Portafix

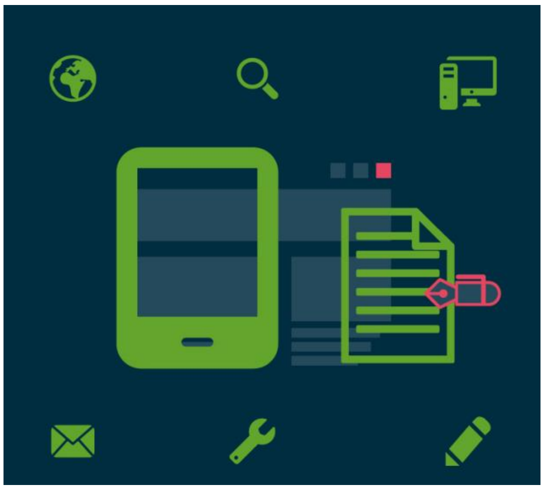
Story

There is a group of technicians / qualified operators who work doing schedule maintenance, general fixes, calibration or refurbishments. They receive their orders / notifications in a daily basis where they work. This information is present within a SAP PM system, but they are not interacting with that system. For that reason, they have to go every time to the office in order to get the new activity plan (in paper) from their coordinator and returning for the customer paperwork when the activity is completed. The company provide them with different tools for facilitating their operations, like a truck, a phone they can use to call directly to a hot line for any inconvenience or update to/from the company, a GPS (to check where and how to travel to the customer) and tools / instructions for the required material during the activity.

Reason:

The need for an app like Portafix is that the technicians / operators are missing a lot of time, re-scheduling, calculating and travel without any help from a direct device in a real/offline scenario. This is because they are not interacting with the existing SAP PM system that they have, but only providing input and paperwork.
With Portafix, the operators can decide which Maintenance plan is more suitable for them based on their location (where they are and where they need to be), they can check which equipment is needed, the specifications and check how they should look like, in other words, saving a lot of time and costs with the use of a friendly/ precise app that will prevent many errors and help them in their operations. They will access the SAP system from a Fiori App without having any special training or calling some other employee for doing it.

Scope / Extensibility:



The app proposed in this documentation is a simple case, but it could cover most of the SAP PM scenarios with simple extensions. It would better work with google maps APIs , cordova methods (GPS, scan of barcodes and pushing notifications), however due to the deadlines and limitations this is presented as a prototype and in a mock-up format. It can be easily adapted for an IS-OIL industry and connected through ODATA services with Ariva in cases that the equipment need to be acquired. These factors could represent an interesting advantage in the actual market.

Name: Juan

Background:
33 years old, married. 10y exp.

Job Title / Role:
Technician

"For me, I would like to ..."
Improve efficiency and delay time in my daily activities

JOB RESPONSIBILITIES
Main tasks and frequency:

-Daily get activity plan, equipment information, customer location, customer priorities.
- Printing orders, getting forms for acceptance / rejections.
-Perform repairs, calibrations, orders,etc.

MAIN GOALS

-Repair/fix/Replace/calibrate equipment from a notification list.
-More fixes, more bonus income.

PAIN POINTS

- Travel from my location to the office and then to a customer close to where i live.
- Papers , forms, getting signs, asking people for that.
- No access to the system, no realtime information

