SAP Fiori UX Design and Build Challenge – Sales Rep app

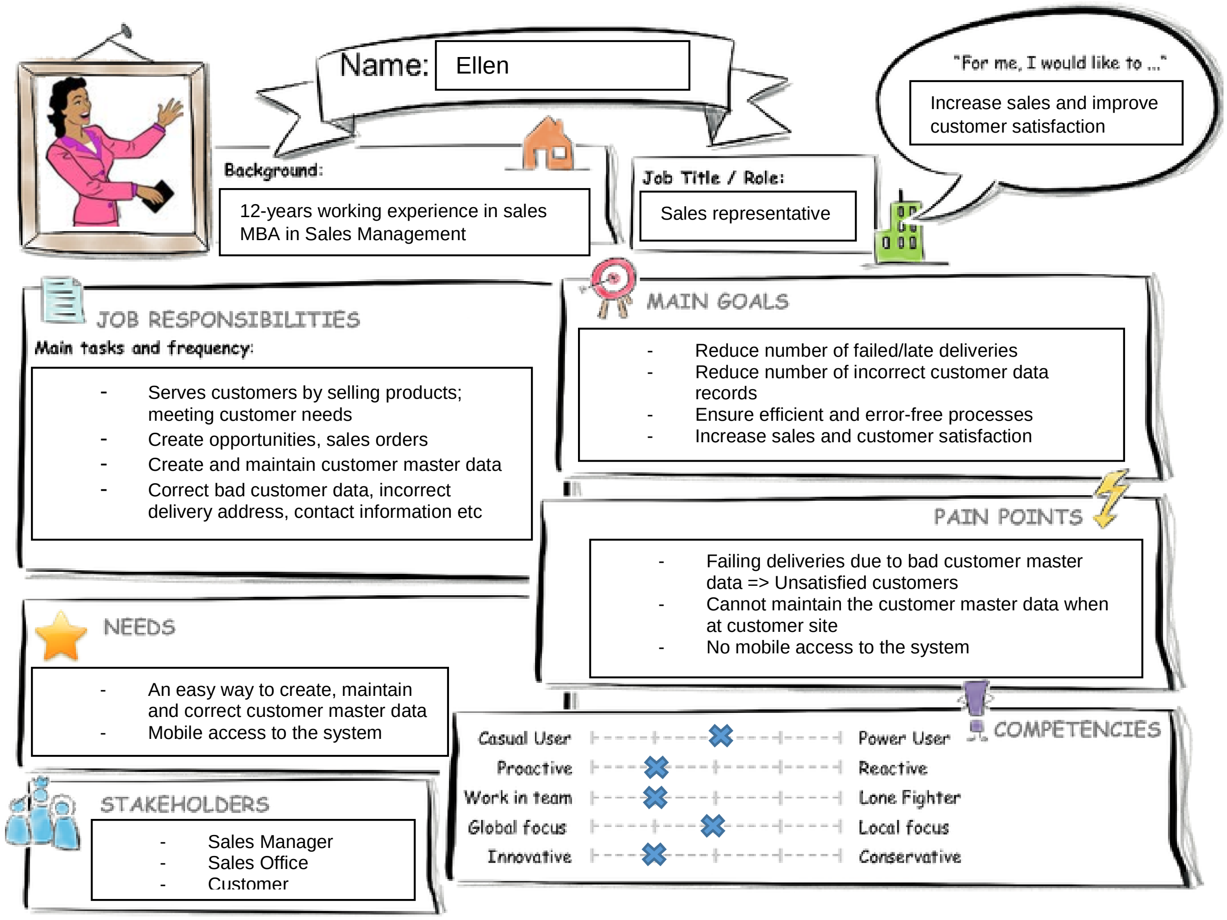
Story – How to make the life easier for a sales representative

The working life for a sales representative consists of customer visits, creating customer orders and maintaining the customer information. The main goal is to increase sales and keep the customers happy.

The sales representative today uses a laptop to look-up customer information and creating sales order. There is no way to maintain the customer information directly, the sales representative instead needs to call the sales office or send an email with the information that has to be corrected. This results in that some updates are forgotten, are entered late or are entered incorrectly into the system. In worst case it leads to incorrect deliveries or late deliveries when for example the delivery address is not correct.

It would be much better if the changes could be done directly when at customer site. The job would also be a lot more efficient if the data could be entered in a mobile app instead of a laptop. Overall the efficiency improvements would give more time with the customers or make it possible to do more customer visits per day, thus increasing the sales opportunities and would also improve the customer satisfaction.

Persona



Name: Ellen

"For me, I would like to ..."

