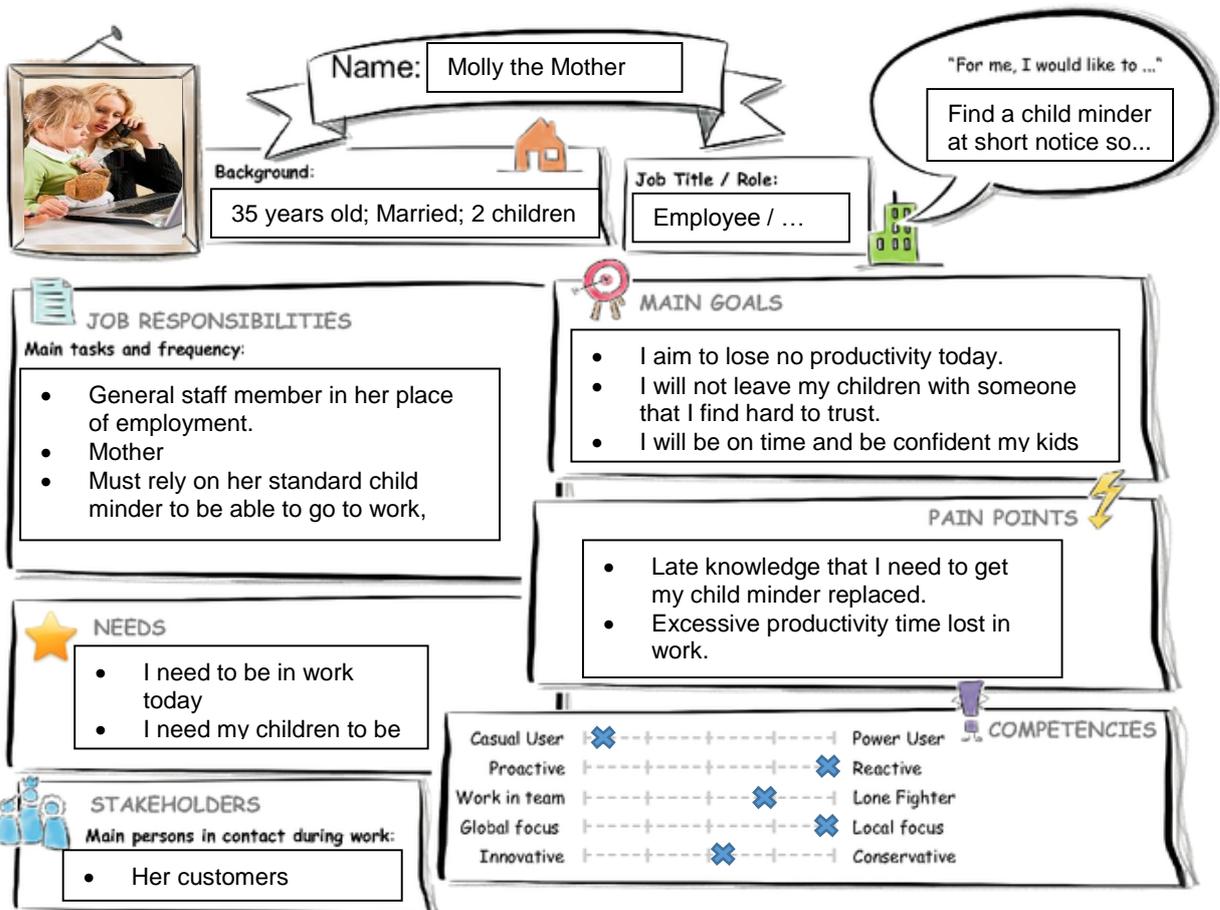## Step 1: Story:

My final assignment will be a Master Detail Fiori application based on the concept of finding a child minder at short notice. Specifically, you would have a master list of child minders listed by availability which you can filter. The detail list would allow you to check their credentials and approve or decline them. It will be called "MinderFinder".

Segmentation: Target customer for this application is a mother (or father) in business. One who relies on their partner / permanent child minder during the day to look after their children while they are in work. My application will come into use as a service within a business when a breakdown occurs on occasion within the usual plan.

