picked: [Picked Units] units". Below the form is an "OK" button.

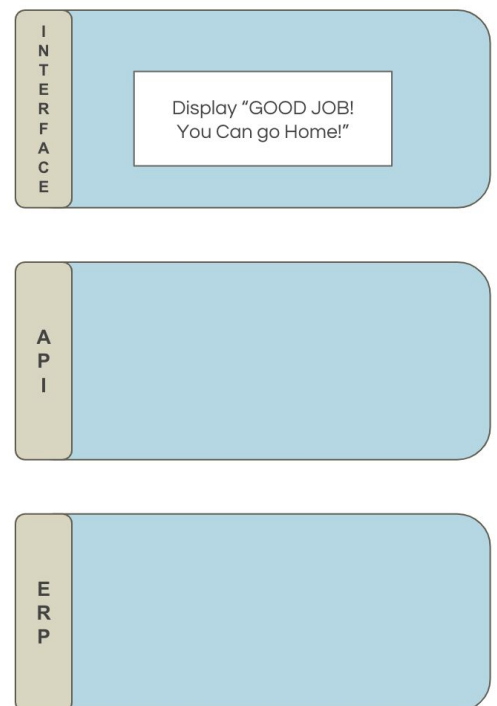
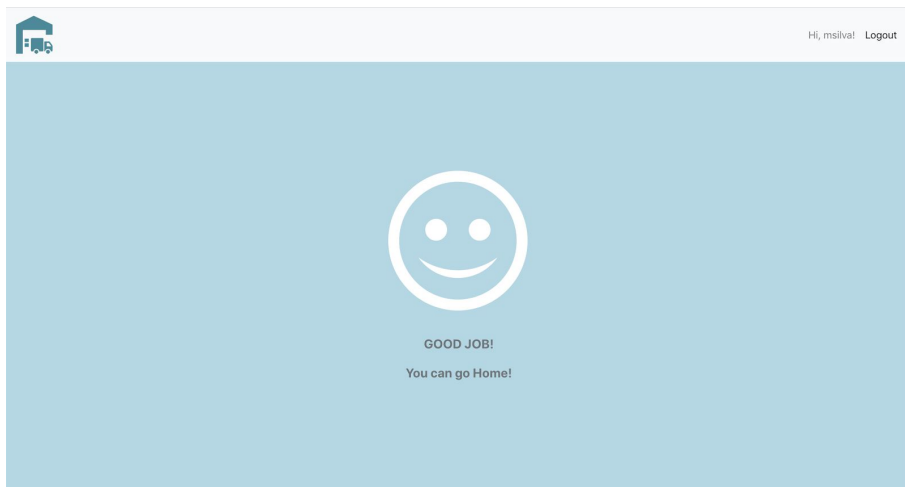




## Page 5

As a picker, when I finish to prepare a sales order, I'll receive a success message and the system will generate a delivery note.

<b>User (Picker) Goals</b>	Finish sales order picking.
<b>Business Goals</b>	Inform the picker that he has finished the job.
<b>Inward Paths/Trigger words</b>	Product Units Page Picked Units Page
<b>Core elements</b>	Create_Delivery_Note
<b>Calls to Action/Outward Paths</b>	Logout button -> Sign In Page

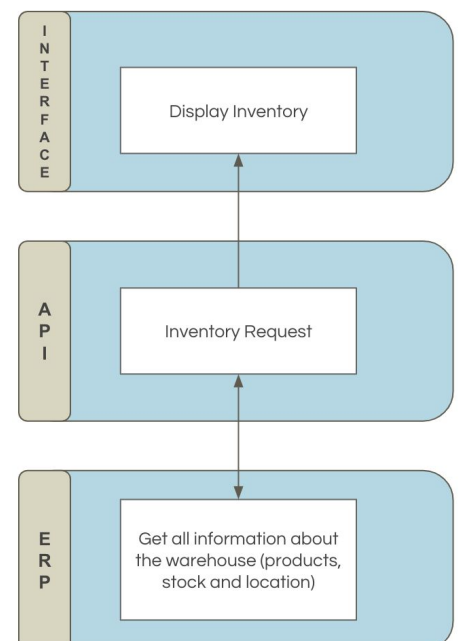
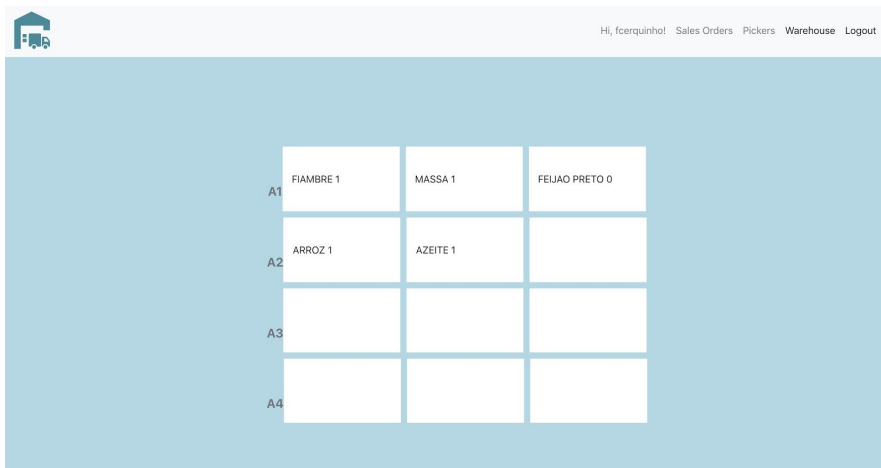