Increase sales and improve customer satisfaction

Background:

12-years working experience in sales
MBA in Sales Management

Job Title / Role:

Sales representative

JOB RESPONSIBILITIES

Main tasks and frequency:

- Serves customers by selling products; meeting customer needs
- Create opportunities, sales orders
- Create and maintain customer master data
- Correct bad customer data, incorrect delivery address, contact information etc

MAIN GOALS

- Reduce number of failed/late deliveries
- Reduce number of incorrect customer data records
- Ensure efficient and error-free processes
- Increase sales and customer satisfaction

PAIN POINTS

- Failing deliveries due to bad customer master data => Unsatisfied customers
- Cannot maintain the customer master data when at customer site
- No mobile access to the system

NEEDS

- An easy way to create, maintain and correct customer master data
- Mobile access to the system

COMPETENCIES

Casual User			✖			Power User
Proactive		✖				Reactive
Work in team		✖				Lone Fighter
Global focus			✖			Local focus
Innovative		✖				Conservative

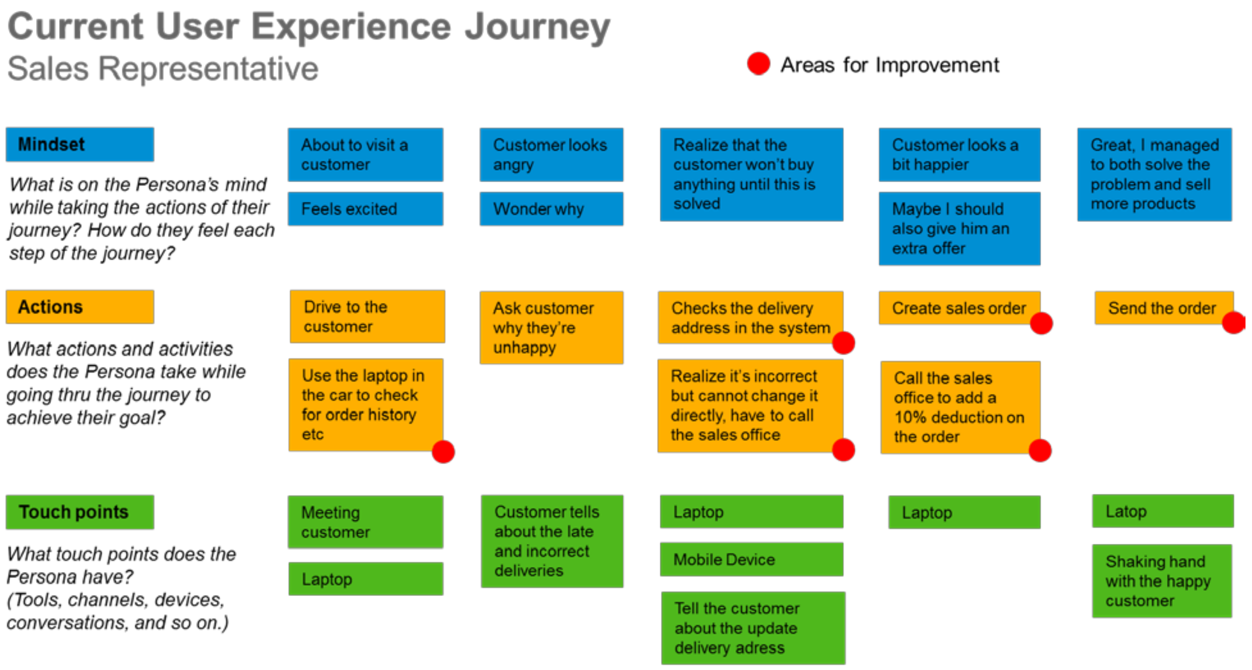
STAKEHOLDERS

- Sales Manager
- Sales Office
- Customer

Current User Experiene

The picture below describes the user experience for a sales rep. The mindset during the journey, the actions points when meeting the customer and the different touch points.

The areas of improvement has been identified with red dots, it’s areas where the user has expressed pains today that causes frustration, inefficiency and sometimes even affecting the business.