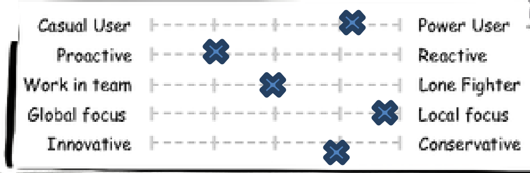
NEEDS

-Reduce paper work and access to the information req. for my activities

STAKEHOLDERS

-Operations Manager
-Plan Coordinator

COMPETENCIES



Current User Experience Journey

Duration of the Journey: 1-8 hs

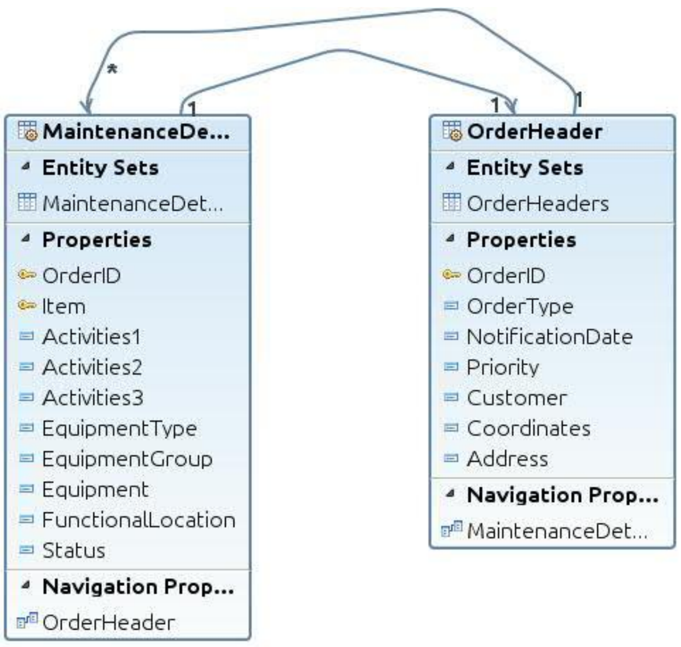
- Again, need to go to the office instead of leaving directly from 🔴
- Check my activities for today
- Normal Maintenance and some 🔵
- Check what I can repair / 🔵
- Estimating Activity
- Estimating Travel 🔴
- Build my activity
- Shall I check traffic / travel alternatives? 🔴
- Get Equipment 🔴
- Get the paperw 🔴
- Load Equipment
- Ready to 🔵

- Go to the first
- Read / Check Equipment 🔴
- Check that I have everything
- Do the Operations / Fix / 🔵
- Notify the planned Activity the customer
- Check everything is working as expected
- Go to the Function
- Get the Sign-off and 🔴
- Check all equipment (on 🔵
- Go to next customer or back

- Repair tools
- Repair / Maintenance protocols
- Colleague /
- Company Truck
- Company Phone
- Company Hot Line
- Mobile Device
- GP
- Loading / Unloading area
- Documents (PM order,
- Pe

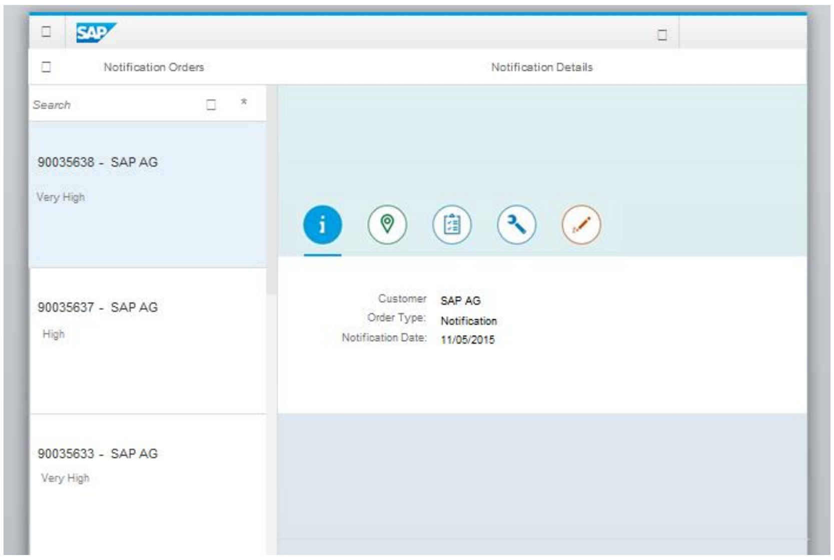
Prototype:

Metadata:



Fiori Prototype:

