

Faculty of Engineering of the University of Porto

Master in Informatics and Computing Engineering

Fourth Year – First Semester



FEUP **FACULDADE DE ENGENHARIA**
UNIVERSIDADE DO PORTO

SINF Sales Force Automation

Deliverable #1 – Project Specifications

15th October 2017

Contents

Team Information	2
Project Overview	2
Functionalities and Features	3
Information Architecture (core views):	6
PINFO_CV - Product Info	6
CLNTINFO_CV - Client Info	7
SLSREP_CV - Sales Representative Info	8
SLSREP_TSKS_CV - Sales Representative Dashboard	9
Interoperability with Primavera	9
Sales Representative	10
Customer	14
Sales Order	18
Customer Visit	21
Customer Group	23
Product	25
Path	28

Team Information

Team Class: 4MIEIC02

Team Identification: Group L

Project Identification: P4 - Sales Force Automation

Team Elements:

- Ines Belac - up201701861
- António Jorge Aguiar do Vale - up201404572
- Simão Lopes Lúcio - up201503845
- Tiago José Grosso Pacheco - up201402722
- Tiago Rafael Ferreira da Silva - up201402841

Project Overview

The purpose of this project is to provide assistance in the daily routines of the salespeople's job by automating tasks when possible and presenting features which ease the work of the sales force.

Sales Force Automation is a software oriented towards inventory control, tracking scheduled and past customer interactions and managing sales orders. It is a tool designed to manage the agenda of the sales representatives.

The main features of this application include customer details for better contacts, as well as their visits, orders and scheduled interactions. By registering recurring customers on the database, the sales representatives can execute orders much quicker which is further improved upon by having a structured repository for all the orders and their respective status.

The work of the salespeople can be more effective if the extensive information which they require is well organized, easily accessible and clearly displayed . Thus, through the use of this software, an increase in productivity and performance of the sales force is expected since it provides that organization which shifts the focus of the sales force towards more productive goals than agenda managing.

Lastly, the system will record relevant data in order to track the work of the salespeople as well as product and client tendencies and it will present this information, enabling an easy assertion of the sales force conduct, product performance and client preferences.

Functionalities and Features

This section the project specification aims to list and explain the main features that will be implemented during the development phase. Therefore, the following table describes those use cases and the authorization required to make use of said functionality. There is also a reference to the core view in which the feature is expected to be accessible.

ID	NAME	DESCRIPTION	AUTHORIZATION	CORE VIEW
F01	Sales Representative Login	Authentication of the sales representative	guest	n/a
F02	Sales Representative Admin Login	Authentication of the sales representative admin	guest	n/a
F03	Sales Representative Logout	Closing the session of the sales representative	Sales Rep.	n/a
F04	Sales Representative Admin Logout	Closing the session of the sales representative admin	Sales Rep. Admin	n/a
F05	Sales Rep. Info	Get a sales rep. information	Sales Rep. Admin	SLSREP_CV
F06	New Sales Rep.	Create a sales rep. account and profile in the system	Sales Rep. Admin	n/a
F07	Edit Own Profile Information	Sales rep's are able to edit their own profile	Sales Rep.	n/a
F08	Edit Other Sales Rep's Profile Information	Edit any Sales Rep. profile information	Sales Rep. Admin	n/a
F09	Delete Sales Rep.	Delete Sales Rep. Accounts	Sales Rep. Admin	n/a
F10	Get Own Orders	Each sales rep. can get a list of all it's orders	Sales Rep.	n/a
F11	Get any Sales Rep. Orders	Get list of sales by any sales rep.	Sales Rep. Admin	SLSREP_CV
F12	Get Customer Visits	Get list of customer visits	Sales Rep.	SLSREP_TSKS_CV
F13	Get Customer Info	Get all the customer personal info	Sales Rep.	CLNTINFO_CV
F14	Add Customer	Add a new customer to the	Sales Rep.	n/a