Targeting: This application could be deployed on any business' Fiori Launchpad as a service. If someone is stuck, they can log on to the system, run this application, scan for available child minder's that the company endorses in their area and get it sorted before the work day starts. I will go through how this might work in detail in my prototype.

Positioning: This application might have two to three different views. The primary view will be for the user, let's call her Molly the Mother. Molly will log on, see a master list of minder's which can be sorted by rating, distance to the office or availability. The detail section will then show qualifications, recommendations, fee, etc. in extra buttons and allow acceptance of the minder. Another view might be for child minders to register their details in a simple form and a third might be for the employer to manage the list.

Step 2: Persona:

Name:
Molly the Mother

"For me, I would like to…":
Find a child minder at short notice so that I can still go to work!

Background:
35 years old; Married; 2 children, aged 3 and 7; Bachelor of Computer Science; 10 years working in SAP as a software developer; Lives not too far from the office; Relies on child minder to look after her kids after school before she comes home and full working hours during the summer.

Job Title / Role:
Employee / Mother.

Job Responsibilities (Main tasks and frequency):
- General staff member in her place of employment.
- Mother.
- Must rely on her standard child minder to be able to go to work, otherwise she cannot go to work.

Main Goals:
- I aim to lose no productivity today.
- I will not leave my children with someone that I find hard to trust.
- I will be on time and be confident my kids will be happy and safe until I return from work this evening.

Needs:
- I need to be in work today.
- I need my children to be looked after.
- I need to sort this out on very short notice.

Pain Points:
- Late knowledge that I need to get my child minder replaced.
- Excessive productivity time lost in work.
- Letting my fellow employees / customers down by not being able to attend work on an important business day.

Stakeholders:
- Her customers
- Her fellow employees
- Her manager
- All of whom will be left at a loss if she cannot make it into work.

Competencies:

In terms of this application, Molly will be a casual user, she will be very reactive to her situation, and she will mostly be a lone fighter. The application will have a heavily local focus (but will be able to be deployed globally on any HANA system). She will be slight more conservative than innovative in terms of her use of this application.

Step 3: User Experience Journey:

🔴 = Negative Experience, 🔵 = Positive Experience, rest = neutral.

MINDSET

- I'm going to be late 🔴
- What am I going to do?
  I have an important meeting today 🔴
- I can't afford to take the day off 🔴
- My customers need me
- How could they let me down like this?
- Will the kids be safe? 🔴
- This child minder looks like she's responsible /capable
- Thank God I found someone 🔵
- I feel comfortable that they are safe now! 🔵

ACTIONS

- Receive a call from my child minder
- Worry about what to do 🔴
- Make calls to friends to find a replacement
- Log on to the PC
- Log on to the child minder app in my company's Fiori Launchpad
- Decided on a suitable child minder for the kids from the options
- Make arrangements for a child minder to pick up the kids from work 🔵
- Say goodbye to the kids before they leave
- Get to work and sit down at desk 🔵

TOUCH POINTS

- Laptop
- Company Fiori Launchpad
- Kids
- Minder
- Ensuring Qualifications of child minder
- Ensuring Experience of child minder
- Ensuring Rating / Feedback of child minder