## CV\_Manager

### Page 1

As a manager, I want to view the layout of the warehouse and its products distribution and available quantity.

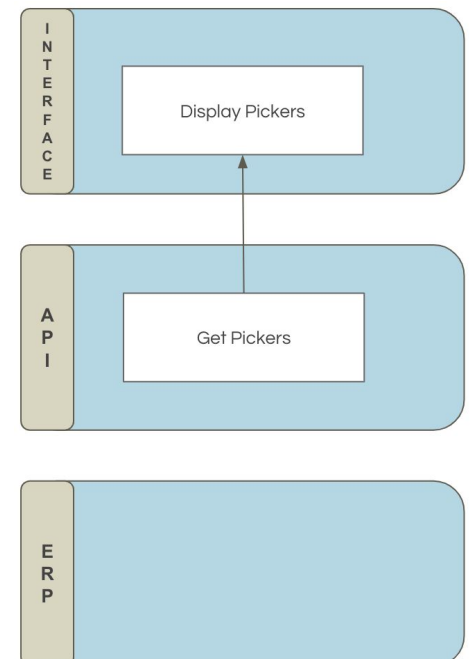
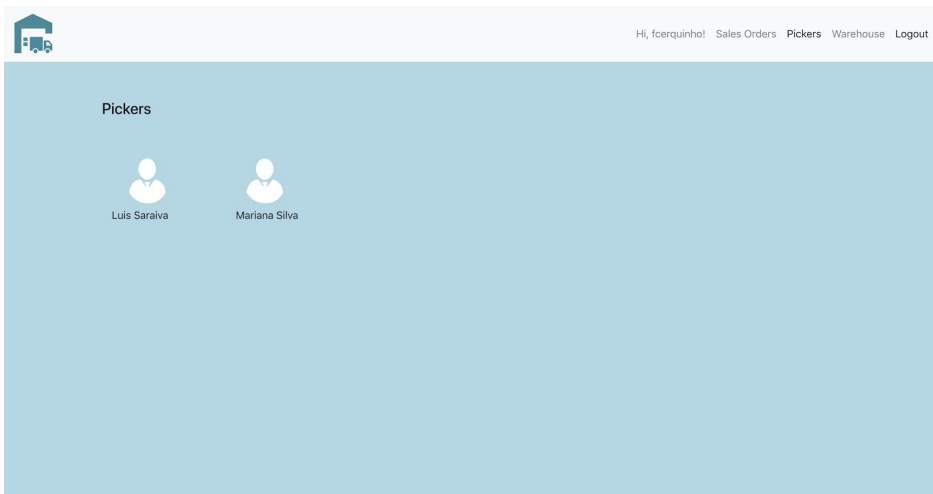
<b>User (Manager) Goals</b>	View the warehouse layout. View products distribution.
<b>Business Goals</b>	View the quantity of each product in its location.
<b>Inward Paths/Trigger words</b>	Sign In Page Sales Orders Page Pickers Page
<b>Core elements</b>	Get_Inventory
<b>Calls to Action/Outward Paths</b>	Sales Orders button -> Sales Orders Page Pickers button -> Pickers Page Logout button -> Sign In Page



## Page 2

As a manager, I want to check all the pickers.