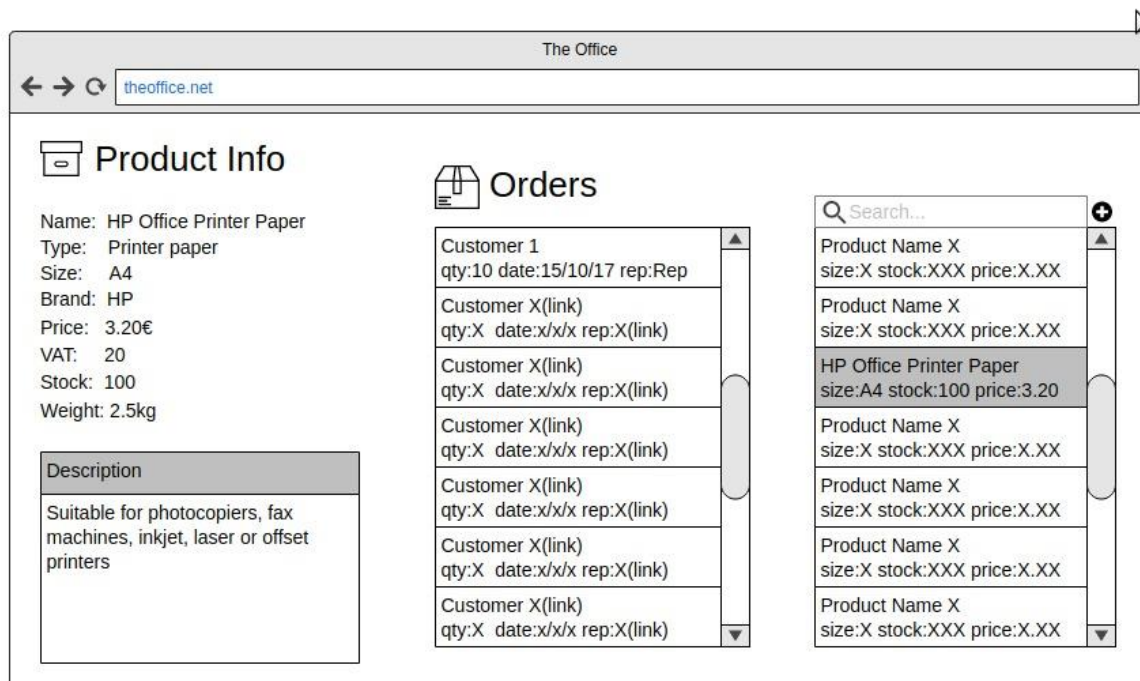
		system		
F15	Update Customer	Update customer info	Sales Rep.	n/a
F16	Delete Customer	Delete customer profile from system	Sales Rep.	n/a
F17	Get Customer Orders	Get a list of orders by a given customer	Sales Rep.	CLNTINFO_CV
F18	Get Customer Visits	Get a list of visits by a given customer	Sales Rep.	n/a
F19	Get Sale	Get info on a given sale	Sales Rep.	n/a
F20	New Sale	Register a new sale	Sales Rep.	n/a
F21	Update Sale	Update a sale registry information	Sales Rep.	n/a
F22	Delete Sale	Delete a sale from the system	Sales Rep.	n/a
F23	Get Customer Visit Info	Get info on a given customer visit	Sales Rep.	SLSREP_TSKS_CV
F24	Create Customer Visit	Register a new customer visit	Sales Rep.	SLSREP_TSKS_CV
F25	Update Customer Visit	Update a customer visit registry information	Sales Rep.	SLSREP_TSKS_CV
F26	Delete Customer Visit	Delete a customer visit from the system	Sales Rep.	n/a
F27	Get Customer Group Info	Get info on a given customer visit	Sales Rep.	n/a
F28	Create Customer Group	Register a new customer group	Sales Rep.	n/a
F29	Update Customer Group	Update a customer group registry information	Sales Rep.	n/a
F30	Delete Customer Group	Delete a customer group from the system	Sales Rep.	n/a
F31	Get Product Info	Get info on a given product	Sales Rep.	PINFO_CV
F32	New Product	Register a new product	Sales Rep.	PINFO_CV
F33	Update Product	Update a product registry	Sales Rep.	n/a

		information		
F34	Delete Product	Delete a product from the system	Sales Rep.	n/a

Information Architecture (core views):

PINFO_CV - Product Info

It's the global product overview for the sales representative to get a better grasp not only on the product but on how it's selling.



