Technical Specification

N26 Case Study

Date: 29/08/2020

In this case study, we will show the list of Objects, fields and the Component and when user will select any object it will dynamically show the related product information

**Step 1 Solution:**

Showing product information to customer service agents with the help of LWC component.

To display the customer’s product information, query data from SF database based on case contact information from Home Country and product information so, that agent whilst they’re interacting get related customer product information only.

Modifying to existing product such as special packages, Fees, and any costs information, or adding entirely new products is too easy sales rep can easily create without any Salesforce admin intervene.

Using the values from contact fields to display the matching product information

**Step 2 Solution**: Created a Rest API for external system, where the customer data is shown from Salesforce to get the relevant customer product information.

External system will pass a unique identifier (UUID) that is mapped to contact

record in Salesforce.

Step 1: Create below mentioned data model

Step 2: Create all three Apex Class including test class

[ContactProductTest.cls](https://github.com/varungaur43/CaseStudyN26/blob/master/force-app/main/default/classes/ContactProductTest.cls)

[getContactProduct.cls](https://github.com/varungaur43/CaseStudyN26/blob/master/force-app/main/default/classes/getContactProduct.cls)

[returnContact.cls](https://github.com/varungaur43/CaseStudyN26/blob/master/force-app/main/default/classes/returnContact.cls)

### Step 3: Create the Lightning Component

[lwc/showProduct](https://github.com/varungaur43/CaseStudyN26/tree/master/force-app/main/default/lwc/showProduct) Lightning Components Bundles

Step 4: Add lightning component to recod page

# technical specifications

## Assumptions

In this section, list the assumptions that are made that pertain to the specific technical specification.

1. Need to display customer issue to customer service agents, and to display the customer’s product information while they’re interacting.
2. This Lightning component will display related product information to the service agents.

## Component details

|  |  |  |
| --- | --- | --- |
| **Object Name** | **API Name** | **Features** |
| Case | Case | To display the customer’s issues |
| Contact | Contact | Customer’s Contact Information |
| Product | Product2 | Customer’s Product Information |

## New Fields

All fields needs to be listed for each object

|  |  |  |  |
| --- | --- | --- | --- |
| **Object Name** | **Field Name** | **Data Type** | **Values** |
| Contact | Home Country | Picklist | DE, FR, ES, IT, UK |
|  | Product | Lookup (Product) |  |
|  |  |  |  |
| Product | Name | Text | Product Name |
|  | Product Type | Picklist | Standard, Black, Metal |
|  | Fees | Text | Product Fees |
|  | Home Country | Picklist | DE, FR, ES, IT, UK |

All input and output files should be described and their location identified. Sub-sections that you may want to consider for this section would be listings of database and flat files, and descriptions of parameters received and sent by call programs.

File layouts can also be placed in an appendix.

This section will be where the bulk of the information is given to the developer for the construction of the unit.

* Pseudo code may be used if desired.
* Because it is difficult for specifications to be written with a consistent level of detail, please be explicit.
* If you know details that a developer may not know, include them here to avoid forcing the developer to find that information.
* If you are aware of certain language information such as specific compiler options include those here.
* If you know there are details left out, make note of the fact that the developer will need to find those details.
* Remember to include error processing and report processing where appropriate.