Current User Experience Journey
Sales Representative

● Areas for Improvement

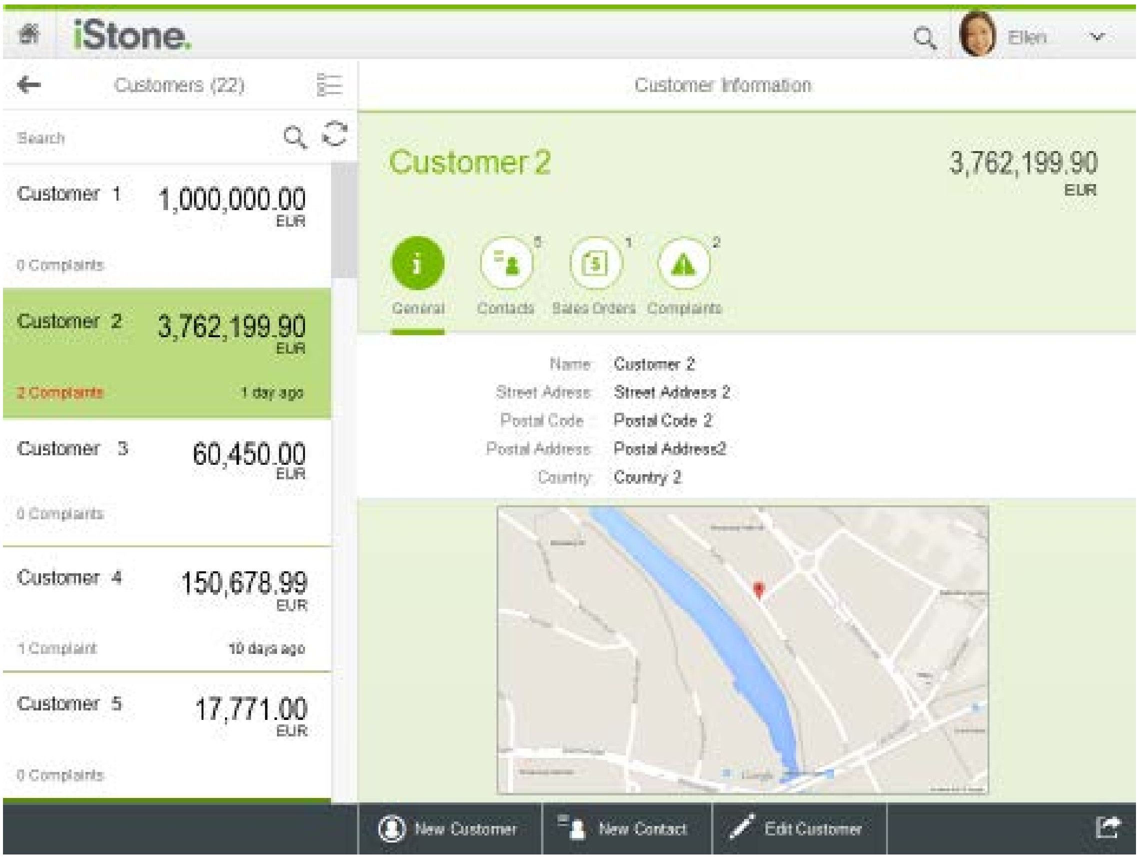
Mindset *What is on the Persona’s mind while taking the actions of their journey? How do they feel each step of the journey?*	About to visit a customer Feels excited	Customer looks angry Wonder why	Realize that the customer won’t buy anything until this is solved	Customer looks a bit happier Maybe I should also give him an extra offer	Great, I managed to both solve the problem and sell more products
Actions *What actions and activities does the Persona take while going thru the journey to achieve their goal?*	Drive to the customer Use the laptop in the car to check for order history etc ●	Ask customer why they’re unhappy	Checks the delivery address in the system ● Realize it’s incorrect but cannot change it directly, have to call the sales office ●	Create sales order ● Call the sales office to add a 10% deduction on the order ●	Send the order ●
Touch points *What touch points does the Persona have? (Tools, channels, devices, conversations, and so on.)*	Meeting customer Laptop	Customer tells about the late and incorrect deliveries	Laptop Mobile Device Tell the customer about the update delivery adress	Laptop	Latop Shaking hand with the happy customer

Mock-up of the sales rep app

The focus of the Sales rep app is to make the life easier for the Sales representatives and the initial focus has been to solve the pains around maintaining and updating customer information and enabling mobile access to the system.

The mock-up shows a prototype of how the new Sales rep application would look like based on the SAP Fiori Design Guidelines and the SAP Fiori Prototyping Kit.

The first picture show the starting page of the application, it shows the sales rep’s list of customers with information on sales revenue and number of complaints. She can also search among the customers. When clicking on a customer in the list to the left, detailed information is displayed to the right. The first tab contains general customer information, such as address and a map showing the location of the customer. The second tab shows the customer contacts, the third shows the sales orders for the customer and the forth tab shows the most recent complaints. It’s planned later to add more information in other tabs to get a complete view of the customer. At the bottom of the page she can click on three different buttons to either create a new customer, create a new contact or edit an existing customer.



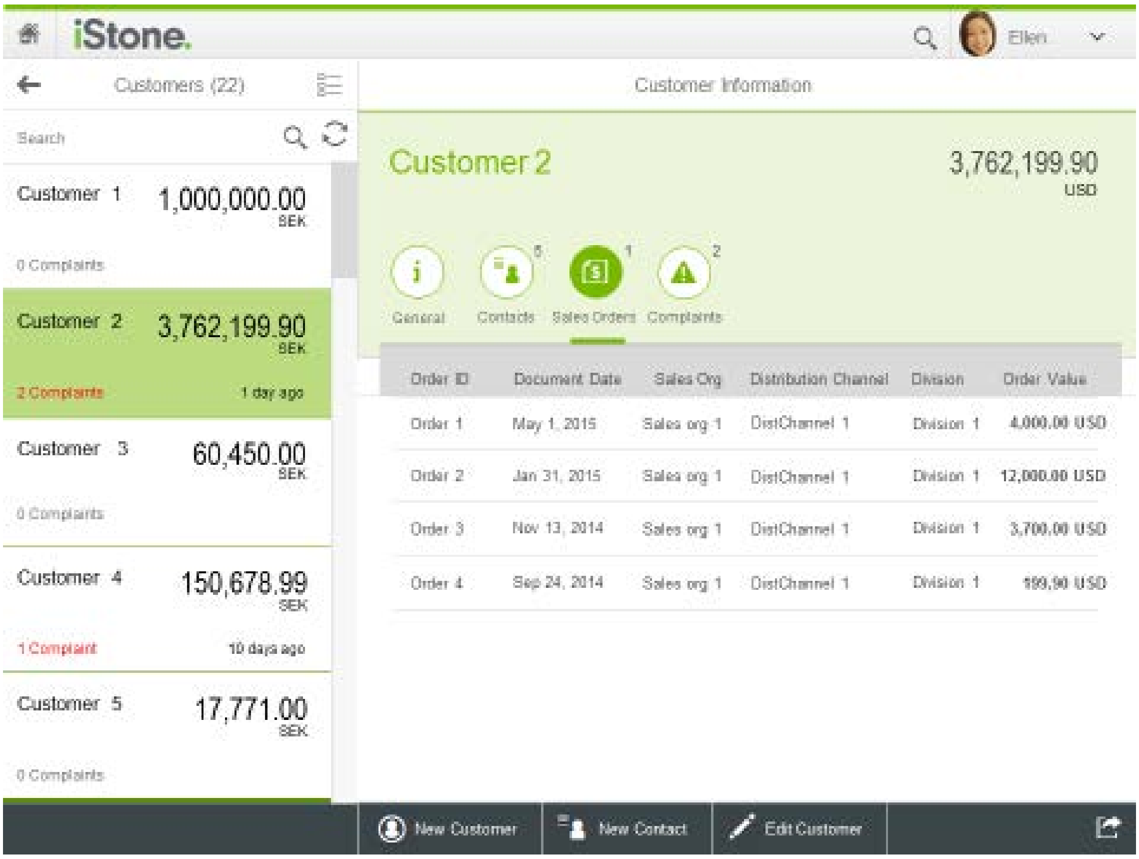
Customer 1	1,000,000.00 EUR	
0 Complaints		
Customer 2	3,762,199.90 EUR	
2 Complaints		1 day ago
Customer 3	60,450.00 EUR	
0 Complaints		
Customer 4	150,678.99 EUR	
1 Complaint		10 days ago
Customer 5	17,771.00 EUR	
0 Complaints		