Inward paths: product search, client orders list, sales rep. sales list

Outward paths: user profile, sales rep. profile, another product info page

Elements of the core view: product description elements, list of orders of the product, list of products with search capabilities and add product button


CLNTINFO_CV - Client Info

It's the global client overview for the sales representative to see what kind of products and in which quantities he's been buying.

The Office

← → ↻ theoffice.net

Client Info



Manuel da Rocha Lima

Faculdade de Engenharia da
Universidade do Porto
4200-391 Porto

Phone: +351911991365
Email: something@else.com
Fiscal ID: 654321456
Birth Date: 1986-09-13

Edit

Description

Good and loyal customer.

Orders

HP A4 Paper	qty:10 date:15/10/17 rep:Rep
Product X (link)	qty:X date:x/x/x rep:X(link)
Product X (link)	qty:X date:x/x/x rep:X(link)
Product X (link)	qty:X date:x/x/x rep:X(link)
Product X (link)	qty:X date:x/x/x rep:X(link)
Product X (link)	qty:X date:x/x/x rep:X(link)
Product X (link)	qty:X date:x/x/x rep:X(link)
Product X (link)	qty:X date:x/x/x rep:X(link)

Search...

Client name X
status:X orders:XXX

Client name X
status:X orders:XXX

Manuel Lima
status:active orders:999

Client name X
status:X orders:XXX

Client name X
status:X orders:XXX

Client name X
status:X orders:XXX

Client name X
status:X orders:XXX

Inward paths: client search, product orders list, sales rep. sales list

Outward paths: product info, sales rep. profile, another client info page, client info edit page

Elements of the core view: client description elements, list of orders placed by the client, list of products with search capabilities and add product button

7

SLSREP_CV - Sales Representative Info

It's the global product overview for the sales representative to get a better grasp not only on the product but on how it's selling.

The screenshot displays a web application interface with a top navigation bar labeled "The Office". Below the navigation bar, there is a search bar containing the text "theoffice.net".

The main content area is divided into two columns. The left column features a profile card for "Pablo dos Santos". The card includes a placeholder image, the name "Pablo dos Santos", and contact information: "Faculdade de Engenharia da Universidade do Porto", "4200-391 Porto", "Phone: +351911991365", "Email: something@else.com", "Fiscal ID: 654321456", and "Birth Date: 1986-09-13". An "Edit" button is located next to the birth date. Below the profile card is a table with a "Description" header and a single row containing the text "Good employee. Likes this company."

The right column features a section titled "\$ Sales". Below the title is a table with a search bar labeled "Search...". The table contains a list of sales records, each with a "Sales Rep. Name X" and a "sales:XXX" value. The records are: "Sales Rep. Name X sales:XXX", "Sales Rep. Name X sales:XXX", "Pablo Santos sales:999" (highlighted), "Sales Rep. Name X sales:XXX", "Sales Rep. Name X sales:XXX", "Sales Rep. Name X sales:XXX", and "Sales Rep. Name X sales:XXX". A vertical scrollbar is visible on the right side of the table.

