

## Client: Barclays Design Office

**Project: Enterprise application to make onboarding corporate clients quicker and easier for Relationship Managers**

### What I did - main activities

- UR** User research: conducted 1-1 interviews with Relationship Managers to understand their role, and their problems.
- IA** Organised app features and tasks around user goals and mental models, for clear navigation.
- IxD** Defined app interactions from an existing pattern library, and defined new patterns where needed.

### Main deliverables

- Axure prototypes to explore, text and validate design concepts.
- Working documents including site maps, journey maps, and wireframes.
- Definition documents including full interaction specification documentation.

### User research

Understanding the needs of the people who will use the product

The product needed to accommodate myriad business rules and processes. Some of these were internal, and could be modified. Some were external, and could not be changed. And all of them needed to be re-framed to suit the needs of the users of the product, Barclays Relationship Managers.

### Building the right thing

A fundamental question at the beginning of any project is "what are we making and why?". At Barclays, I tried to answer these questions by speaking directly to the people the product was for: the relationship managers whose job it is to onboard new corporate clients.

I didn't ask relationship managers to design the product: I asked them to "teach me their job". That way, I got to understand their requirements, and frustrations, and started to learn how to think like them.

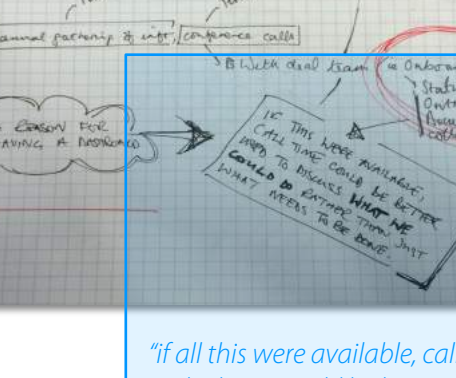
It also helped build rapport and trust between user groups and designers.

This led to useful insights about scope and features of the product and what it needed to do.

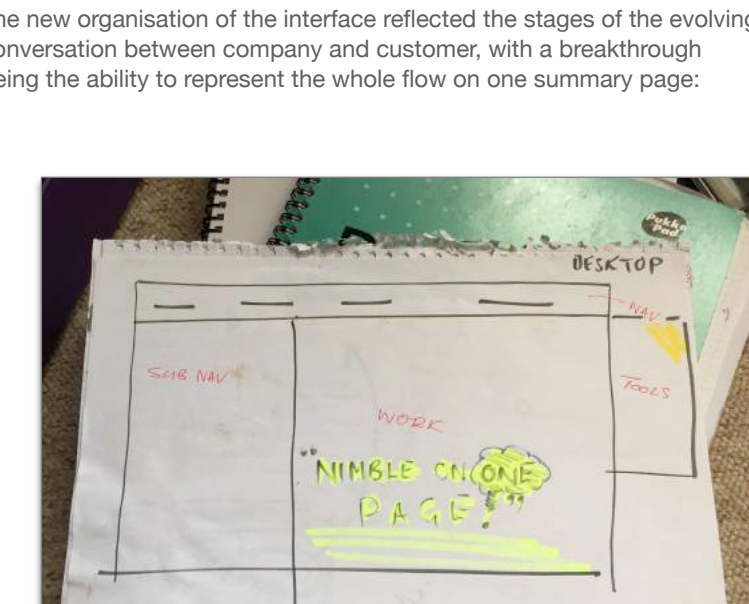
(Later on, I held 1-1 sessions to evaluate and test ideas more formally for feedback.)