Customer 2

3,762,199.90
EUR

Name	Customer 2
Street Adress	Street Address 2
Postal Code	Postal Code 2
Postal Address	Postal Address2
Country	Country 2

Mockup Picture 1. Showing a list of customers and detailed customer information for Customer 2.



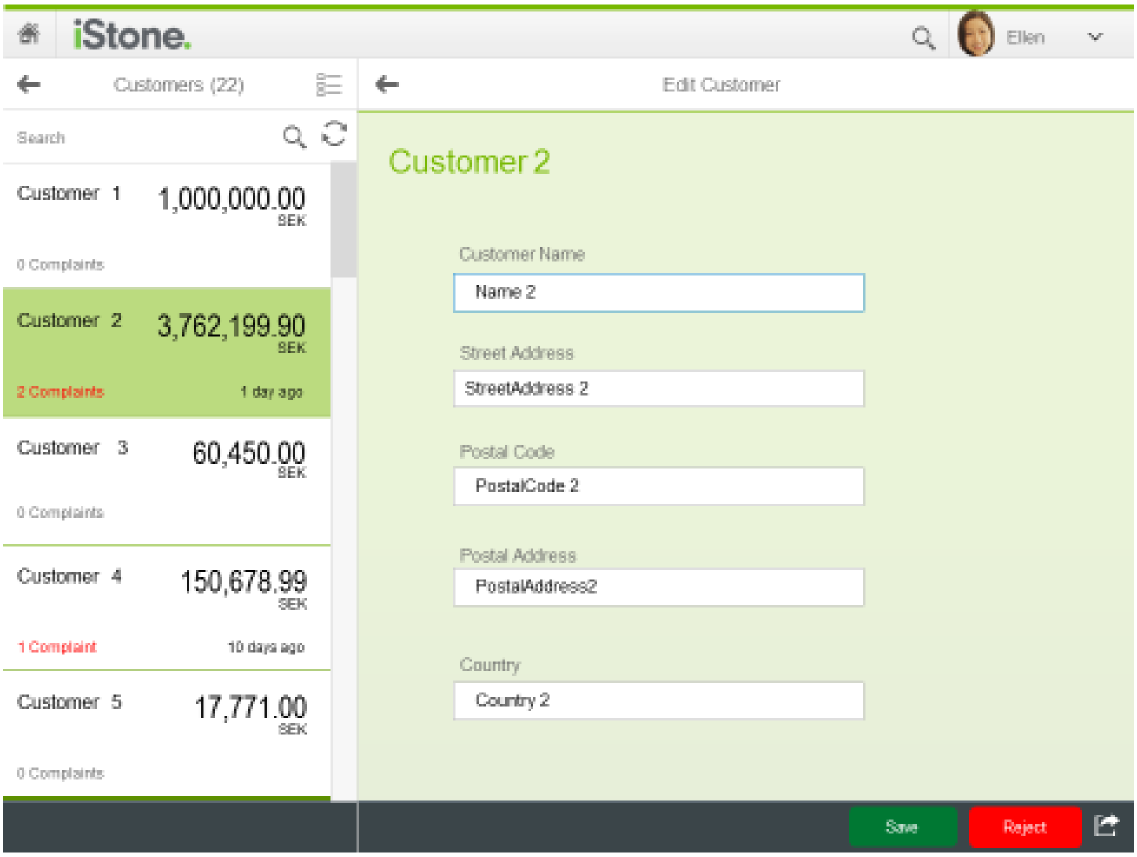
Customer 1	1,000,000.00 SEK	
0 Complaints		
Customer 2	3,762,199.90 SEK	
2 Complaints		1 day ago
Customer 3	60,450.00 SEK	
0 Complaints		
Customer 4	150,678.99 SEK	
1 Complaint		10 days ago
Customer 5	17,771.00 SEK	
0 Complaints		

Customer 2

3,762,199.90
USD

Order ID	Document Date	Sales Org	Distribution Channel	Division	Order Value
Order 1	May 1, 2015	Sales org 1	DistChannel 1	Division 1	4,000.00 USD
Order 2	Jan 31, 2015	Sales org 1	DistChannel 1	Division 1	12,000.00 USD
Order 3	Nov 13, 2014	Sales org 1	DistChannel 1	Division 1	3,700.00 USD
Order 4	Sep 24, 2014	Sales org 1	DistChannel 1	Division 1	199.90 USD

Mockup Picture 2. Showing Sales Order information for Customer 2.