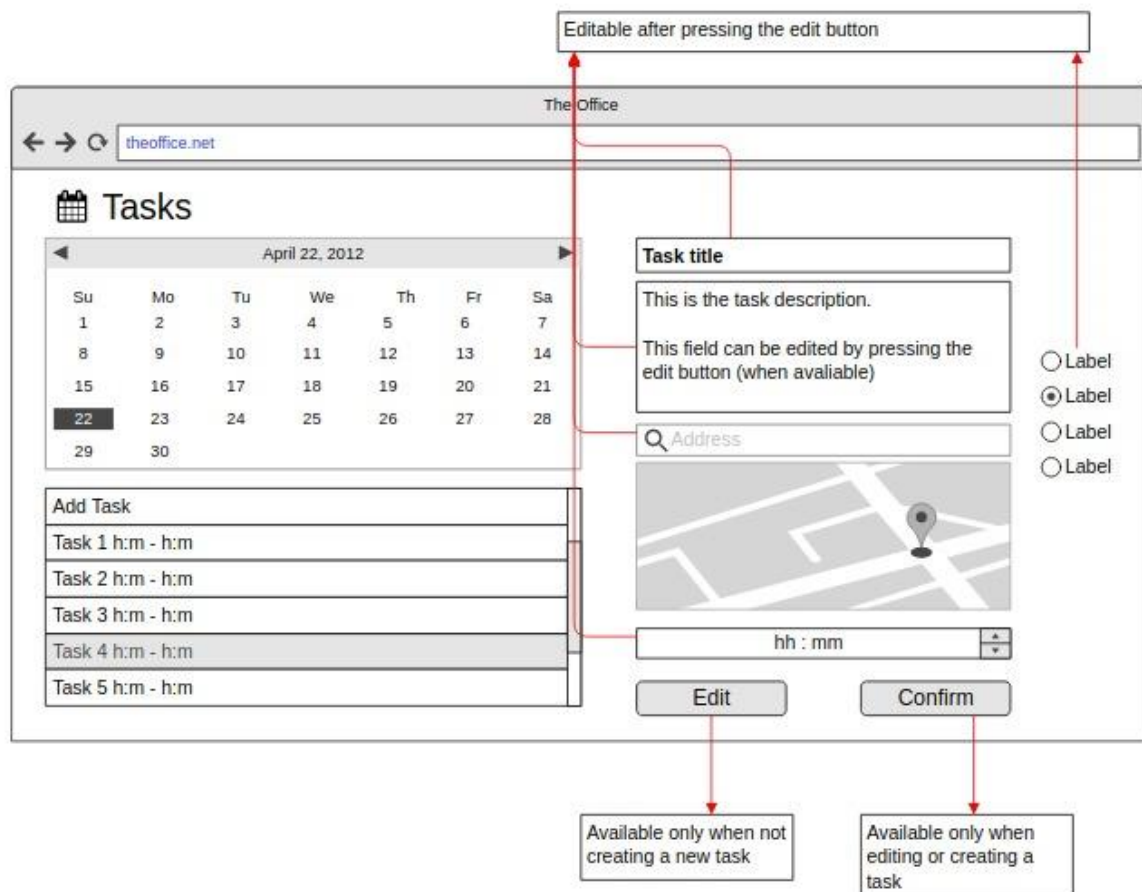
Inward paths: admin group user sales rep. search, product orders list, client orders list

Outward paths: product info, sales rep. profile, another client info page, client info edit page

Elements of the core view: sales rep. description elements, list of sales by the rep. , list of rep's with sales values and add sales rep. button

SLSREP_TSKS_CV - Sales Representative Dashboard

The main sales representative's dashboard for the weekly tasks. Here the sales reps can manage their tasks.



Inward paths: sales rep. dashboard

Outward paths: client info page (from the tasks related to sales)

Elements of the core view: sales rep. calendar, list of tasks with links and labels for the selected day, task details and option to edit and add tasks

Interoperability with Primavera

To operate with Primavera ERP we had to define several webservices. In the following tables we will describe the different methods, routes and parameters that can be used. Tables also provide examples of input and output for each webservice.

To provide a better overview, we divided them into sections:

1. [Sales Representative](#)
2. [Customer](#)
3. [Sales Order](#)
4. [Customer Visit](#)
5. [Customer Group](#)
6. [Product](#)

Sales Representative

Protocols to define how to get, create, update, delete sales representative and how to get his sales orders or customer visits.