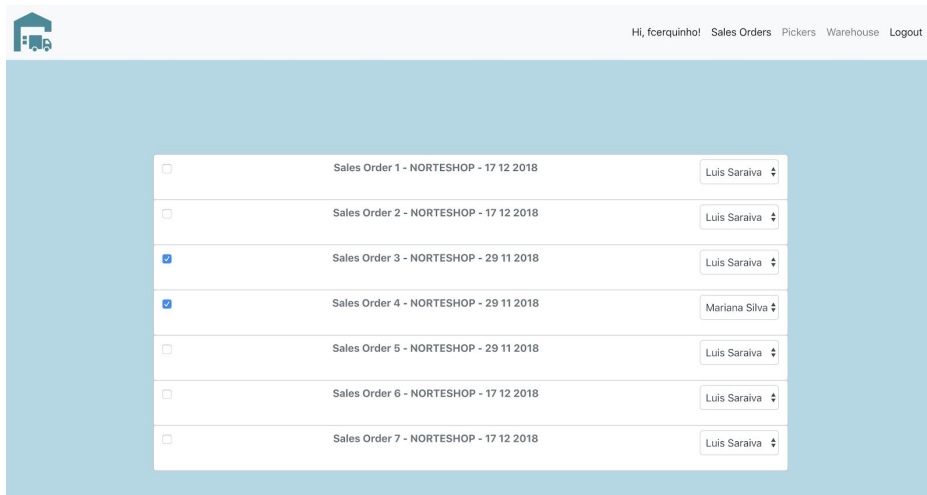
<b>User (Manager) Goals</b>	List the pickers.
<b>Business Goals</b>	Analyse pickers.
<b>Inward Paths/Trigger words</b>	Sales Orders Page Warehouse Page
<b>Core elements</b>	Get_Pickers
<b>Calls to Action/Outward Paths</b>	Sales Orders button -> Sales Orders Page Warehouse button -> Warehouse Page Logout button -> Sign In Page



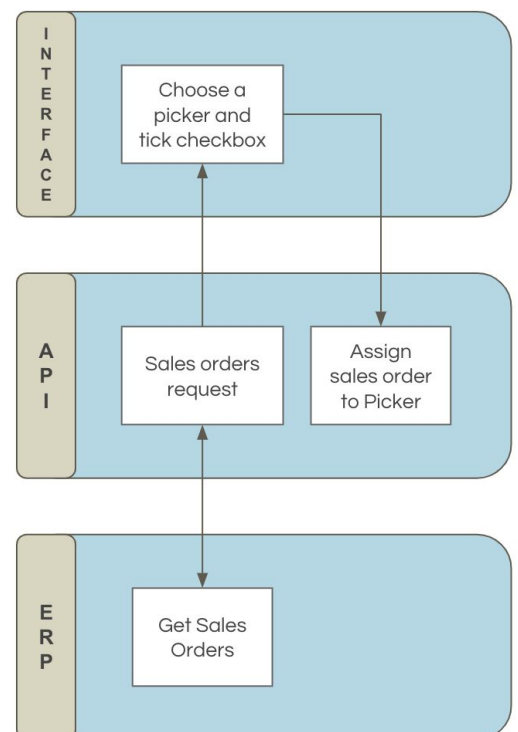
## Page 3

As a manager, I should be able to view all open sales orders and the pickers chosen for each one. Pickers may be altered. When desired choice is made, select the sale order to update.

<b>User (Manager) Goals</b>	Assign pickers to sales orders.
<b>Business Goals</b>	Manage sales orders.
<b>Inward Paths/Trigger words</b>	Pickers Page Warehouse Page
<b>Core elements</b>	Get_Sales_Orders, Assign_Pickers
<b>Calls to Action/Outward Paths</b>	Pickers button -> Pickers Page Warehouse button -> Warehouse Page Logout button -> Sign In Page



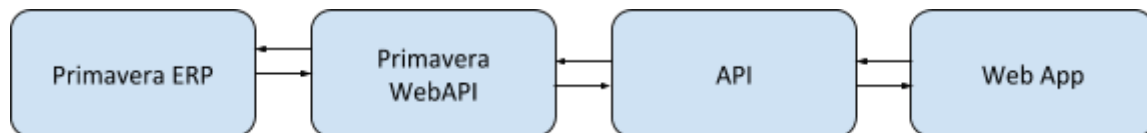
	Sales Order	Picker
<input type="checkbox"/>	Sales Order 1 - NORTESHOP - 17 12 2018	Luis Saraiva
<input type="checkbox"/>	Sales Order 2 - NORTESHOP - 17 12 2018	Luis Saraiva
<input checked="" type="checkbox"/>	Sales Order 3 - NORTESHOP - 29 11 2018	Luis Saraiva
<input checked="" type="checkbox"/>	Sales Order 4 - NORTESHOP - 29 11 2018	Mariana Silva
<input type="checkbox"/>	Sales Order 5 - NORTESHOP - 29 11 2018	Luis Saraiva
<input type="checkbox"/>	Sales Order 6 - NORTESHOP - 17 12 2018	Luis Saraiva
<input type="checkbox"/>	Sales Order 7 - NORTESHOP - 17 12 2018	Luis Saraiva



## Interoperability with Primavera

The communication of our web app with Primavera ERP is done through our API and Primavera's Web API.

