Florian Gouloubi

Senior Product Designer

Site	floriangouloubi.com
Email	floriangouloubi@gmail.com
LinkedIn	linkedin.com/in/floriangouloubi

Case studies

01 **BetRivers Poker** 3
Rush Street Interactive · Design System & Design Ops

02 **Archipelago** 16
Pernod Ricard · Design System Governance

03 **Bridge** 48
Own product · AI Exploration Platform

04 **CardPlayer** 65
ClickOut Media · Template Ecosystem Redesign

05 **Poker Training Simulator** 88
ClickOut Media · Architectural Rebuild

06 **Poker Odds Calculator** 101
ClickOut Media · SEO Growth Tool

07 **FCODE** 112
IDEMIA · Tablet App Design

08 **MBIS** 128
IDEMIA · Forensic Software Research

09 **Verinium** 142
IDEMIA · Biometric Access Terminal

10 **Phygital** 159
IDEMIA · Field Research

11 **Program Dashboard** 169
IDEMIA · UX Research

12 **Loxam** 192
Peppermint Company · B2B Mobile App

13 **Ynergie** 211
Peppermint Company · B2C Web App

14 **Delivery Park** 223
Peppermint Company · Mobile App Redesign



Case study · Rush Street Interactive, iGaming app

Product Design, Design System & Design Ops

BetRivers Poker is a real-money poker platform operated by Rush Street Interactive (NYSE: RSI), targeting the regulated US market. I joined pre-launch as the design systems lead, to accelerate production velocity across desktop, mobile and tablet.

My role	Senior Product Designer | Design System Lead | Design Ops
Crew	4 Product Designers, 1 Product Director, 1 QA, 1 PM, dev team
Problem	The product was in active development but lacked DS governance, had no structured handoff process, and carried massive design debt. The challenge: rebuild the DS foundation, establish Design Ops workflows, and enable the dev team to ship consistently across PA, MI, DE and WV.