Webservice ID	getRep
Webservice Description	Gets all information for given representative
Related Core View(s)	SLSREP_CV
Method	GET
Route	/rep?repId=<repId>
Parameters	+repId - id of given representative
Input Example	<repId> = 654321
Expected Output	<pre>{ "repId": 654321, "fiscalId": 654321456, "name": "Manuel da Rocha Lima", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto", "phone": "+351911991365", "email": "something@else.com", "birthDate": "1986-09-13", "hiredDate": "2015-03-15", "sales": 999, "description": "Good employee. Likes this company", "picture": "path/to/picture.jpg" }</pre>

	}
--	---

Webservice ID	addRep
Webservice Description	Adds a new sales representative
Related Core View(s)	N/A
Method	POST
Route	/rep/add
Parameters	+fiscalId - fiscal id of new representative +name - name of new representative +address - address of new representative ?phone - phone number of new representative ?email - email address of new representative +birthDate - date of birth of new representative +hiredDate - date new representative was hired ?sales - number of sales accomplished by new representative ?description - small description of new representative +picture - path to picture of new representative's face
Input Example	{ "fiscalId": 654321456, "name": "Manuel da Rocha Lima", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto", "birthDate": "1986-09-13", "hiredDate": "2015-03-15", "picture": "path/to/picture.jpg" }
Expected Output	N / A

Webservice ID	updateRep
Webservice Description	Updates information for given representative
Related Core View(s)	N/A
Method	POST
Route	/rep/update

Parameters	+repId - id of given representative ?fiscalId - fiscal id of given representative ?name - name of given representative ?address - address of given representative ?phone - phone number of given representative ?email - email address of given representative ?birthDate - date of birth of given representative ?hiredDate - date given representative was hired ?sales - number of sales accomplished by given representative ?description - small description of given representative ?picture - path to picture of given representative's face
Input Example	<pre>{ "repId": 654321, "fiscalId": 654321456, "name": "Manuel da Rocha Lima", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto", "phone": "+351911991365", "email": "something@else.com", "birthDate": "1986-09-13", "hiredDate": "2015-03-15", "sales": 999, "description": "Good employee. Likes this company", "picture": "path/to/picture.jpg" }</pre>
Expected Output	N / A

Webservice ID	deleteRep
Webservice Description	Deletes given representative
Related Core View(s)	N/A
Method	DELETE
Route	/rep/delete
Parameters	+repId - id of given representative
Input Example	<pre>{ "repId": 654321 }</pre>
Expected Output	N / A

Webservice ID	getRepSalesOrders
Webservice Description	Gets all sales orders for given representative
Related Core View(s)	SLSREP_CV
Method	GET
Route	/sales/rep?repId=<repId>
Parameters	+repId - id of given representative
Input Example	<repId> = 654321
Expected Output	<pre>{ "salesOrders": [{ "salesOrderId": 987654, "customerId": 123456, "repId": 654321, "productId": 789, "quantity": 10, "orderDate": "2017-10-15", "deliveryDate": "2017-11-13", "status": "shipped", "deliveryAddress": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }] }</pre>

Webservice ID	getRepCustomerVisits
Webservice Description	Gets all customer visits for given representative
Related Core View(s)	SLSREP_TSKS_CV
Method	GET
Route	/visits/rep?repId=<repId>
Parameters	+repId - id of given representative
Input Example	<repId> = 654321
Expected Output	<pre>{ "customerVisits": [</pre>

	<pre> { "visitId": 987654, "customerId": 123456, "repId": 654321, "date": "2017-10-15T10:24:00", "description": "It was a good visit.", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }] } </pre>
--	--

Customer

Protocols to define how to get, create, update and delete customer and how to get his sales orders or visits