Customer 1 1,000,000.00 SEK
0 Complaints

Customer 2 3,762,199.90 SEK
2 Complaints 1 day ago

Customer 3 60,450.00 SEK
0 Complaints

Customer 4 150,678.99 SEK
1 Complaint 10 days ago

Customer 5 17,771.00 SEK
0 Complaints

Customer 2

Customer Name: Name 2

Street Address: StreetAddress 2

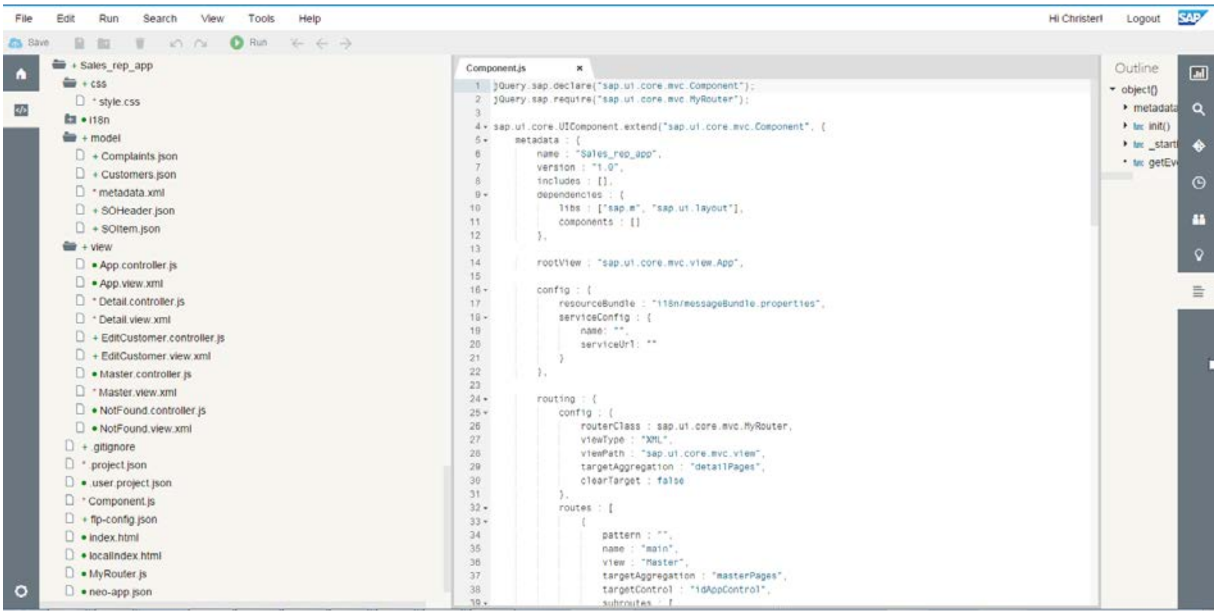
Postal Code: PostalCode 2

Postal Address: PostalAddress2

Country: Country 2

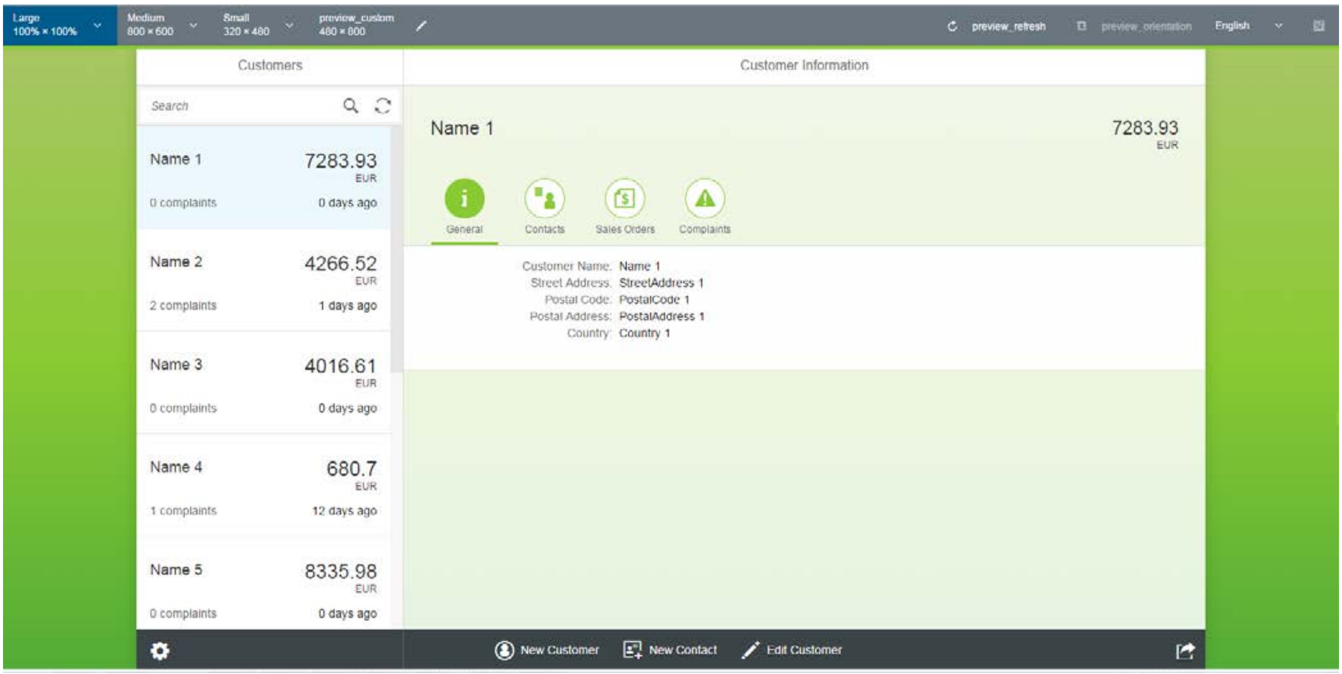
Mockup Picture 4. Showing the Edit customer screen after clicking the Edit Customer button on previous screen. The input fields are filled with the current values and they can then be changed by the sales rep, making corrections directly in the system.