1. Prototype for Main Screen



90035638 - SAP AG
Very High

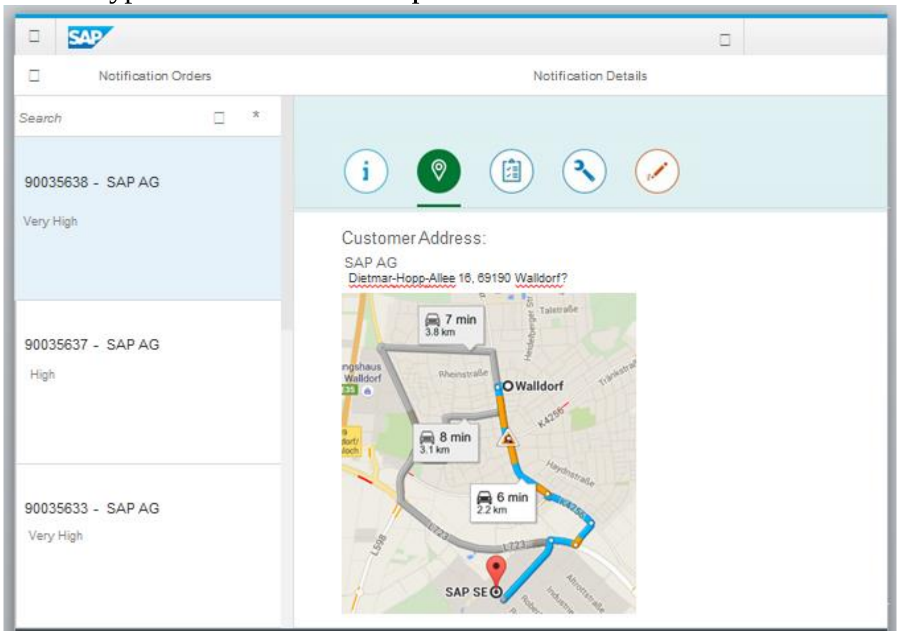
90035637 - SAP AG
High

90035633 - SAP AG
Very High

Customer: SAP AG
Order Type: Notification
Notification Date: 11/05/2015

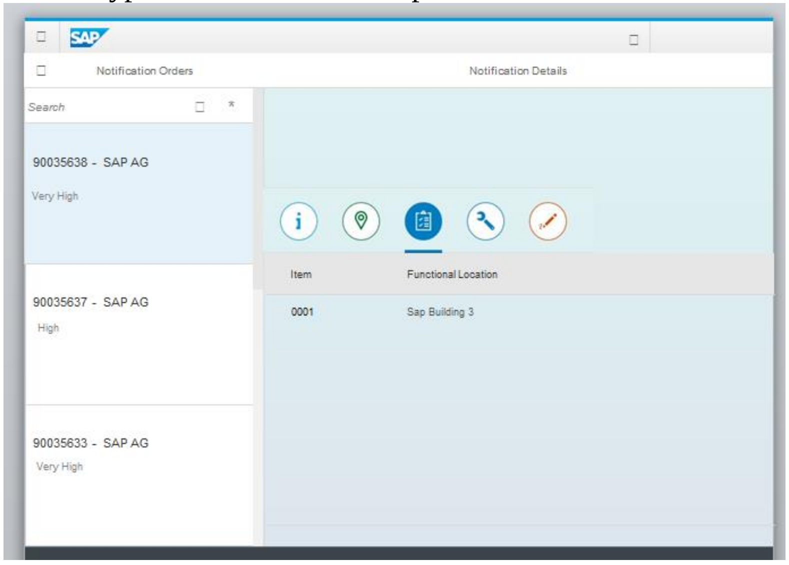
The App is a Detail Master list style (creating from the Fiori Master List template), very simple an most of the operations are sorted in each Icon Filter with different components.

2-Prototype for Activities to be performed



Because of the limitations for this Challenge the google maps object is going to be replaced with an URL with the corresponding customer coordinates

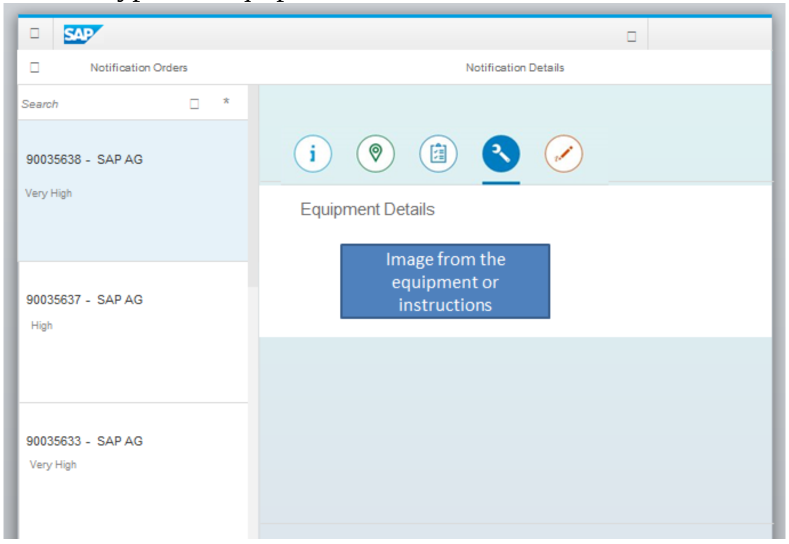
3-Prototype for Activities to be performed