Webservice ID	getCustomer
Webservice Description	Gets all information for given customer
Related Core View(s)	CLNTINFO_CV
Method	GET
Route	/customer?customerId=<customerId>
Parameters	+customerId - id of given customer
Input Example	<customerId> = 123456
Expected Output	<pre> { "customerId": 123456, "fiscalId": 654321456, "groupId": 987654321, "name": "Manuel da Rocha Lima", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto", "phone": "+351911991365", "email": "something@else.com", "birthDate": "1986-09-13", "status": "active", "orders": 999, "description": "Good and loyal customer.", "picture": "path/to/picture.jpg" } </pre>

Webservice ID	addCustomer
Webservice Description	Adds a new sales customer
Related Core View(s)	N/A
Method	POST
Route	/customer/add
Parameters	+fiscalId - fiscal id of given customer ?groupId - id of given customer's customer group +name - name of given customer +address - address of given customer ?phone - phone number of given customer ?email - email address of given customer ?birthDate - date of birth of given customer +status - status of given customer ?orders - number of orders made by given customer ?description - small description of given customer ?picture - path to picture of given customer's face
Input Example	<pre>{ "fiscalId": 654321456, "name": "Manuel da Rocha Lima", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto", "status": "active" }</pre>
Expected Output	N / A

Webservice ID	updateCustomer
Webservice Description	Updates information for given customer
Related Core View(s)	N/A
Method	POST
Route	/customer/update
Parameters	+customerId - id of given customer ?fiscalId - fiscal id of given customer ?groupId - id of given customer's customer group

	?name - name of given customer ?address - address of given customer ?phone - phone number of given customer ?email - email address of given customer ?birthDate - date of birth of given customer ?status - status of given customer ?orders - number of orders made by given customer ?description - small description of given customer ?picture - path to picture of given customer's face
Input Example	<pre>{ "customerId": 123456, "fiscalId": 654321456, "groupId": 987654321, "name": "Manuel da Rocha Lima", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto", "phone": "+351911991365", "email": "something@else.com", "birthDate": "1986-09-13", "status": "active", "orders": 999, "description": "Good and loyal customer.", "picture": "path/to/picture.jpg" }</pre>
Expected Output	N / A

Webservice ID	deleteCustomer
Webservice Description	Deletes given customer
Related Core View(s)	N/A
Method	DELETE
Route	/customer/delete
Parameters	+customerId - id of given customer
Input Example	<pre>{ "customerId": 123456 }</pre>
Expected Output	N / A

Webservice ID	getCustomerSalesOrders
Webservice Description	Gets all sales orders for given customer
Related Core View(s)	CLNTINFO_CV
Method	GET
Route	/sales/customer?customerId=<customerId>
Parameters	+customerId - id of given customer
Input Example	<customerId> = 123456
Expected Output	<pre>{ "salesOrders": [{ "salesOrderId": 987654, "customerId": 123456, "repId": 654321, "productId": 789, "quantity": 10, "orderDate": "2017-10-15", "deliveryDate": "2017-11-13", "status": "shipped", "deliveryAddress": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }] }</pre>

Webservice ID	getCustomerCustomerVisits
Webservice Description	Gets all customer visits for given customer
Related Core View(s)	SLSREP_TSKS_CV
Method	GET
Route	/visits/customer?customerId=<customerId>
Parameters	+customerId - id of given customer
Input Example	<customerId> = 123456
Expected Output	<pre>{ "customerVisits": [{ "visitId": 987654,</pre>

	<pre> "customerId": 123456, "repId": 654321, "date": "2017-10-15T10:24:00", "description": "It was a good visit.", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }] }</pre>
--	--

Sales Order

Protocols to define how to get, create, update and delete sales order.