### Business logic versus user workarounds

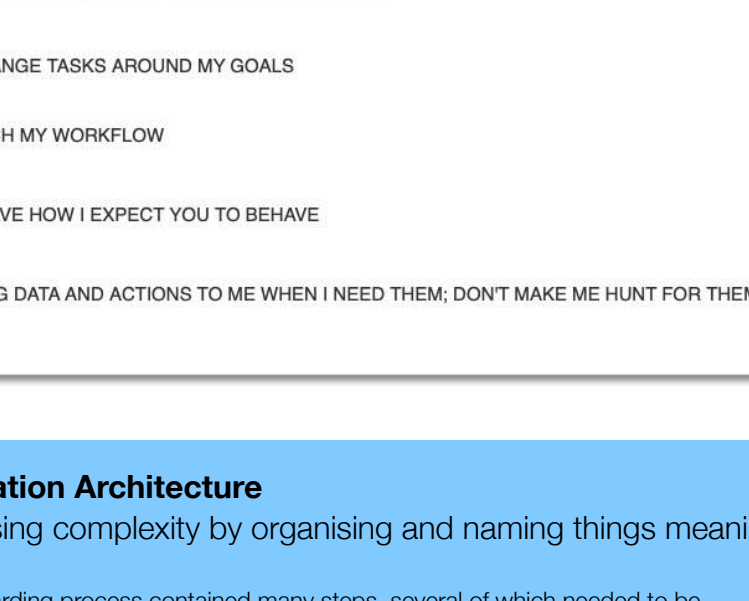
Early notes show a developing understanding of process – not just the official documented process, but the workarounds and shortcuts that experienced staff took every day. These workarounds were pointers to some of the real problems that needed to be solved.



As my familiarity with the relationship manager role grew, and my understanding of real user needs became more clear, I began to rough out ideas for improvements – either at the service end of the product, or at the more detailed interaction end, or both at the same time:

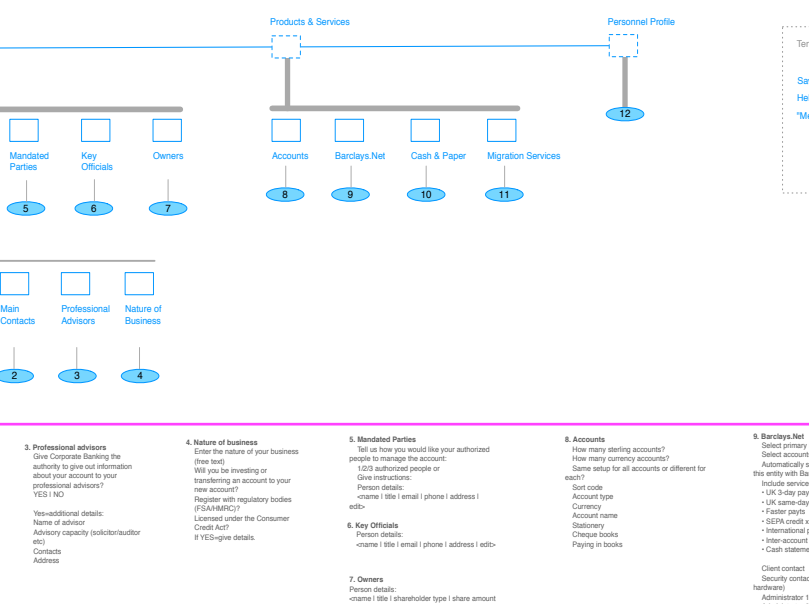


Sketching alongside users as they spoke, or shortly afterwards in follow-up chats, led to some great, informal and unbounded "what-if?" conversations, and captured important insights that would shape later design work:



For example, the small note in the call-out below contains a chance user remark that threw new light on the whole purpose of the project.

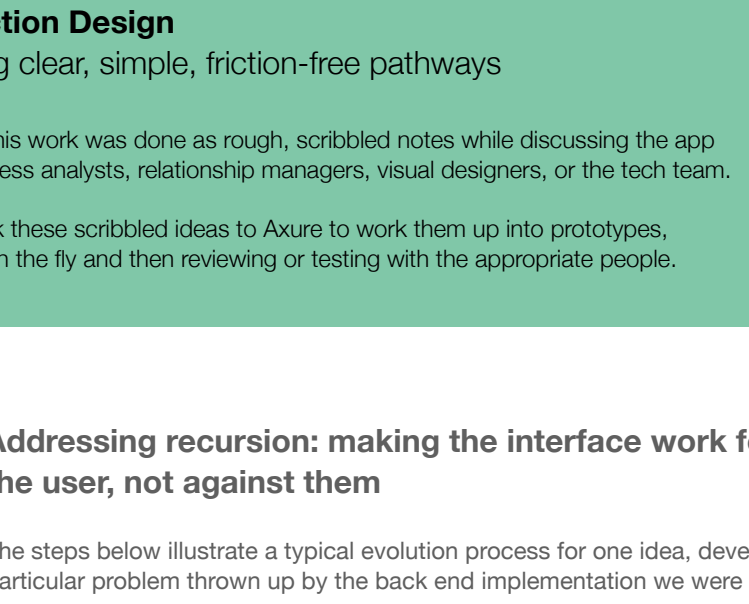
While the functional objective of the application was to gather administrative and legal details, the broader objective was to allow relationship managers to expand and develop their relationship with the customer: they wanted to offer more services and up-sell, not badger them to complete a form.