Sales Rep app prototype – Web IDE screenshots



```
jQuery.sap.declare("sap.ui.core.mvc.Component");
jQuery.sap.require("sap.ui.core.mvc.MyRouter");

sap.ui.core.UIComponent.extend("sap.ui.core.mvc.Component", {
    metadata : {
        name : "Sales_rep_app",
        version : "1.0",
        includes : [],
        dependencies : {
            libs : ["sap.m", "sap.ui.layout"],
            components : []
        },

        rootView : "sap.ui.core.mvc.view.App",

        config : {
            resourceBundle : "i18n/messageBundle.properties",
            serviceConfig : {
                name: "",
                serviceUrl: ""
            }
        },

        routing : {
            config : {
                routerClass : sap.ui.core.mvc.MyRouter,
                viewType : "XML",
                viewPath : "sap.ui.core.mvc.view",
                targetAggregation : "detailPages",
                clearTarget : false
            },
            routes : [
                {
                    pattern : "",
                    name : "main",
                    view : "Master",
                    targetAggregation : "masterPages",
                    targetControl : "idAppControl",
                    subroutes : [
```

Web IDE Screenshot 1. This shows the content of the Sales_rep_app project that has been created in Web IDE based on the Fiori Master Detail template. The application has three views; Master view, Detail View and Edit Customer View. The application has been branded by first creating a Theme in the UI Theme Designer and then *taking the generated css code and copied it into the css file for the app. In a production scenario this would be done by choosing the Theme for the Fiori Launchpd directly, but as all the screenshot below shows examples with mock data I wanted to use this way to get a branded application even in the preview in Web IDE.*