The duration of this user experience journey **will not exceed an hour** from receiving the bad news to sitting down at her desk in work (dependent on child minder availability / notice required). In general, the user will be worried about what to do, they will remember the app, find a suitable minder and have the kids picked up from outside the office. They can then go about their standard day without needing to worry.
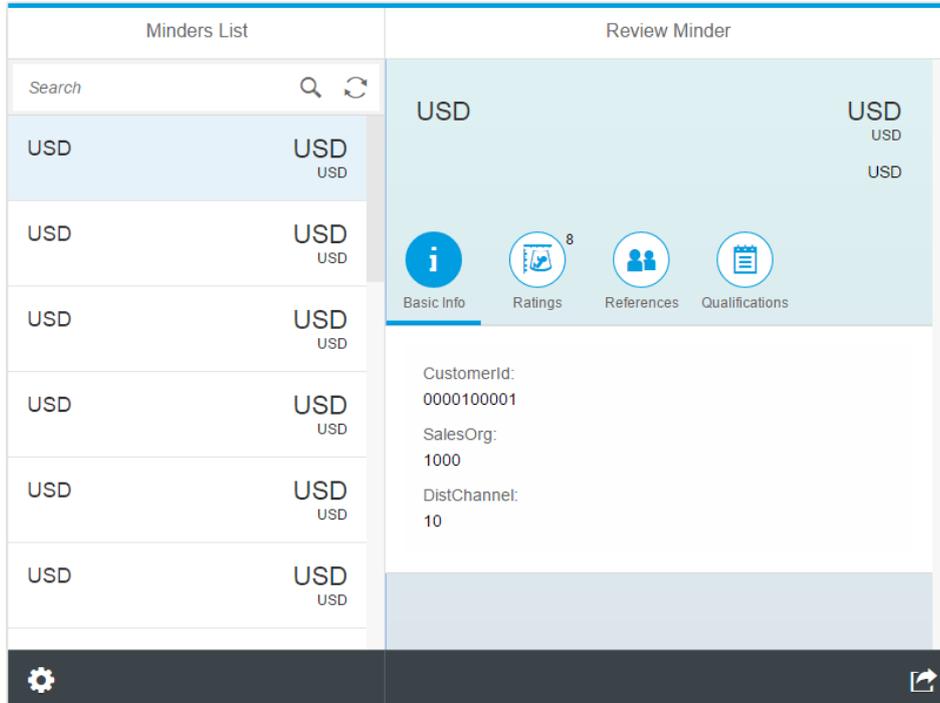
Step 4: Mock up:

This is the "customer" view of my application. When a user logs on to their company Launchpad, they will see this screen of minders. It is sorted by default by rating, but can be changed to be sorted by salary, name, distance or availability. From here they can choose a child minder in the master list and then review their basic information, ratings, references and qualifications in the detail pane before hiring a particular minder or not. If they have previously hired this child minder, they will also see an option to rate the minder in the ratings tab.

Two other views will exist in this application. A "child minder" view, which will basically be a form that a pre-approved child minder can fill in. This will list them on the customer view after approval by someone in the company' admin team. The third view is the company view, where they would approve new child minders, manage existing ones, change information on the minder's etc.
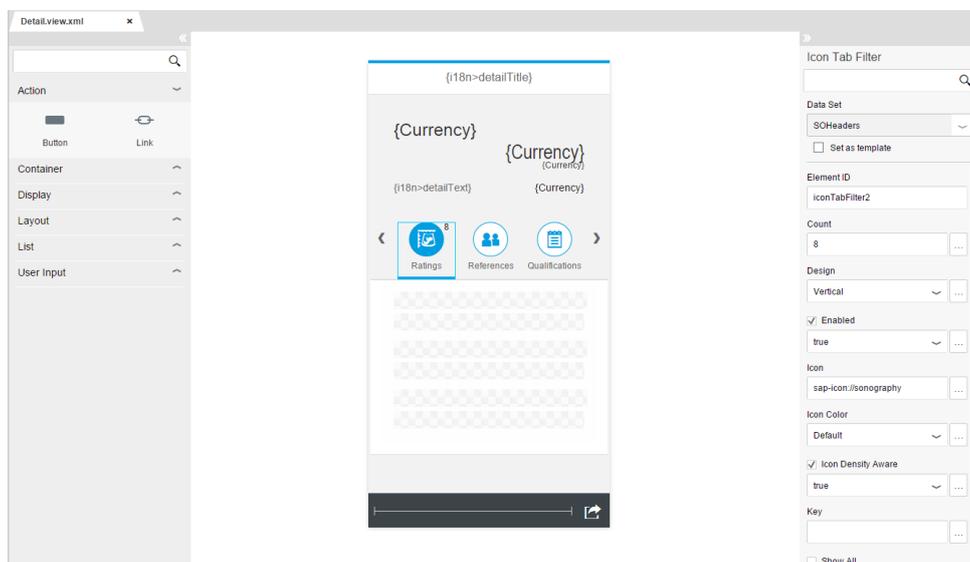
On our main "customer view", when the "Hire" button is pressed, the user will be taken to a screen with a quick form that asks what days they need the child minder, how fast they are needed, contact details etc. This goes straight to the child minder who calls them back and then confirms on the system that they are booked. This can be done with one button press on the phone application.