"If all this were available, call time with clients could be better used discussing what we could do for them, rather than just what needs to be done."

... and this insight led to a completely new idea for interacting with the app, framing tasks around the conversation between relationship manager and customer.

The new organisation of the interface reflected the stages of the evolving conversation between company and customer, with a breakthrough being the ability to represent the whole flow on one summary page:



This phase of work gave a clear set of user insights which I balanced with the business requirements to define scope, objectives, constraints and direction for the new interface.

I distilled the findings into 5 key design principles:

- NAVIGATE ACCORDING TO HOW I SEE MY TASK**
- ARRANGE TASKS AROUND MY GOALS**
- MATCH MY WORKFLOW**
- BEHAVE HOW I EXPECT YOU TO BEHAVE**
- BRING DATA AND ACTIONS TO ME WHEN I NEED THEM. DON'T MAKE ME HUNT FOR THEM**

## IA

### Information Architecture

Addressing complexity by organising and naming things meaningfully

The onboarding process contained many steps, several of which needed to be completed at different times. Onboarding has several potential different pathways through all of the steps, depending on the nature of the new business.

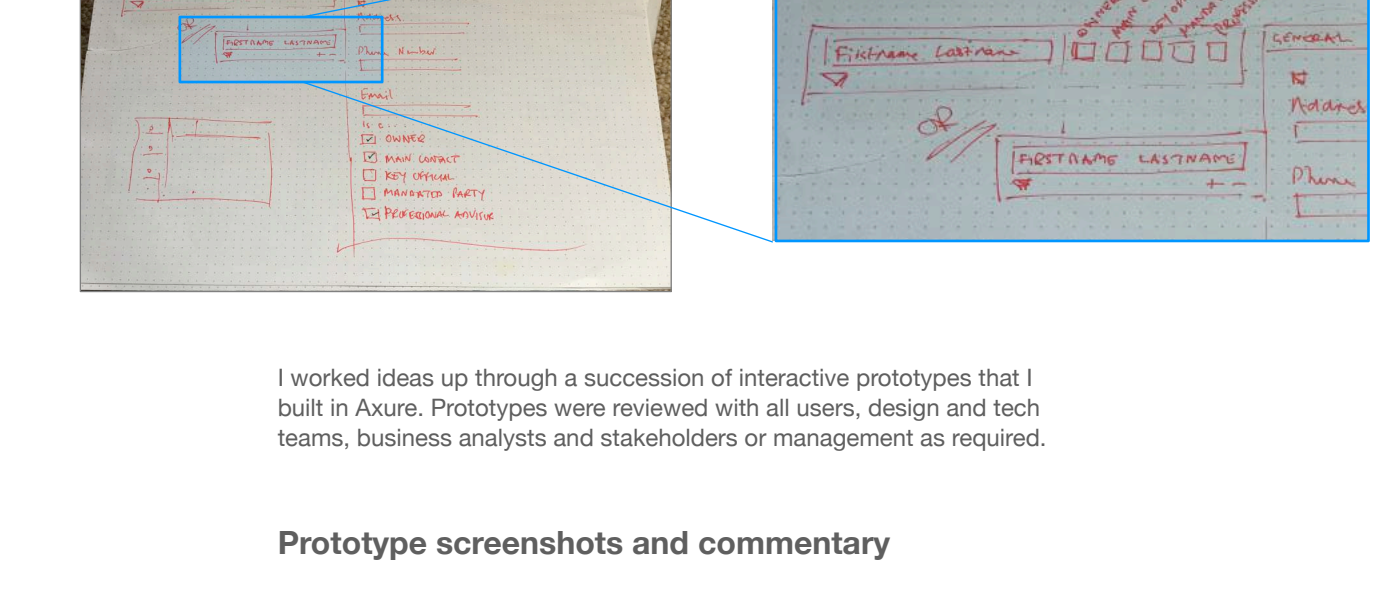
Users needed to track progress, fix errors, see what was missing, and be able to offer quick help to new clients.

All of this needed a very clear presentation and organisation of tasks, and an overview that would show progress at-a-glance.

### Creating an overview: a map of the application

This map was a key working document, and a milestone on the way towards achieving a clear and coherent interface.