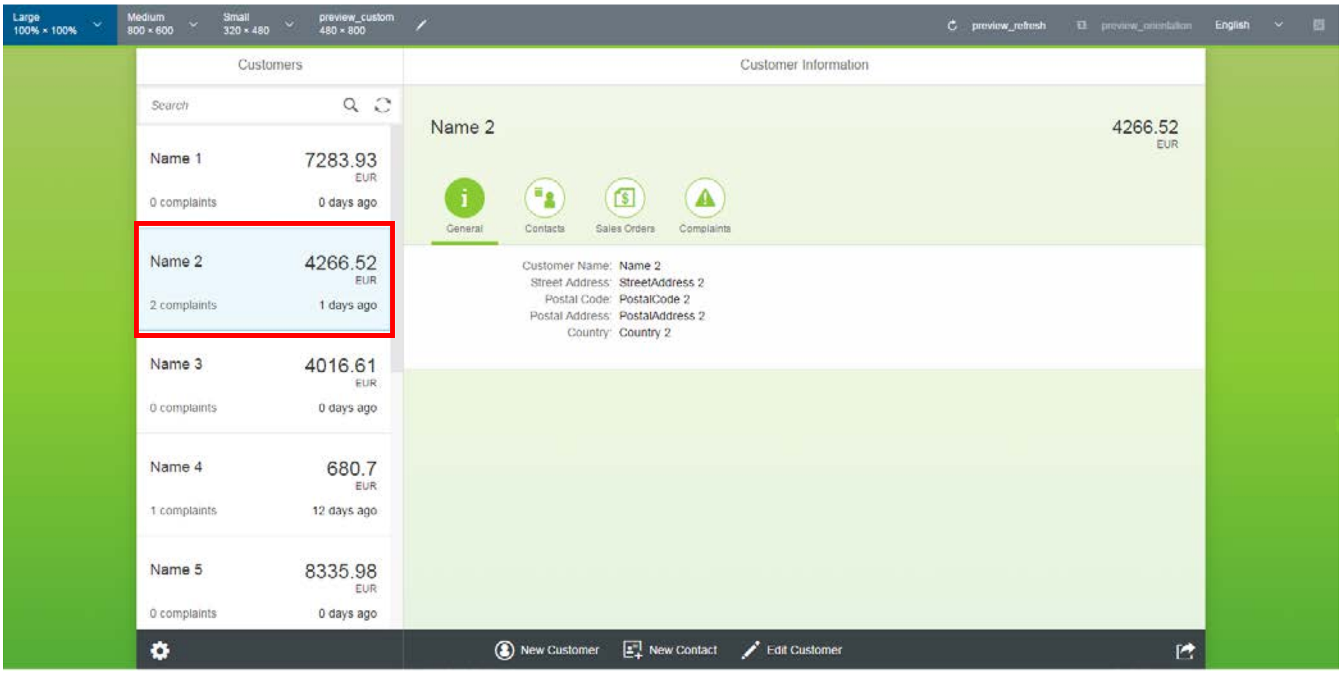


Name 1	7283.93 EUR
0 complaints	0 days ago
Name 2	4266.52 EUR
2 complaints	1 days ago
Name 3	4016.61 EUR
0 complaints	0 days ago
Name 4	680.7 EUR
1 complaints	12 days ago
Name 5	8335.98 EUR
0 complaints	0 days ago

Name 1 7283.93 EUR

Customer Name: Name 1
Street Address: StreetAddress 1
Postal Code: PostalCode 1
Postal Address: PostalAddress 1
Country: Country 1

Web IDE Screenshot 2. This shows the preview of the application when run with mock data. It shows a list of customers in the Master view and detailed customer information in the Detail view using a simple form. The Detail view has 4 tabs, General, Contacts, Sales Orders and Complaints. The Footer has 3 action buttons; New Customer, New Contact and Edit Customer.

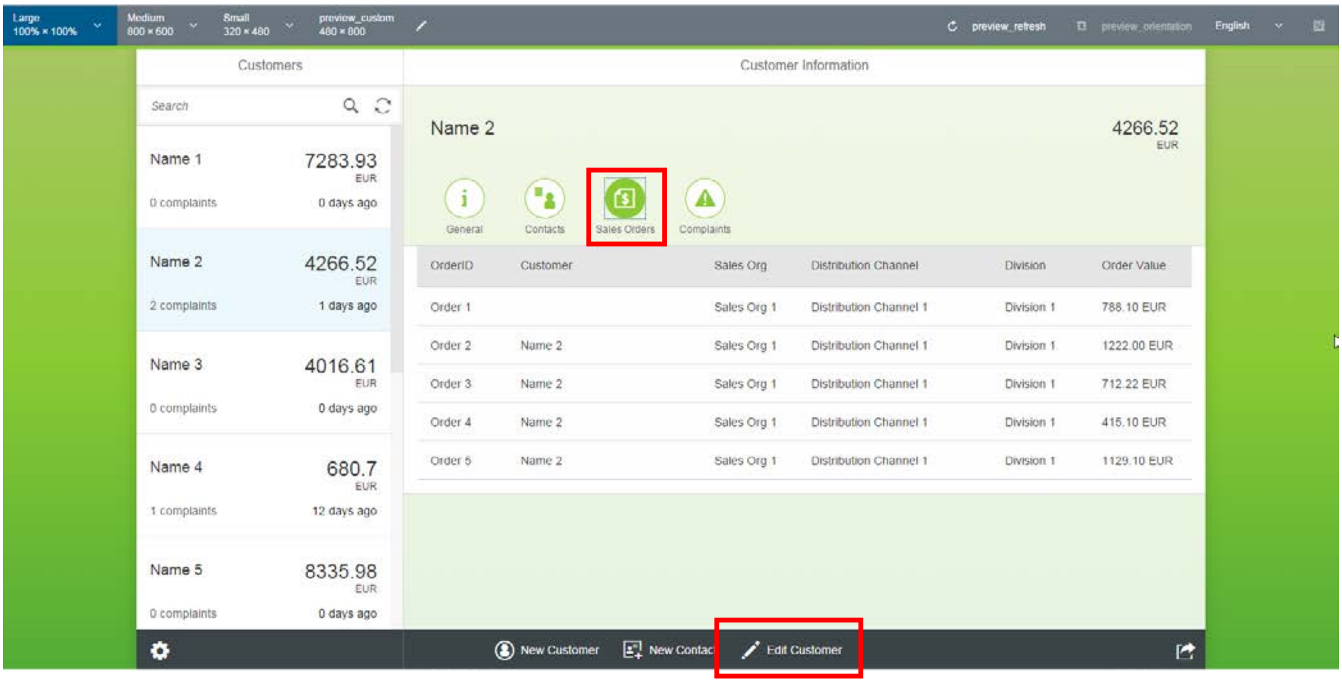


Name 1	7283.93 EUR
0 complaints	0 days ago
Name 2	4266.52 EUR
2 complaints	1 days ago
Name 3	4016.61 EUR
0 complaints	0 days ago
Name 4	680.7 EUR
1 complaints	12 days ago
Name 5	8335.98 EUR
0 complaints	0 days ago

Name 2 4266.52 EUR

Customer Name: Name 2
Street Address: StreetAddress 2
Postal Code: PostalCode 2
Postal Address: PostalAddress 2
Country: Country 2

Web IDE Screenshot 3. This shows that the detailed customer information to the right changes when clicking on another customer in the customer list to the left.