Step 5: App Prototyping, design component 1:

- The first design component of my application that I implemented in Web IDE was a full set of Icon Tab Filters that I would need to display the information about each child minder.
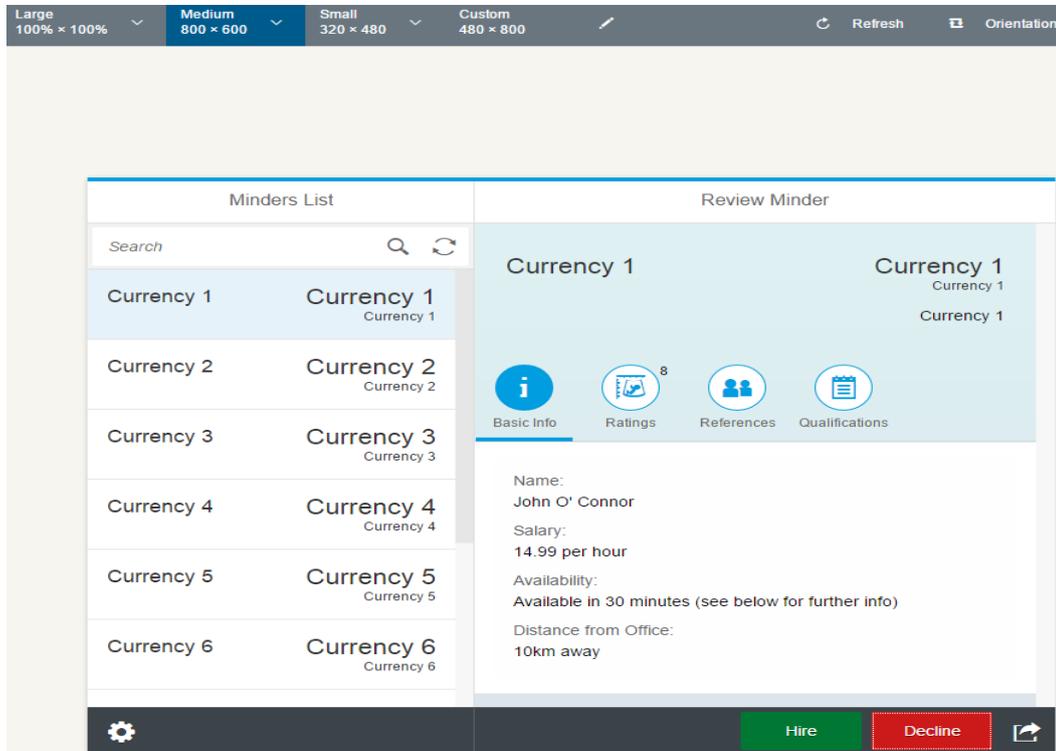

- This involved two main things. The first was finding the Icons I needed (and had used in my mock up in step 4) from the Icon Explorer in the UI5 toolkit. Once I had found these, I used the Web IDE's Visual Layout Editor to create three new "Icon Tab Filters" beside the existing one. For each of them, I then clicked on the tab and filled in a text for underneath them, changed their color to default, and changed their sap-icon to point to the correct one. For the rating's tab, I also put a counter in. This is just a placeholder number at this stage, later it would be bound to a ratings table.

Step 5: App Prototyping, design component 2:

- The second design component of my application that I implemented in Web IDE was a simple form displaying information about the basic details of the child minder on the information tab.

- This involved modifying the names of the three existing text fields in the layout editor. I also needed to add a fourth text fields to the form, as it wasn't in the template. To do this, I used the Code editor, and added in the code for a fourth label based on the existing ones. Doing this in the layout editor was proving troublesome for me. After doing this, I filled in the mock data for this form. For now, I simply overwrote the existing bindings from the template, but with more time I would make my own bindings from scratch. I also added in my hire and decline buttons at this point as shown.