It was the first time that all the major task groups had been identified, summarised, grouped, sequenced, and shown on the same page:



The map hides a lot of background work: where task flows were repeated, I re-organised or re-named to remove the repetition; where tasks had confusing or ambiguous names, I renamed them to be meaningful for the user; where flows existed as orphan tasks groups apparently without context, I grouped them with the correct parent to create cohesion within the app.

**A** This was the first time that the complete set of tasks had been summarised in a single view, and gave a clear idea of the breadth and scope of the product.

Tasks could be seen in relation to each other for the first time, creating context and better understanding of the product among stakeholders and designers.

**B** Next I introduced a second grouping level to make the task groups easier to navigate and explore.

The naming of this grouping layer was matched to user mental models and the goals they had in using the product.

**C** Finally, I organised the second-level task groups under major onboarding steps. This organisation is what allowed a much simpler overview page to be created that gave an at-a-glance view of progress.

As well as helping to untangle the underlying business logic, this work allowed me to create a clear and simple drill-down into a large number of different sections, pages and individual tasks.

It allowed different tasks to be completed at different times, and the structure allowed a summary page to show progress.

All of this was worked out in detail during the interaction design stage, which always includes detailed IA as its start-point but which I've separated here just for clarity.

## IxD

### Interaction Design

Creating clear, simple, friction-free pathways

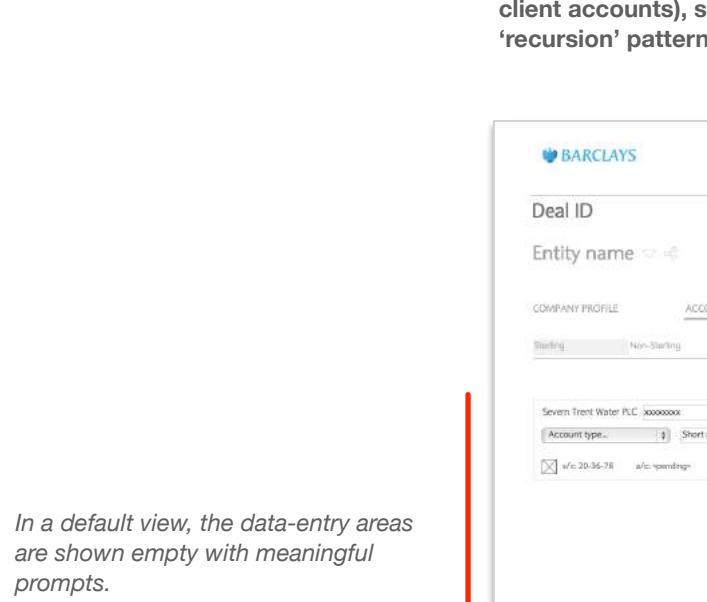
Much of this work was done as rough, scribbled notes while discussing the app with business analysts, relationship managers, visual designers, or the tech team.

I then took these scribbled ideas to Axure to work them up into prototypes, iterating on the fly and then reviewing or testing with the appropriate people.

### Addressing recursion: making the interface work for the user, not against them

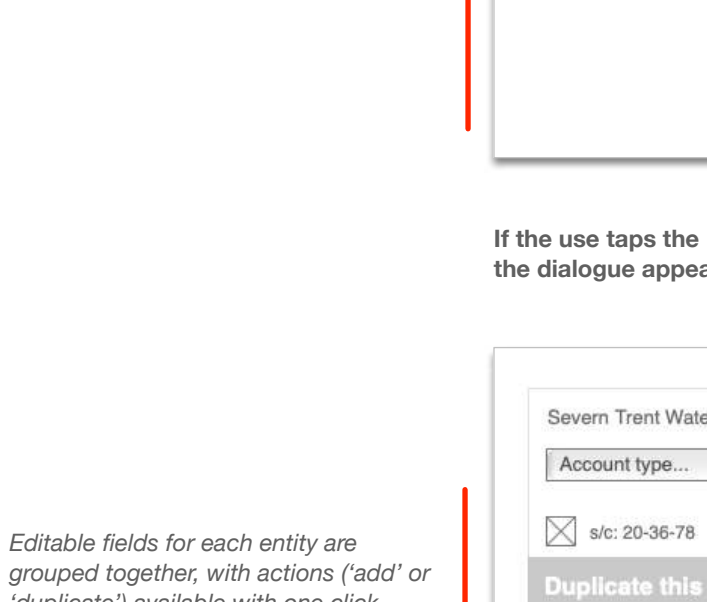
The steps below illustrate a typical evolution for one idea, developed to solve a particular problem thrown up by the back end implementation we were using.