Essentially, our web app makes the necessary requests for each core view to our API. In turn, our API communicates with Primavera's Web API, which returns the necessary information back to our API. Once that information is received, our API informs the web app.



## Webservices

### WS1\_Access\_Token

<b>Webservice ID</b>	WS1_Access_Token
<b>Webservice Description</b>	Authenticate in the ERP
<b>Route</b>	POST
<b>Input Example WebAPI</b>	<b>/WebApi/token</b> HTTP/1.1 <b>Host:</b> {{PRIMAVERA_API_URL}} <b>Content-Type:</b> application/x-www-form-urlencoded <b>cache-control:</b> no-cache username=FEUP&password=qualquer1& company=CONTINENTE&instance=DEFAULT&grant_type =password&line=professional
<b>Expected Output WebAPI</b>	{ <b>"access_token":</b> "OfRJrQ...", <b>"token_type":</b> "bearer", <b>"expires_in":</b> 1199 }
<b>Input Example API</b>	<b>/login</b> HTTP/1.1 <b>Host:</b> {{EXPRESS_API_URL}} <b>Content-Type:</b> application/json <b>cache-control:</b> no-cache { <b>"username":</b> "Isaraiva", <b>"password":</b> "12345678" }

Expected Output API	<pre>{   "user": {     "id": 4,     "username": "msilva",     "name": "Mariana Silva",     "userType": "picker"   },   "primavera_token": {     "access_token": "SrxRnNW...",     "token_type": "bearer",     "expires_in": 1199   } }</pre>
---------------------	--

## WS2\_Get\_Sales\_Order

Webservice ID	WS2_Get_Sales_Order
Webservice Description	Get the sales order to be prepared.
Related Core Views	CV_Picker
Route	GET
Input Example WebAPI	<p><b>/WebApi/Administrador/Consulta</b>  HTTP/1.1  <b>Host:</b> {{PRIMAVERA_API_URL}}  <b>Content-Type:</b> application/json  <b>Authorization:</b> Bearer TBu02n...  <b>cache-control:</b> no-cache</p> <p><b>{ query: `SELECT CabecDoc.Data, LinhasDoc.Artigo, LinhasDoc.Quantidade, Localizacao, CabecDoc.Entidade, LinhasDoc.IdCabecDoc, LinhasDoc.Id</b>  <b>FROM LinhasDoc, CabecDoc WHERE</b>  <b>CabecDoc.Id=LinhasDoc.IdCabecDoc</b>  <b>AND Artigo IS NOT NULL</b>  <b>AND LinhasDoc.IdCabecDoc='` + req.body.idCabecDoc + `</b>  <b>+ assignSalesOrder.id_salesOrder + `')`};</b></p>
Expected Output WebAPI	<pre>{   "DataSet": {     "Table": [       {         "Data": "2018-11-29T00:00:00",         "Artigo": "ARROZ",         "Quantidade": 1,</pre>

	<pre> "Localizacao": "A1", "Entidade": "NORTESHOP", "IdCabecDoc": "8799f449-f3e7-11e8-bd69-...", "Id": "8799f444-f3e7-11e8-bd69-..." }, {   "Data": "2018-11-29T00:00:00",   "Artigo": "FEIJAO PRETO",   "Quantidade": 1,   "Localizacao": "A1",   "Entidade": "NORTESHOP",   "IdCabecDoc": "8799f449-f3e7-11e8-bd69-...",   "Id": "8799f448-f3e7-11e8-bd69-..." } ] }, "Query": "System.Data.SqlClient.SqlCommand" } </pre>
<b>Input Example API</b>	<pre> /getSalesOrder HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache </pre>
<b>Expected Output API</b>	<pre> {"products": [   {     "Data": "2018-12-17T00:00:00.000Z",     "Artigo": "MASSA",     "Quantidade": 10,     "IdCabecDoc": "1229ba85..."   },   {     "Data": "2018-12-17T00:00:00.000Z",     "Artigo": "ARROZ",     "Quantidade": 15,     "IdCabecDoc": "1229ba85..."   } ], "data": "17/12/2018" } </pre>

### WS3\_Check\_Digit

<b>Webservice ID</b>	WS3_Check_Digit
<b>Webservice Description</b>	Get check digit for specific location.
<b>Related Core Views</b>	CV_Picker