Mission 01 — Enhancing the Poker Design System — 5 sub-missions

Mission 02 — Scaling the Design Ops Strategy — 4 sub-missions

Mission 03 — Designing & Structuring the MTT — 5 sub-missions

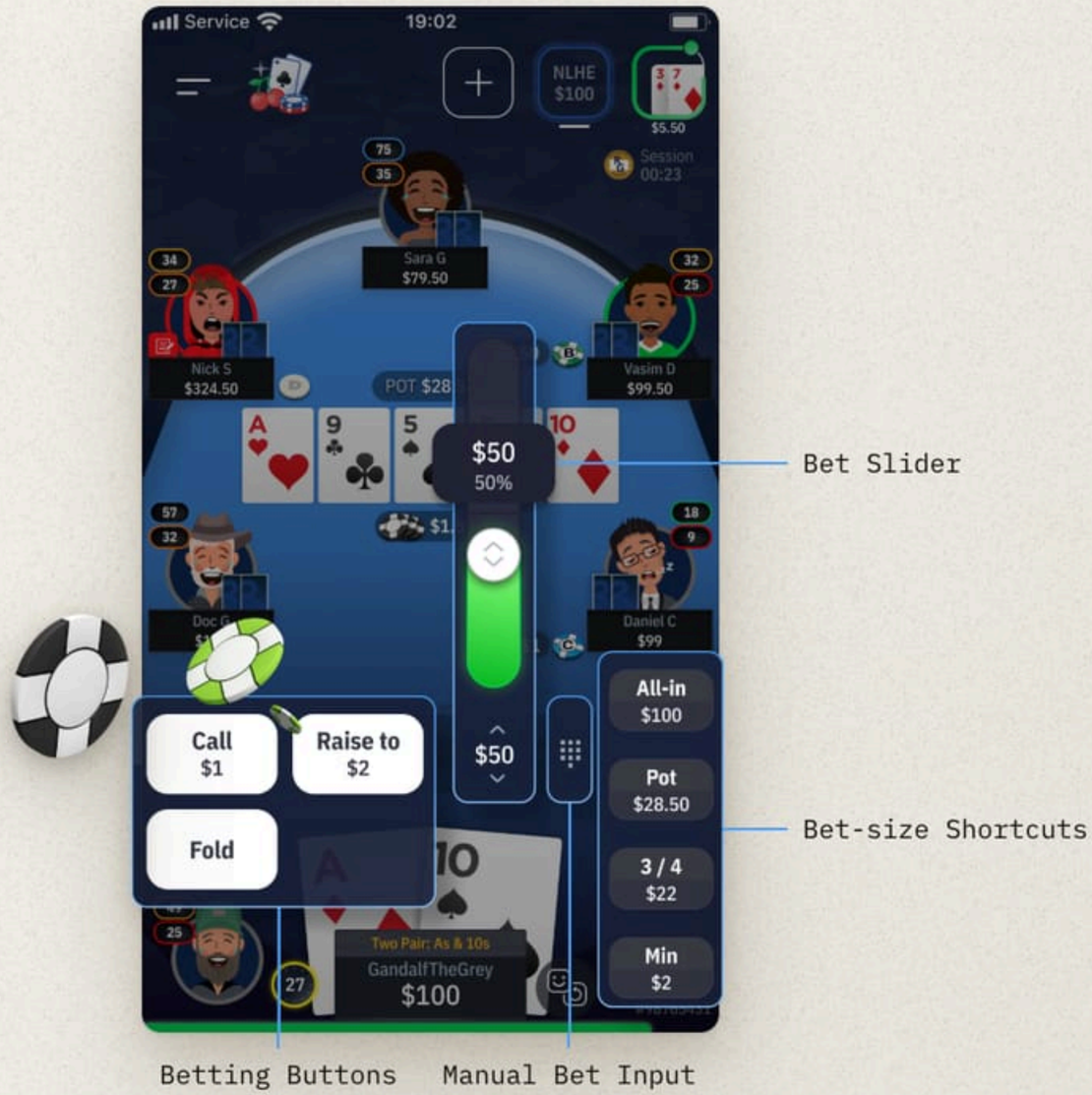
Mission 04 — Micro-Interactions & Gesture Design — 5 sub-missions

Mission 05 — Structuring Documentation & Handoff — 6 sub-missions

Enhancing the Poker Design System

BetRivers Poker lacked a structured Design System, leading to inconsistencies, broken components and heavy files. With no clear naming conventions or validation process, scaling design was inefficient. The challenge was to streamline production while integrating with RSI's global DS.

Component optimization	Design debt reduced with an atomic design strategy.
Naming & structuring	Clear file structure, DS alignment, handoff processes.
Accessibility & UI quality	Styles standardized, UI consistency refined.
Integration with RSI DS	A custom Poker DS that co-exists with RSI's.
Refined UI Kit structure	Base components separated from contextual ones.



Scaling the Design Ops Strategy

The lack of Design Ops processes created handoff inefficiencies and collaboration gaps. Developers relied on scattered specs across Slack, Asana, and Figma, causing delays and misunderstandings. Additionally, poor file governance led to frequent version conflicts and lost work. The goal was to establish clear workflows, documentation, and governance to optimize efficiency.

UX team organization audit

Pain points, impact-effort matrix, roadmap.

Centralized documentation

Specs, navigation, guidelines, animations.

File & branch management

Governance rule and branching guidelines.

