**N26 Case Study**

**Case Study 1:**

There are 2 ways we can achieve this requirement

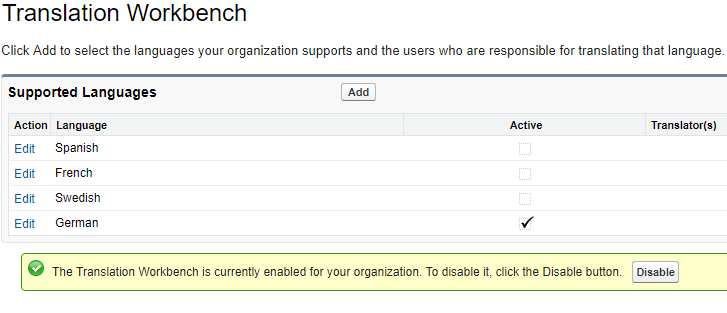
1. Batch class using Visual force template with translate labels.
2. Batch class using email template (based on Contact language will send an email template to customers)

I prefer and gone with 1st way

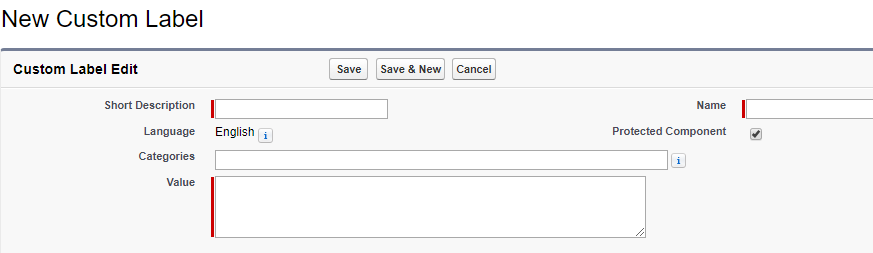
**1. Use Vf template with translate labels:**

**Steps:**

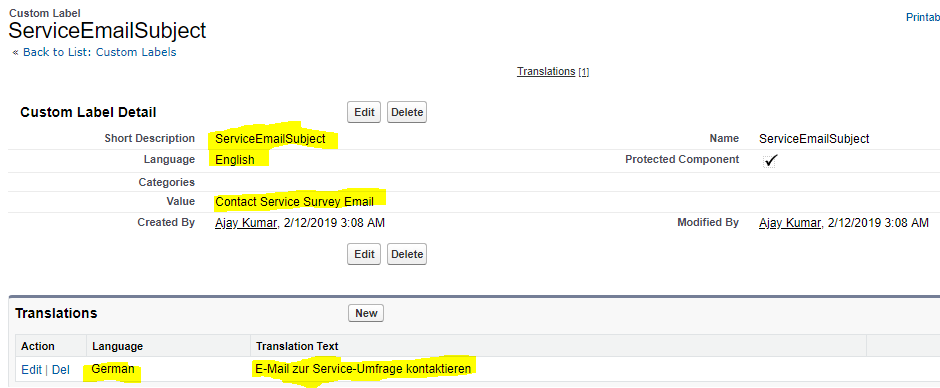
1. Firstly, enable the Translate in Salesforce Org (Note: once translate is enabled we cannot disable it)
2. Add your required language, (for this requirement, I added German in my personal Org)