Route	GET
Input Example WebAPI	<b>/WebApi/Administrador/Consulta</b> HTTP/1.1 <b>Host:</b> {{PRIMAVERA_API_URL}} <b>Content-Type:</b> application/json <b>Authorization:</b> Bearer TBu02n... <b>cache-control:</b> no-cache { <b>query:</b> "SELECT * FROM ArmazemLocalizacoes WHERE Localizacao='" + req.body.localizacao + "' AND Descricao='" + req.body.checkDigit + "'";}
Expected Output WebAPI	{   "DataSet": {     "Table": [       {         "Id": "5632ea80-ef37-11e8-b0ab-...",         "Localizacao": "A1.1.1",         "Armazem": "A1",         "Descricao": "4875",         "Activa": true,         "Manual": true,         "IdPai": "5df71c73-ef34-11e8-b0ab-...",         "VersaoUltAct": "AAAAAAAAXg0=",         "NomeNivel": null       }     ]   },   "Query": "System.Data.SqlClient.SqlCommand" }
Input Example API	<b>/checkDigit</b> HTTP/1.1 <b>Host:</b> {{EXPRESS_API_URL}} <b>cache-control:</b> no-cache {   "location": "A1.2.1",   "checkDigit": "7653" }
Expected Output API	[[   "Id": "2e5f8812-024d-11e9-938d-08002724e16e",   "Localizacao": "A1.2.1",   "Armazem": "A1",   "Descricao": "7653",   "Activa": true,   "Manual": true,   "IdPai": "0bb57168...",   "VersaoUltAct": "AAAAAAAflg=",



	<pre> "NomeNivel": null     }   ] </pre>
--	--

#### WS4\_Check\_Stock

Webservice ID	WS4_Check_Stock
Webservice Description	Get the product stock.
Related Core Views	CV_Picker
Route	GET
Input Example WebAPI	<pre> /WebApi/Administrador/Consulta HTTP/1.1 Host: {{PRIMAVERA_API_URL}} Content-Type: application/json Authorization: Bearer TBu02n... cache-control: no-cache {   query: "SELECT Artigo, Stock FROM INV_ValoresActuaisStock WHERE Artigo='\" req.body.artigo + \"\" AND Stock&gt;=\" req.body.expectedStock ;} </pre>
Expected Output WebAPI	<pre> {   "DataSet": {     "Table": [       {         "Artigo": "ARROZ",         "Stock": 1       }     ]   },   "Query": "System.Data.SqlClient.SqlCommand" } </pre>
Input Example API	<pre> /checkStock HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache {   "product": "ARROZ",   "expectedStock": "1" } </pre>
Expected Output API	<pre> [   {     "Artigo": "ARROZ", </pre>

	<pre>         "Stock": 1       }     ] </pre>
--	---

#### WS5\_Pickers

<b>Webservice ID</b>	WS5_Pickers
<b>Webservice Description</b>	Get all pickers
<b>Related Core Views</b>	CV_Manager
<b>Route</b>	GET
<b>Input Example API</b>	<pre> /pickers HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache </pre>
<b>Expected Output API</b>	<pre> {   "Username": "Isaraiva",   "Name": "Luís Saraiva" } {   "Username": "msilva",   "Name": "Mariana Silva" } {   "Username": "fcerquinho",   "Name": "Francisca Cerquinho" } {   "Username": "fpauperio",   "Name": "Francisca Paupério" } {   "Username": "ecunha",   "Name": "Eduarda Cunha" } </pre>

#### WS6\_Sales\_Orders

<b>Webservice ID</b>	WS6_Sales_Orders
<b>Webservice Description</b>	Get all Sales Orders.
<b>Related Core Views</b>	CV_Manager

Route	GET
Input Example WebAPI	<b>/WebApi/Administrador/Consulta</b> HTTP/1.1 <b>Host:</b> {{PRIMAVERA_API_URL}} <b>Content-Type:</b> application/json <b>Authorization:</b> Bearer TBu02n... <b>cache-control:</b> no-cache { <b>query:</b> "SELECT CabecDoc.Data, LinhasDoc.Artigo, LinhasDoc.Quantidade, Localizacao, CabecDoc.Entidade, CabecDoc.Id <b>FROM</b> LinhasDoc, CabecDoc <b>WHERE</b> CabecDoc.Id=LinhasDoc.IdCabecDoc <b>AND</b> Artigo <b>IS NOT NULL ORDER BY</b> Artigo <b>ASC</b> , LinhasDoc.Data <b>DESC</b> ";}
Expected Output WebAPI	{   "DataSet": {     "Table": [       {         "Data": "2018-11-29T00:00:00",         "Artigo": "ARROZ",         "Quantidade": 1,         "Localizacao": "A1.1.2",         "Entidade": "NORTESHOP",         "IdCabecDoc": "4b2b5c51-f3e3-11e8-bd69-...",         "Id": "4b2b5c53-f3e3-11e8-bd69-..."       },       {         "Data": "2018-11-29T00:00:00",         "Artigo": "ARROZ",         "Quantidade": 1,         "Localizacao": "A1.1.2",         "Entidade": "NORTESHOP",         "IdCabecDoc": "8799f449-f3e7-11e8-bd69-...",         "Id": "8799f444-f3e7-11e8-bd69-..."       },       {         "Data": "2018-11-29T00:00:00",         "Artigo": "FEIJAO PRETO",         "Quantidade": 1,         "Localizacao": "A1.2.1",         "Entidade": "NORTESHOP",         "IdCabecDoc": "8799f449-f3e7-11e8-bd69-...",         "Id": "8799f448-f3e7-11e8-bd69-..."       },       {         "Data": "2018-11-29T00:00:00",         "Artigo": "FEIJAO PRETO",