Webservice ID	getSalesOrder
Webservice Description	Gets all information for given sales order
Related Core View(s)	N/A
Method	GET
Route	/sales?orderId=<orderId>
Parameters	+orderId - id of given sales order
Input Example	<orderId> = 987654
Expected Output	<pre> { "salesOrderId": 987654, "customerId": 123456, "repId": 654321, "productId": 789, "quantity": 10, "orderDate": "2017-10-15", "deliveryDate": "2017-11-13", "status": "shipped", "deliveryAddress": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }</pre>

Webservice ID	addSalesOrder
Webservice Description	Adds a new entry on the sales orders registry
Related Core View(s)	N/A

Method	POST
Route	/sales/add
Parameters	+customerId - id of the customer that made this order +repId - id of sales representative connected to this order +productId - id of ordered product +quantity - quantity of ordered product ?orderDate - date the order was made +deliveryDate - date of delivery ?status - current status of sales order +deliveryAddress - address to deliver product to
Input Example	<pre>{ "customerId": 123456, "repId": 654321, "productId": 789, "quantity": 10, "deliveryDate": "2017-11-13", "deliveryAddress": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }</pre>
Expected Output	N / A

Webservice ID	updateSalesOrder
Webservice Description	Updates information for given sales order
Related Core View(s)	N/A
Method	POST
Route	/sales/update
Parameters	+orderId - id of given sales order ?customerId - id of the customer that made this order ?repId - id of sales representative connected to this order ?productId - id of ordered product ?quantity - quantity of ordered product ?orderDate - date the order was made ?deliveryDate - date of delivery ?status - current status of sales order ?deliveryAddress - address to deliver product to
Input Example	<pre>{ "salesOrderId": 987654, "customerId": 123456,</pre>

	<pre> "repId": 654321, "productId": 789, "quantity": 10, "orderDate": "2017-10-15", "deliveryDate": "2017-11-13", "status": "shipped", "deliveryAddress": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" </pre>
Expected Output	N / A

Webservice ID	deleteSalesOrder
Webservice Description	Deletes given sales orer
Related Core View(s)	N/A
Method	DELETE
Route	/sales/delete
Parameters	+orderId - id of given sales order
Input Example	<pre> { "orderId": 987654 } </pre>
Expected Output	N / A

Customer Visit

Protocols to define how to get, add, update and delete customer visit.

Webservice ID	getCustomerVisit
Webservice Description	Gets all information for given customer visit
Related Core View(s)	SLSREP_TSKS_CV
Method	GET
Route	/visits?visitId=<visitId>
Parameters	+visitId - id of given visit
Input Example	<visitId> = 987654
Expected Output	<pre>{ "visitId": 987654, "customerId": 123456, "repId": 654321, "date": "2017-10-15T10:24:00", "description": "It was a good visit.", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }</pre>

Webservice ID	addCustomerVisit
Webservice Description	Adds a new entry on the customer visits registry
Related Core View(s)	SLSREP_TSKS_CV
Method	POST
Route	/visits/add
Parameters	+customerId - id of the customer connected to this visit +repId - id of sales representative connected to this visit +date - date of the visit ?description - small description about the visit ?address - address of the visit
Input Example	<pre>{ "customerId": 123456, "repId": 654321, "date": "2017-10-15T10:24:00",</pre>

	<pre>"description": "It was a good visit.", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }</pre>
Expected Output	N / A

Webservice ID	updateCustomerVisit
Webservice Description	Updates the information for given customer visit
Related Core View(s)	SLSREP_TSKS_CV
Method	POST
Route	/visits/update
Parameters	<pre>+visitId - id of given visit ?customerId - id of the customer connected to this visit ?repId - id of sales representative responsible for this visit ?date - date of the visit ?description - small description about the visit ?address - address of the visit</pre>
Input Example	<pre>{ "visitId": 987654, "customerId": 123456, "repId": 654321 "date": "2017-10-15T10:24:00", "description": "It was a good visit.", "address": "Faculdade de Engenharia da Universidade do Porto, 4200-391 Porto" }</pre>
Expected Output	N / A

Webservice ID	deleteCustomerVisit
Webservice Description	Deletes given customer visit
Related Core View(s)	N/A
Method	DELETE

Route	/visits/delete
Parameters	+visitId - id of given visit
Input Example	{ "visitId": 987654 }
Expected Output	N / A