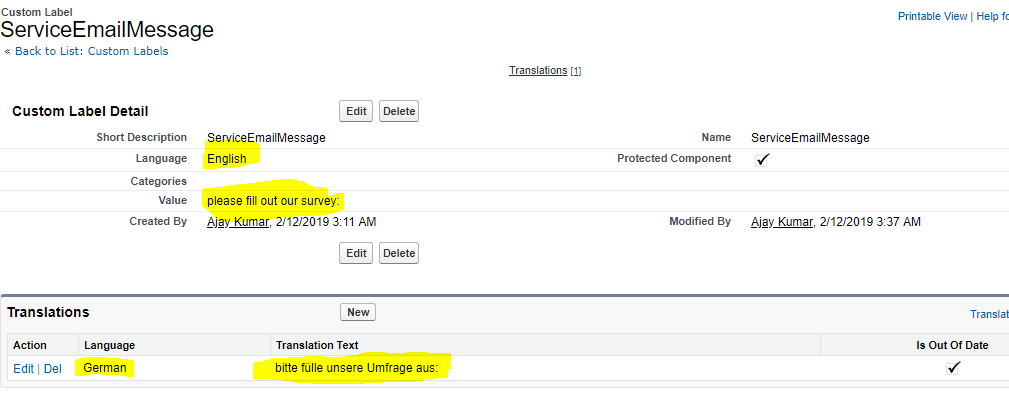


1. Created a custom label one for Subject(ServiceEmailSubject) and another for Body message (ServiceEmailMessage)

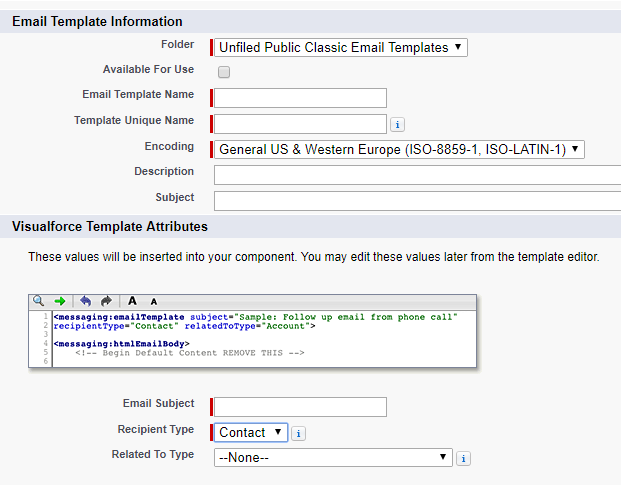


1. Mention the translate of the entered label under Translation section of custom label in your preferred language (created German for this)

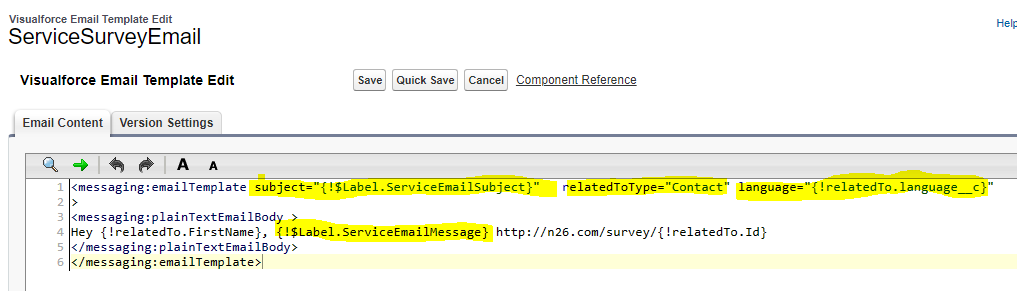




1. Create a Visualforce template with Template Name, Email Subject and Recipient Type (contact for this requirement)



1. Edit the template and make Subject and Body as dynamic by placing the created custom Labels as below



1. Created a Batchable apex(N26\_Contact\_Survey\_Batch) should send out surveys to customers with a survey date before now and on successful send, empty the surveydate filed value
2. Created a test class for batchable apex (N26\_Contact\_Survey\_Test)

**Test:** Have done the Unit testing, working good and meet the requirement.

**Components:**

1. Custom fields on Contact object: survey\_date\_\_c, language\_\_c
2. Custom labels: ServiceEmailMessage, ServiceEmailSubject
3. Visualforce template name: ServiceSurveyEmail
4. Apex batch class: N26\_Contact\_Survey\_Batch
5. Apex batch test class: N26\_Contact\_Survey\_Test

**Case study 2:**

For this requirement we can implement in following ways

1. Use Standard Objects:

* Create product\_\_c lookup field with standard Product object, home\_country\_\_c lookup field with Pricebook2 standard object and all three fields (Cost per month, ATM fee and Card Replacement) on standard PricebookEntry object.
* Create a lightning component to get Products Information (three field values) based on Case contacts product and homecountry values.