	<pre> "Quantidade": 1, "Localizacao": "A1.2.1", "Entidade": "NORTESHOP", "IdCabecDoc": "d1635bf1-f3e6-11e8-bd69-...", "Id": "e229471b-f3e6-11e8-bd69-..."     }   ] }, "Query": "System.Data.SqlClient.SqlCommand" } </pre>
<b>Input Example API</b>	<pre> /salesOrders HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache </pre>
<b>Expected Output API</b>	<pre> [   {     "Data": "2018-11-29T00:00:00",     "Artigo": "ARROZ",     "Quantidade": 1,     "Localizacao": "A1.1.2",     "Entidade": "NORTESHOP",     "IdCabecDoc": "4b2b5c51-f3e3-11e8-bd69-...",     "Id": "4b2b5c53-f3e3-11e8-bd69-..."   },   {     "Data": "2018-11-29T00:00:00",     "Artigo": "ARROZ",     "Quantidade": 1,     "Localizacao": "A1.1.2",     "Entidade": "NORTESHOP",     "IdCabecDoc": "8799f449-f3e7-11e8-bd69-..",     "Id": "8799f444-f3e7-11e8-bd69-..."   },   {     "Data": "2018-11-29T00:00:00",     "Artigo": "FEIJAO PRETO",     "Quantidade": 1,     "Localizacao": "A1.2.1",     "Entidade": "NORTESHOP",     "IdCabecDoc": "8799f449-f3e7-11e8-bd69-...",     "Id": "8799f448-f3e7-11e8-bd69-..."   },   {     "Data": "2018-11-29T00:00:00",     "Artigo": "FEIJAO PRETO",     "Quantidade": 1,     "Localizacao": "A1.2.1", </pre>

	<pre>         "Entidade": "NORTESHOP",         "IdCabecDoc": "d1635bf1-f3e6-11e8-bd69-...",         "Id": "e229471b-f3e6-11e8-bd69-..."       }     ] </pre>
--	--

#### WS7\_Assign\_Sales\_Order

<b>Webservice ID</b>	WS7_Assign_Sales_Order
<b>Webservice Description</b>	Assigns a sales order to a picker.
<b>Related Core Views</b>	CV_Manager
<b>Route</b>	POST
<b>Input Example API</b>	<pre> /assignSalesOrder HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache {   "username_picker": "Isaraiva",   "id_salesOrder": "d1635bf1-f3e6-11e8-bd69-...", } </pre>
<b>Expected Output API</b>	Ok

#### WS8\_Get\_Product

<b>Webservice ID</b>	WS8_Get_Product
<b>Webservice Description</b>	Get all product information.
<b>Related Core Views</b>	CV_Picker
<b>Route</b>	POST
<b>Input Example API</b>	<pre> /getProduct HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache {   "salesOrderId": "d1635bf1-f3e6-11e8-bd69-...", } </pre>
<b>Expected Output API</b>	<pre> {   "id_salesOrder": "1229ba85-...",   "product": "MASSA",   "quantity": 10, } </pre>