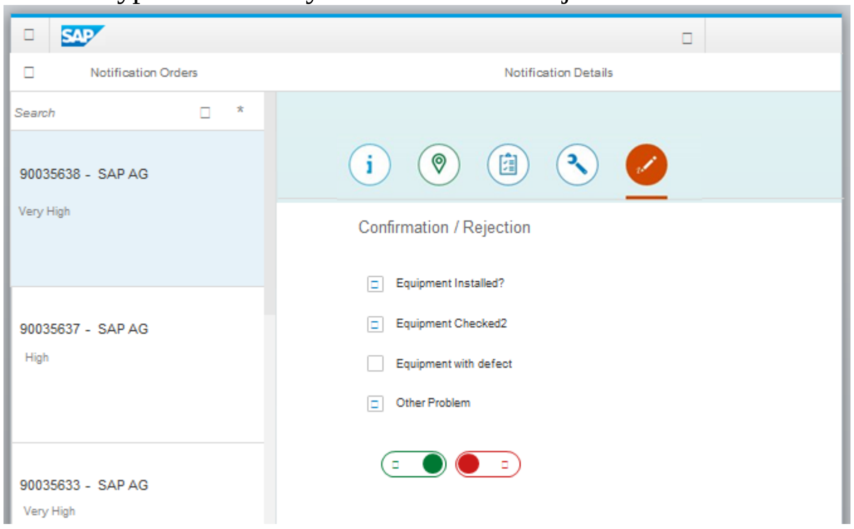
Item	Functional Location
0001	Sap Building 3

Using a table list , we show the first line of the header and then the items in the subsequent navigation. (equipments)