1. Use custom objects:

* Create product\_\_c lookup field with standard Product object, home\_country\_\_c as a picklist or lookup field with custom object and create custom object with the combination of Prdocut2 + home country values, create all three fields (Cost per month, ATM fee and Card Replacement) on the same custom junction object.
* Create a lightning component to get Products Information (three field values) based on Case contacts product and homecountry values

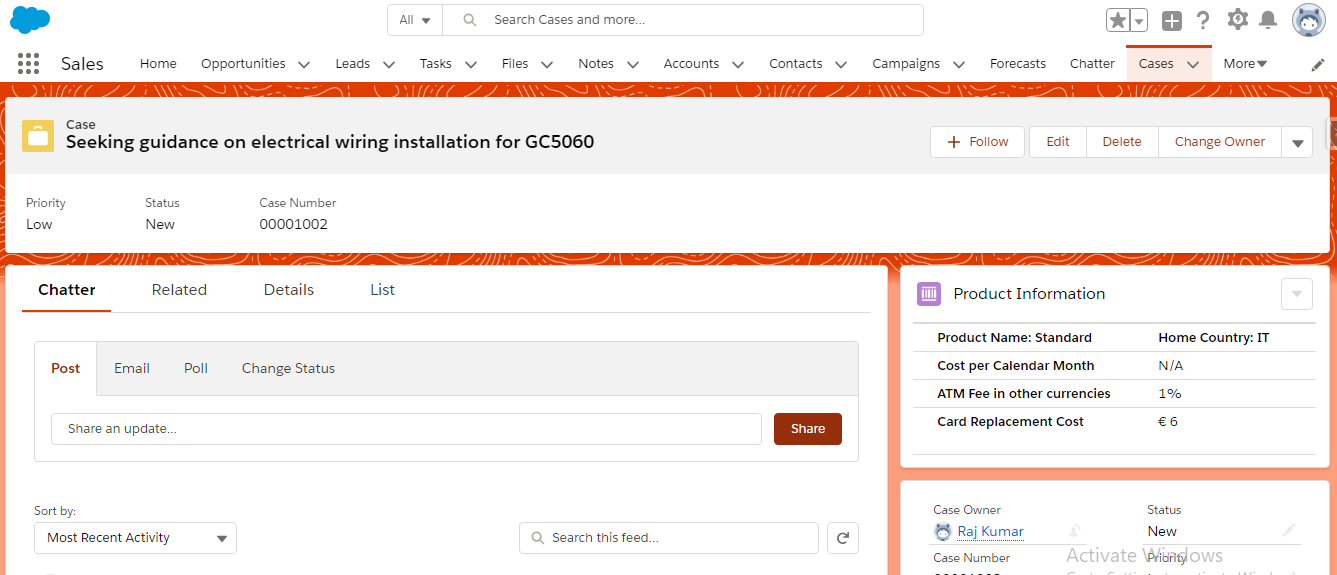
**Note:** In future if we require any values as mentioned in case study, we can create them as a fields.

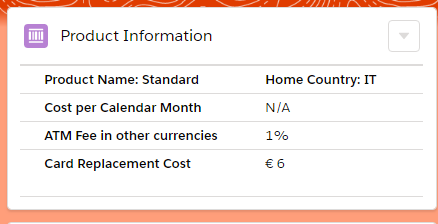
I prefer and gone with 1st way as utilizing the standard objects.

1. **Use Standard Objects:**

**Steps:**

* Create Product\_\_c lookup field with standard Product object.
* Create Home\_Country\_\_c lookup field with standard Pricebook2 object.
* Create three new fields (Cost per month, ATM fee and Card Replacement) on standard PricebookEntry object
* Create a Lightning component to get info based on Case Contacts product and homelocation values
* Add the component on Case layout using Lighting App Builder.





**Test:** Done unit testing working fine as expected

**Components:**

1. **Custom fields on Contact:** Home\_Country\_\_c, Product\_\_c
2. **Custom fields on Pricebook Entry:** Cost\_per\_Calendar\_Month\_\_c, ATM\_Fee\_in\_other\_currencies\_\_c, Card\_Replacement\_Cost\_\_c
3. **Lightning component name:** N26\_ProductInformation
4. **Apex Class:** N26\_ProductInformation
5. Test class: N26\_ProductInformation\_Test