	<pre>"location": "A1.1.2", "notEnoughQuantity": false }</pre>
--	---

#### WS9\_Delete\_Product\_And\_Delivery\_Note

<b>Webservice ID</b>	WS9_Delete_Product_And_Delivery_Note
<b>Webservice Description</b>	Deletes the product that has been picked and if it's the last product of sales order creates the delivery note.
<b>Related Core Views</b>	CV_Picker
<b>Route</b>	POST
<b>Input Example API</b>	<pre>/deleteProduct HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache {   "salesOrderId": "d1635bf1-f3e6-11e8-bd69-...",   "product": "ARROZ", }</pre>
<b>Expected Output API</b>	<pre>{   "id": 2,   "id_salesOrder": "1229ba85-...",   "product": "ARROZ",   "quantity": 15,   "location": "A1.2.1",   "notEnoughQuantity": false,   "date": "2018-12-17T00:00:00.000Z",   "createdAt": "2018-12-19T14:54:04.238Z",   "updatedAt": "2018-12-19T14:57:55.438Z" }</pre>
<b>Input Example WebApi</b>	<pre>/WebApi/Vendas/Docs/TransformDocument/ECL/A/1/000/false HTTP/1.1 Host: {{PRIMAVERA_API_URL}} Content-Type: application/json Authorization: Bearer RjMG_Y... cache-control: no-cache {   "Tipodoc": "GR",   "Serie": "A",   "Entidade": "NORTESHOP",   "TipoEntidade": "C",   "DataDoc": "12/11/2018",   "DataVenc": "12/12/2018"</pre>

	}
<b>Expected Output WebApi</b>	true

### WS10\_Update\_Product

<b>Webservice ID</b>	WS10_Update_Product
<b>Webservice Description</b>	Updates product quantity that couldn't be picked.
<b>Related Core Views</b>	CV_Picker
<b>Route</b>	POST
<b>Input Example API</b>	<pre> /deleteProduct HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache {   "salesOrderId": "1229ba85-....",   "product": "MASSA",   "quantity": "1", } </pre>
<b>Expected Output API</b>	<pre> [   {     "id": 4,     "id_salesOrder": "1229ba85-...",     "product": "MASSA",     "quantity": 9,     "location": "A1.1.2",     "notEnoughQuantity": true,     "date": "2018-12-17T00:00:00.000Z",     "createdAt": "2018-12-19T15:01:07.353Z",     "updatedAt": "2018-12-19T15:01:07.353Z"   } ] </pre>

### WS11\_Warehouse

<b>Webservice ID</b>	WS11_Warehouse
<b>Webservice Description</b>	Displays warehouse layout, where each product is and its quantity.
<b>Related Core Views</b>	CV_Manager
<b>Route</b>	GET

Input Example API	/warehouse HTTP/1.1 Host: {{EXPRESS_API_URL}} cache-control: no-cache
Expected Output API	<pre>[   {     "Localizacao": "A1.1.1",     "Armazem": "A1",     "Descricao": "4875",     "product": {       "Localizacao": "A1.1.1",       "Artigo": "FIAMBRE",       "Stock": 1     }   },   {     "Localizacao": "A1.1.2",     "Armazem": "A1",     "Descricao": "9282",     "product": {       "Localizacao": "A1.1.2",       "Artigo": "MASSA",       "Stock": 1     }   },   {     "Localizacao": "A1.1.3",     "Armazem": "A1",     "Descricao": "7432",     "product": {       "Localizacao": "A1.1.3",       "Artigo": "FEIJAO PRETO",       "Stock": 0     }   },   {     "Localizacao": "A1.2.1",     "Armazem": "A1",     "Descricao": "7653",     "product": {       "Localizacao": "A1.2.1",       "Artigo": "ARROZ",       "Stock": 1     }   },   {     "Localizacao": "A1.2.2",     "Armazem": "A1",</pre>