4- Prototype for Equipment details

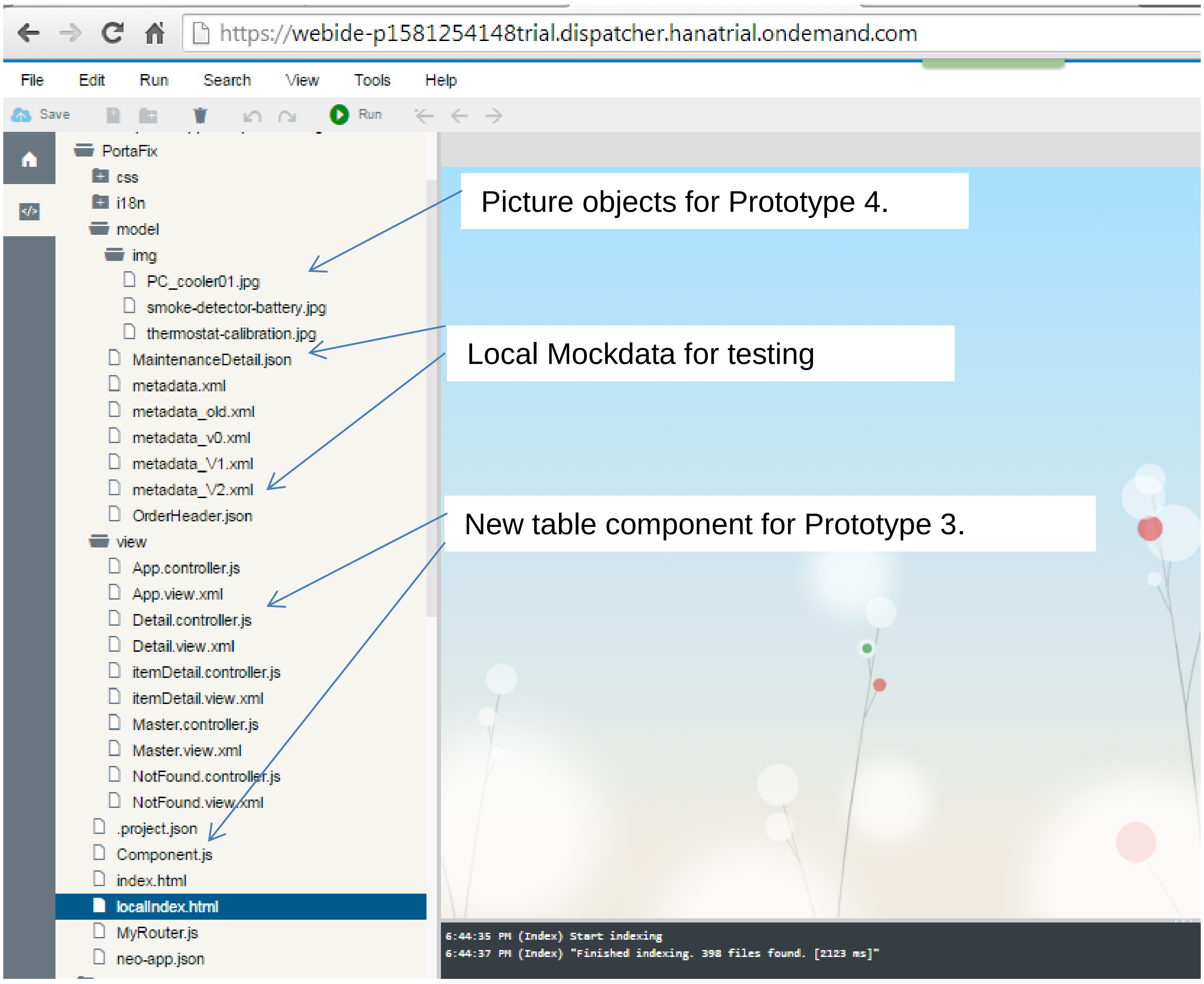


5- Prototype for Activity Confirmation / Rejection

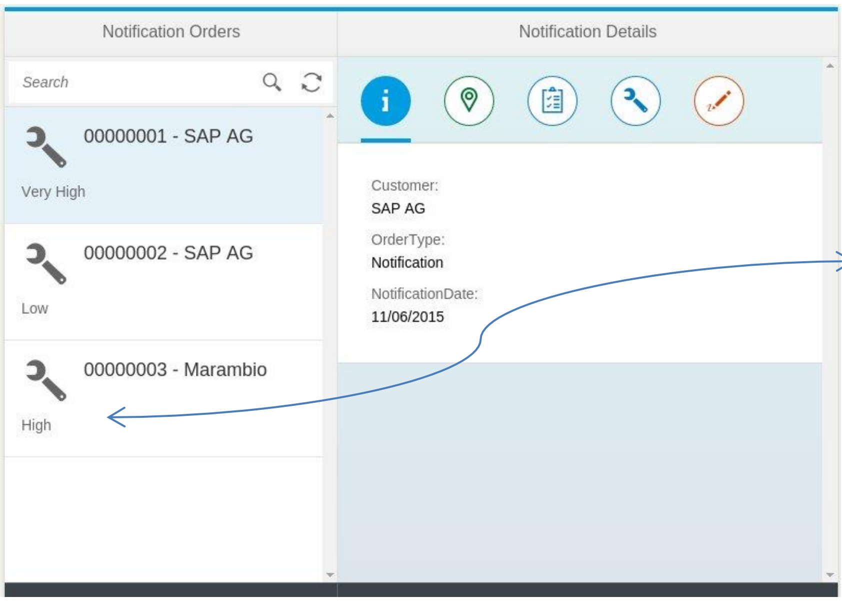


Confirmation / Rejection is contained inside a SimpleForm with the following objects:
Checkboxes, Switch (reject type)

SAP WEBIDE Build:



```
6:44:35 PM (Index) Start indexing
6:44:37 PM (Index) "Finished indexing. 398 files found. [2123 ms]"
```