Customer Group

Protocols to define how to get, add, update and delete customer group

Webservice ID	getCustomerGroup
Webservice Description	Gets all information for given customer group
Related Core View(s)	N/A
Method	GET
Route	/group?groupId=<groupId>
Parameters	+groupId - id of given group
Input Example	<groupId> = 123
Expected Output	{ "groupId": 123, "name": "Platinum", "description": "Most loyal customers." }

Webservice ID	addCustomerGroup
Webservice Description	Adds a new customer group
Related Core View(s)	N/A
Method	POST
Route	/group/add
Parameters	+name - name of the customer group +description - small description about this customer group

Input Example	{ "name": "Platinum", "description": "Most loyal customers." }
Expected Output	N / A

Webservice ID	updateCustomerGroup
Webservice Description	Updates information for given customer group
Related Core View(s)	N/A
Method	POST
Route	/group/update
Parameters	+groupId - id of given customer group ?name - name of the customer group ?description - small description about this customer group
Input Example	{ "groupId": 123, "name": "Platinum", "description": "Most loyal customers." }
Expected Output	N / A

Webservice ID	deleteCustomerGroup
Webservice Description	Deletes given customer group
Related Core View(s)	N/A
Method	DELETE
Route	/group/delete
Parameters	+groupId - id of given customer group
Input Example	{ "groupId": 123 }
Expected Output	N / A

Product

Protocols to define how to get, add, update and delete product

Webservice ID	getProduct
Webservice Description	Gets all information for a given product
Related Core View(s)	PINFO_CV
Method	GET
Route	/product?productId=<productId>
Parameters	+productId - id of given product
Input Example	<productId> = 789
Expected Output	<pre>{ "productId": 789, "name": "HP Office Printer Paper", "brand": "HP", "price": 3.20, "VAT": 20,, "size": "A4", "type": "Printer Paper", "itemNumber": 500, "stock": 100 "weight": "2,5 kg", "description": "Suitable for photocopiers, fax machines, inkjet, laser or offset printers" }</pre>

Webservice ID	addProduct
Webservice Description	Adds a new product
Related Core View(s)	PINFO_CV
Method	POST
Route	/product/add
Parameters	+name - name of given product ?brand - brand of given product +price - price of given product +VAT - value added tax on given product

	?size - size of the given product ?type - type of given product +itemNumber - number of items in given product +stock - number of given products available for sale ?weight - weight of given product ?description - small description about given product
Input Example	<pre>{ "name": "HP Office Printer Paper", "brand": "HP", "price": 3.20, "VAT": 20,, "size": "A4", "type": "Printer Paper", "itemNumber": 500, "stock": 100 "weight": "2,5 kg", "description": "Suitable for photocopiers, fax machines, inkjet, laser or offset printers "</pre>
Expected Output	N / A

Webservice ID	updateProduct
Webservice Description	Updates information for given product
Related Core View(s)	PINFO_CV
Method	POST
Route	/product/update
Parameters	+productId - id of given product ?name - name of given product ?brand - brand of given product ?price - price of given product ?VAT - value added tax on given product ?size - size of the given product ?type - type of given product ?itemNumber - number of items in given product ?stock - number of given products available for sale ?weight - weight of given product ?description - small description about given product
Input Example	<pre>{ "productId": 789,</pre>

	<pre> "name": "HP Office Printer Paper", "brand": "HP", "price": 3.20, "VAT": 20,, "size": "A4", "type": "Printer Paper", "itemNumber": 500, "stock": 100 "weight": "2,5 kg", "description": "Suitable for photocopiers, fax machines, inkjet, laser or offset printers " } </pre>
Expected Output	N / A

Webservice ID	deleteProduct
Webservice Description	Deletes given product
Related Core View(s)	N/A
Method	DELTE
Route	/product/delete
Parameters	+productId - id of given product
Input Example	<pre> { "productId": 789 } </pre>
Expected Output	N / A

Path