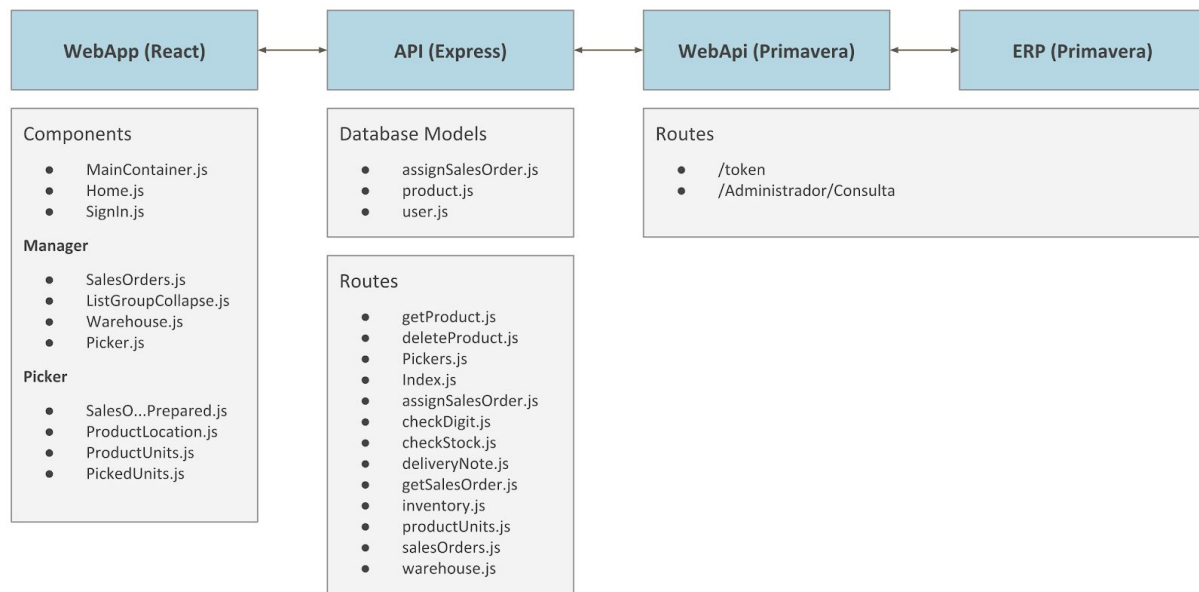
	<pre> "Descricao": "8523", "product": {   "Localizacao": "A1.2.2",   "Artigo": "AZEITE",   "Stock": 1 } }, {   "Localizacao": "A1.2.3",   "Armazem": "A1",   "Descricao": "1234",   "product": {     "Artigo": "",     "Stock": ""   } }, {   "Localizacao": "A1.3.1",   "Armazem": "A1",   "Descricao": "4327",   "product": {     "Artigo": "",     "Stock": ""   } }, {   "Localizacao": "A1.3.2",   "Armazem": "A1",   "Descricao": "2394",   "product": {     "Artigo": "",     "Stock": ""   } }, {   "Localizacao": "A1.3.3",   "Armazem": "A1",   "Descricao": "8325",   "product": {     "Artigo": "",     "Stock": ""   } }, {   "Localizacao": "A1.4.1",   "Armazem": "A1",   "Descricao": "6724",   "product": { </pre>
--	---

	<pre> "Artigo": "", "Stock": "" } }, { "Localizacao": "A1.4.2", "Armazem": "A1", "Descricao": "9531", "product": { "Artigo": "", "Stock": "" } }, { "Localizacao": "A1.4.3", "Armazem": "A1", "Descricao": "3358", "product": { "Artigo": "", "Stock": "" } } </pre>
--	--

## System Architecture

As we have referred before in the section [Interoperability with Primavera](#), our system architecture consists of an Interface, our own API and Primavera's Web API.

The WebApp was developed in React and our API in Express.



## Project Specifications vs Delivered Project

In this final delivery, we are handing over the complete implementation of the described app, which includes the web app with appropriate design developed in React Native and the communication of said app with Primavera ERP, which is established through our API and Primavera's Web API.

Our final product includes everything specified in the "Project Specifications" report but between that early draft of what our project was supposed to consist of and the end result, the idea of what we wanted to do became clearer and so there were a few changes.

Initially, we had set out to make pickers pick one product only one time even if it appeared in different sales orders, i.e., for example, if two different sales orders had the same product, the picker would pick the quantities needed for both in one turn. Now, pickers take care of one single sales order at a time.

In addition to this change, there was also an issue with how to deal with pickers working simultaneously. Essentially, we were not sure how to deal with inventory updating as several pickers might be picking similar products and the wrong quantities could be displayed if they were to pick units of the same product and the inventory was not updated instantaneously. We ultimately decided, although this way there might be conflicts regarding available quantities, that in the end of each sales order being picked a delivery

note is issued and it is responsible for updating the inventory since Primavera does not allow for instant updating as the products are being picked.

## Lessons Learned

We now believe we have a much better understanding of how a warehouse works in real life and all the difficulties that come with translating such a process to a digital environment.

What first appeared to be a very simple system to implement actually turned out to have a lot of complications. The main example of this is the issue we had considering how to update the quantities of the products picked. Whilst in real life several pickers working at the same time can see if a picker before them picked a similar product and, therefore, affected the available quantity of that product, using Primavera that becomes a much harder task as discussed in the section Project Specifications vs Delivered Product. An alternative to what we opted to do would be the creation of a fictional warehouse where the products' quantities would be updated as needed by the requirements of the existing sales orders. This would avoid the conflicts between pickers working simultaneously.

Finally, this project overall contributed to our knowledge of Enterprise Resource Planning tools, such as Primavera, and the interoperability with those tools and an API developed on the side, our own API.

## Anex

### Locations and Check Digits

Location	Check Digit
A1.1.1	9319
A1.1.2	3782
A1.1.3	7361
A1.2.1	8423
A1.2.2	5712
A1.2.3	7281
A1.3.1	2901
A1.3.2	3842
A1.3.3	4627
A1.4.1	6812
A1.4.2	9342
A1.4.3	4682

### User Credentials

Email	Password
ecunha	12345678
fcerquinho	12345678
fpauperio	12345678
lsaraiva	12345678
msilva	12345678