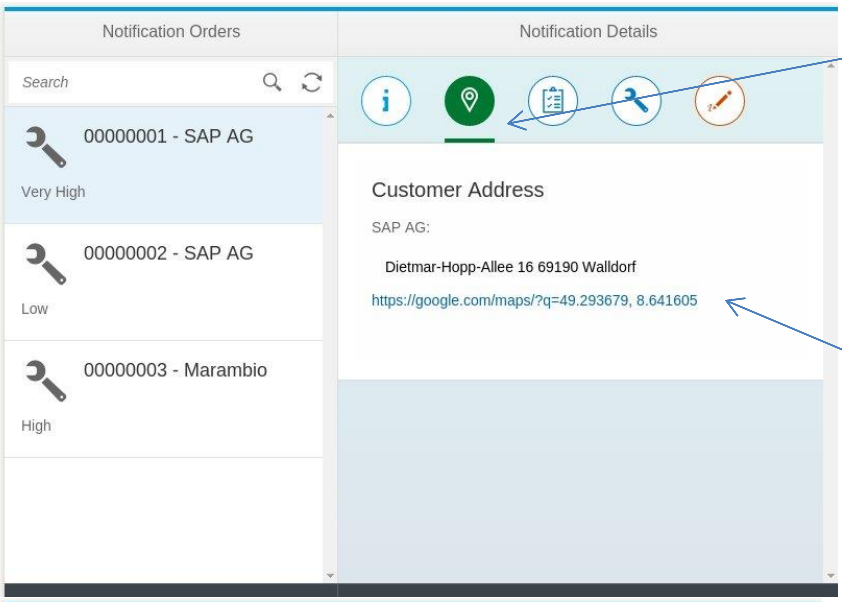
00000001 - SAP AG
Very High

00000002 - SAP AG
Low

00000003 - Marambio
High

Customer:
SAP AG
OrderType:
Notification
NotificationDate:
11/06/2015

```
<ObjectListItem id="mainListItem"
press="onSelect"
type="{device&gt;/listItemType}" counter="0"
title="{OrderID} - {Customer}" icon="sap-
icon://wrench" iconDensityAware="true"
number="" numberUnit="" markFavorite="false"
markFlagged="false" showMarkers="false"
selected="true">
        <customData id="customData2">
<core:CustomData id="coreCustomData2"
key="sapDtResourcePath"
value="OrderHeaders"></core:CustomData>
</customData>
        <attributes>
        <ObjectAttribute text="{Priority}" />
                        </attributes>
<core:ExtensionPoint name="extListItemInfo"/>
                </ObjectListItem>
```