Customers

Search

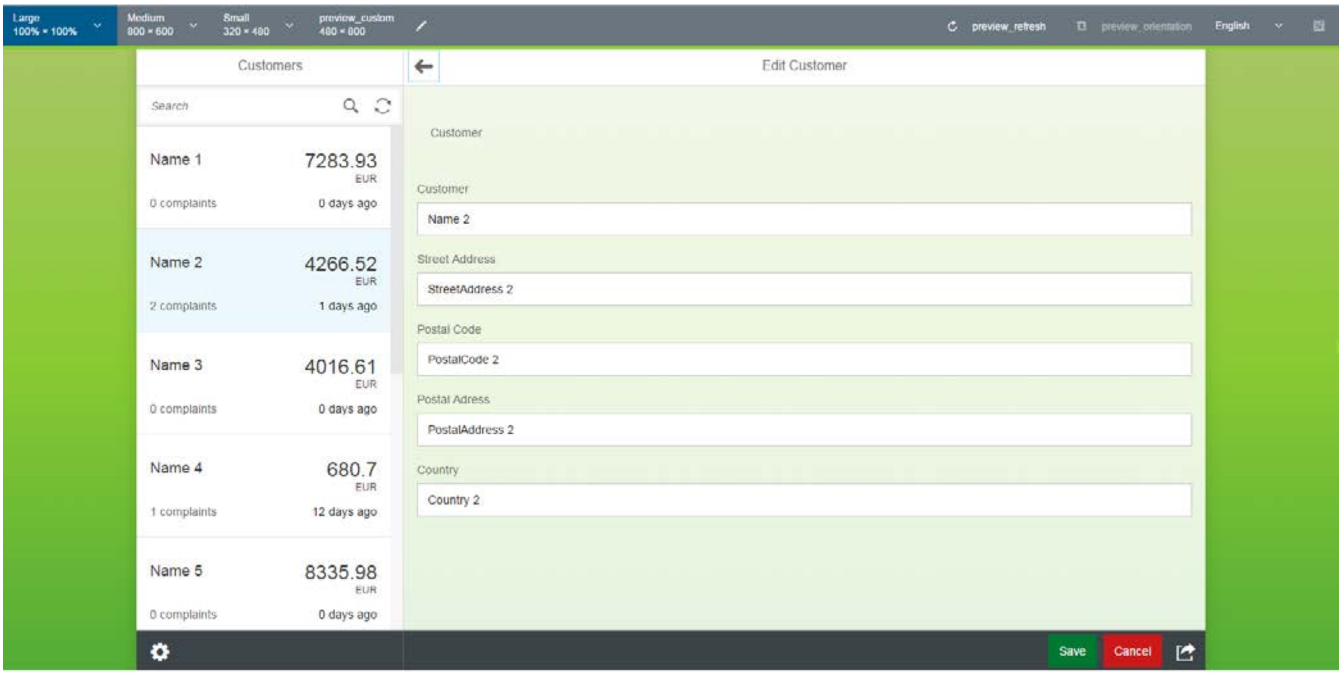
Name 1	7283.93 EUR	
0 complaints	0 days ago	
Name 2	4266.52 EUR	
2 complaints	1 days ago	
Name 3	4016.61 EUR	
0 complaints	0 days ago	
Name 4	680.7 EUR	
1 complaints	12 days ago	
Name 5	8335.98 EUR	
0 complaints	0 days ago	

Customer Information

Name 2 4266.52 EUR

OrderID	Customer	Sales Org	Distribution Channel	Division	Order Value
Order 1		Sales Org 1	Distribution Channel 1	Division 1	788.10 EUR
Order 2	Name 2	Sales Org 1	Distribution Channel 1	Division 1	1222.00 EUR
Order 3	Name 2	Sales Org 1	Distribution Channel 1	Division 1	712.22 EUR
Order 4	Name 2	Sales Org 1	Distribution Channel 1	Division 1	415.10 EUR
Order 5	Name 2	Sales Org 1	Distribution Channel 1	Division 1	1129.10 EUR

Web IDE Screenshot 4. This shows the Sales Order tab, which displays a list of sales order for the selected customer by using a Table component. As the next step the user clicks on the “Edit Customer”-button to maintain the customer information, this will open the “EditCustomer” view below.



Customers

Search

Name 1	7283.93 EUR
0 complaints	0 days ago
Name 2	4266.52 EUR
2 complaints	1 days ago
Name 3	4016.61 EUR
0 complaints	0 days ago
Name 4	680.7 EUR
1 complaints	12 days ago
Name 5	8335.98 EUR
0 complaints	0 days ago

Edit Customer

Customer

Customer: Name 2

Street Address: StreetAddress 2

Postal Code: PostalCode 2

Postal Adress: PostalAddress 2

Country: Country 2

Web IDE Screenshot 5. This shows the EditCustomer view after clicking the “Edit Customer”-button. The input fields are filled with the current values for the selected customer and they can then be changed by the sales rep, making corrections directly in the system. The footer contains two action buttons; a Save button (type=”Accept”) and a Cancel button (Type=”Reject”). The view also has a Back-button in the header to navigate back to the previous screen.