#### The problem



A research note emphasises the frustration users felt when forced to enter the same data set for multiple objects. Often, each set was identical, apart from one or two elements...

...and this was compounded by the structure of the interface, which forced users to follow any tasks page by page.



Up to 48 repeated steps

While creating an initial feeling of simplicity – the user seeing only a few fields at a time – it actually created significant usability issues:

- Prevented exploration and learning.
- If a process could not be completed in a single session, users had to navigate step-by-step away from, and later on back to, the place they had reached.
- Created feelings of disorientation – users had no concept of progress or location within the broader context of a single client's application.
- Not possible to see multiple clients' progress as an overview.

All of this added up to a confusing, frustrating and error-prone experience for the user.

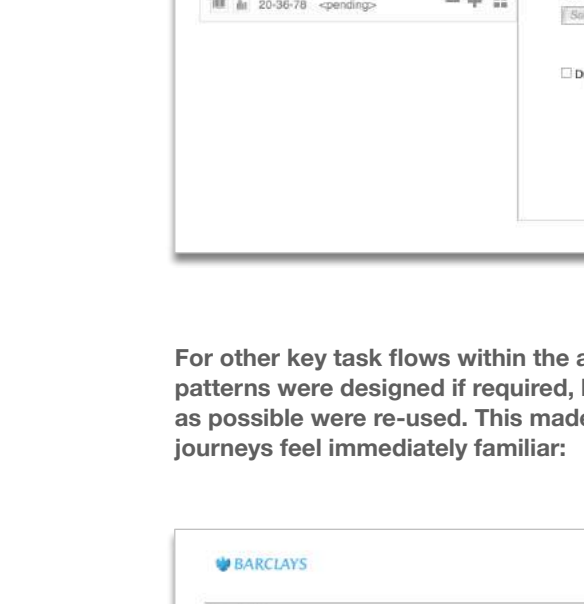
### The solution: one new micro-interaction that unlocked a whole new user experience

Stepping back from the problem and summarising it from the user's perspective pointed towards a different form of expression for the interface.

Instead of simply replicating the structure of a database on the back-end, and prompting users to complete form fields over and again, I wrote the problem as a user goal:

**"I want to replicate a set of data, then change single elements within the set easily and at any time."**

That insight suggested a different interaction:



Here, each parent-object (a branch of a supermarket, for example) is listed on one side of the interface. Selecting one of these objects reveals associated tasks: including 'duplicate'.

This solved the recursion problem with a single click.

Once another version of the same data set is stamped out, the user just has to change the few unique elements (the address, for example) – and move on to the next legal entity to set up.

It eliminated repetition and redundancy, and created an obvious point at which to surface progress, alarms or alerts.

And it reduced page-count for a task from up to 48 down to one page with three sequential panes.

Not only that, but the same interaction pattern could be applied throughout the app, leading to even more efficiency and reducing training time:



I worked ideas up through a succession of interactive prototypes that I built in Axure. Prototypes were reviewed with all users, design and tech teams, business analysts and stakeholders or management as required.

### Prototype screenshots and commentary

The prototypes I created followed key journeys from beginning to end. They contained no final visual design, but I paid very careful attention to **typography, information density and visual hierarchy**, as a guide to the visual designers who would pick this work up to complete. The success of the design of this interface depended upon subtle visual balance, as well as the detail of the organisation and interactions beneath the surface.

The 'home' page, after logging on, shows all deals under the relationship manager's responsibility:



The overview page gives a high-level summary of progress for all clients.

Status bars show sections that are not started, incomplete or flagged for attention.

In the summary pane are key contacts and the target completion date. This gives a measure for progress and quick access to all deal stakeholders.

For each 'entity' in a deal – supermarket, branch or admin office – the status bar is expanded, and grouped to summarise each key stage of the onboarding flow.

Deal-level tasks are shown in the header: these are available only when all lower-level tasks are complete.

The data-entry pane is always shown under the client's header, for context.

To aid the relationship manager, status indicators are clickable, and show anything that is outstanding, or otherwise flagged.

This allows managers to quickly and easily trace work done by various team members.