00000001 - SAP AG
Very High

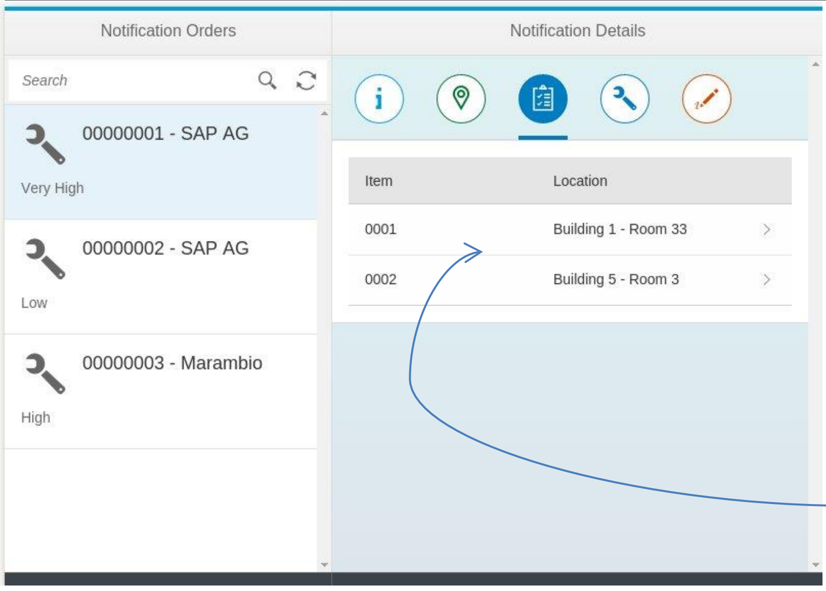
00000002 - SAP AG
Low

00000003 - Marambio
High

Customer Address
SAP AG:
Dietmar-Hopp-Allee 16 69190 Walldorf
https://google.com/maps/?q=49.293679, 8.641605

```
<IconTabFilter text="" count=""
icon="sap-icon://map"
iconColor="Positive">
</IconTabFilter>
```

```
<Link
        xmlns="sap.m" id="id"
text="https://google.com/maps/?
q={Coordinates}" enabled="true"
visible="true"
target="https://google.com/maps
/?q={Coordinates}" width=""
href="https://google.com/maps/?
q={Coordinates}"
wrapping="false" subtle="false"
emphasized="false" press="">
</Link>
```



00000001 - SAP AG
Very High

00000002 - SAP AG
Low

00000003 - Marambio
High

Item	Location
0001	Building 1 - Room 33
0002	Building 5 - Room 3

```
<Table id="idItemsTable"
itemPress="onItemDetailPressed"
items="{path: 'MaintenanceDetails'}">
        <columns>

        <Column minScreenWidth="Tablet"
demandPopin="true">
        <Text text="Item"/>
        </Column>
        <Column minScreenWidth="Tablet"
demandPopin="true">
                <Text text="Location"/>
        <items>
        <ColumnListItem type="Navigation">
        <cells>
                <Text text="{Item}"/>
                <Text
text="{FunctionalLocation}"/>
        </cells>
        </ColumnListItem>
        </items>
                </Table>
```

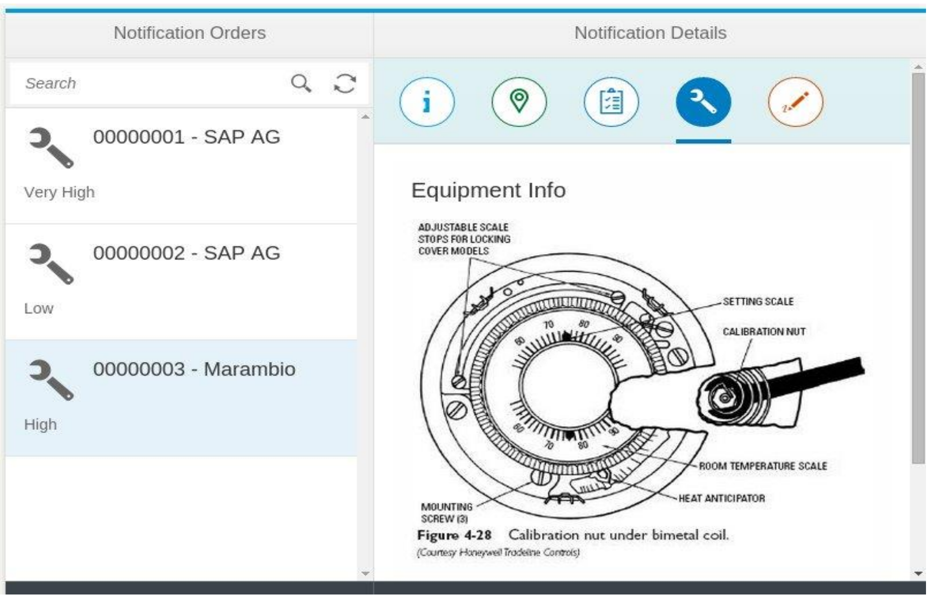
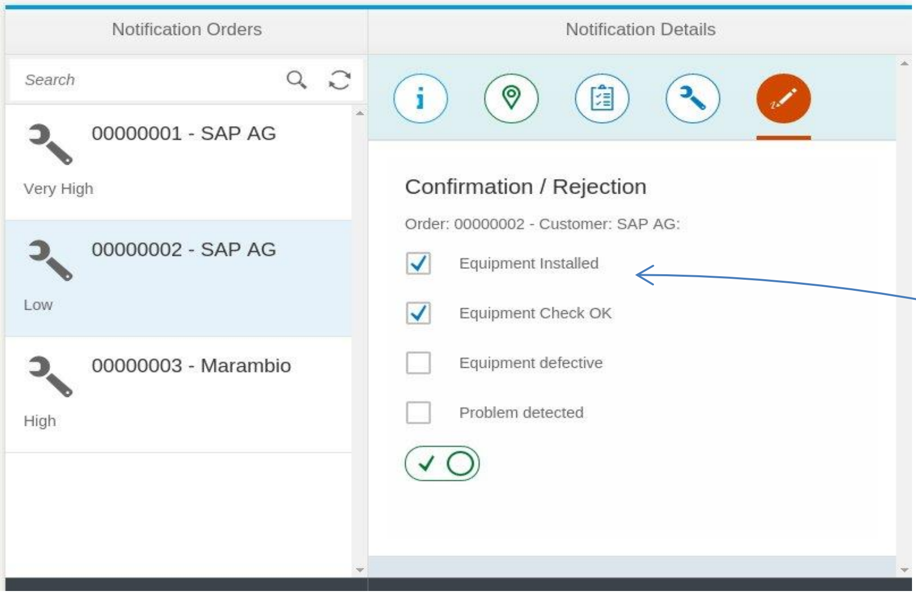


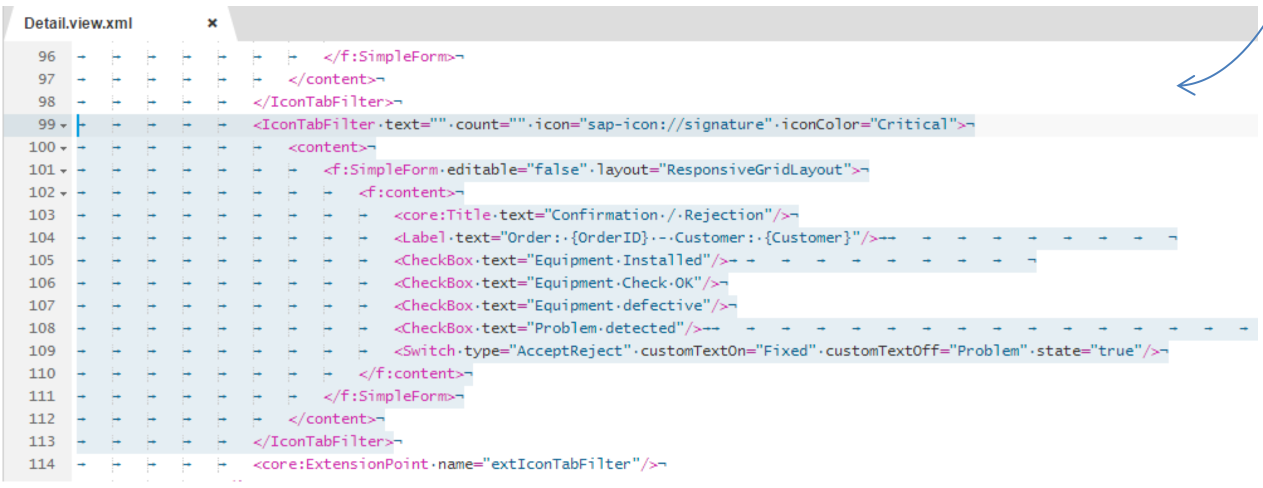
Image is coming from the “img” folder inside the Model. Is done that way for an offline scenario.



Confirmation / Rejection

Order: 00000002 - Customer: SAP AG:

- [✓] Equipment Installed
- [✓] Equipment Check OK
- [ ] Equipment defective
- [ ] Problem detected



```
              </f:SimpleForm>
            </content>
          </IconTabFilter>
          <IconTabFilter text="" count="" icon="sap-icon://signature" iconColor="Critical">
            <content>
              <f:SimpleForm editable="false" layout="ResponsiveGridLayout">
                <f:content>
                  <core:Title text="Confirmation / Rejection"/>
                  <Label text="Order: {OrderID} - Customer: {Customer}"/>
                  <CheckBox text="Equipment Installed"/>
                  <CheckBox text="Equipment Check OK"/>
                  <CheckBox text="Equipment defective"/>
                  <CheckBox text="Problem detected"/>
                  <Switch type="AcceptReject" customTextOn="Fixed" customTextOff="Problem" state="true"/>
                </f:content>
              </f:SimpleForm>
            </content>
          </IconTabFilter>
          <core:ExtensionPoint name="extIconTabFilter"/>
```