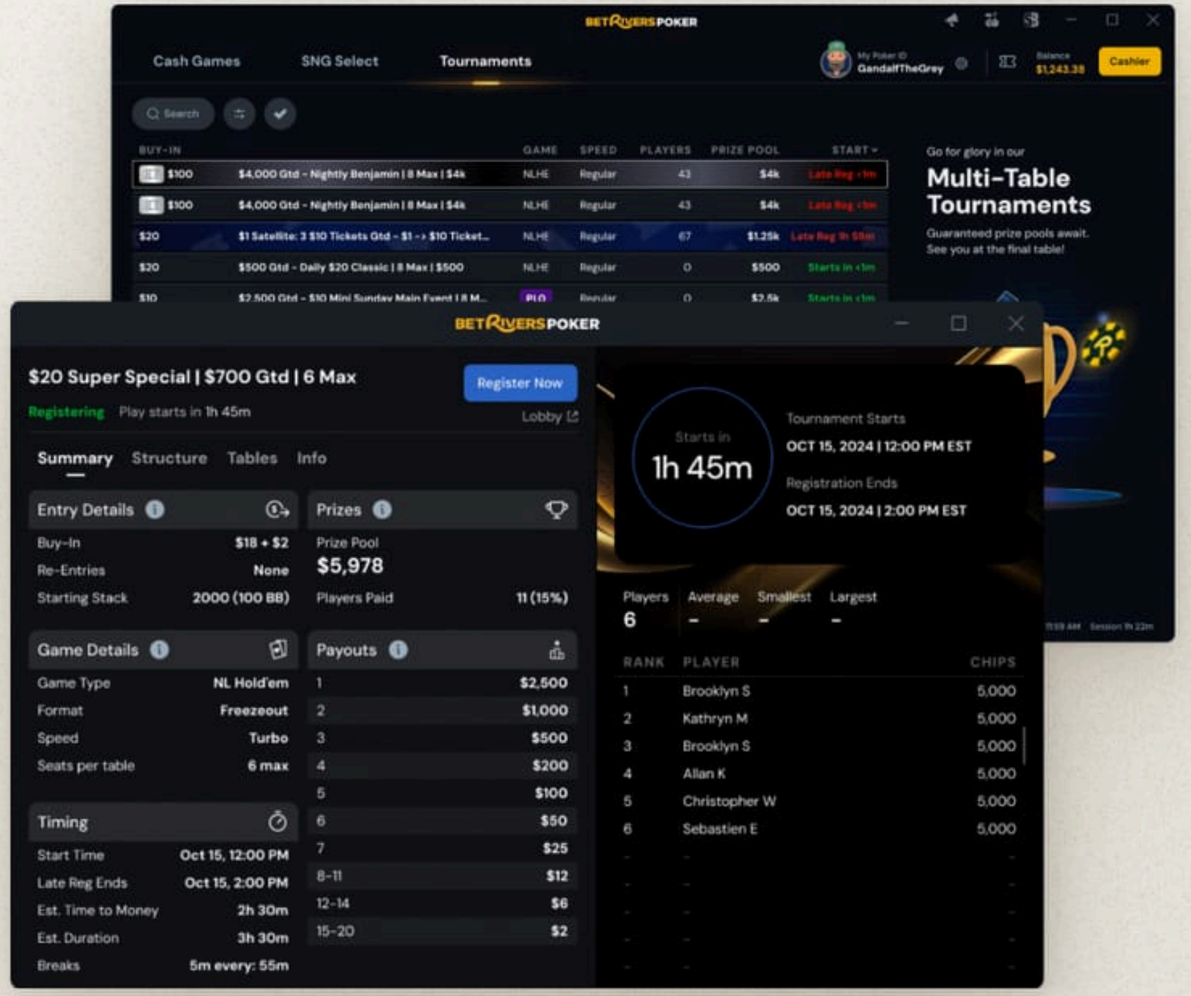
Communication process

Between the RSI and Poker teams.

Designing & Structuring the MTT

The MTT Lobby & Detail Lobby serve as crucial interfaces for players to register, track tournament progress, and access key information. When I joined the project, the design was incomplete, missing key use cases, and lacked a structured handoff process. My role was to structure, finalize, and optimize the design while ensuring a smooth developer handoff.

Project structure	Structured to ensure efficient development.
Navigation & hierarchy	Optimized navigation and information hierarchy.
Adaptive layout	Designed for cross-platform consistency.
Contextual components	Modular tournament configurations.
Tournament templates	Standardized for scalability.



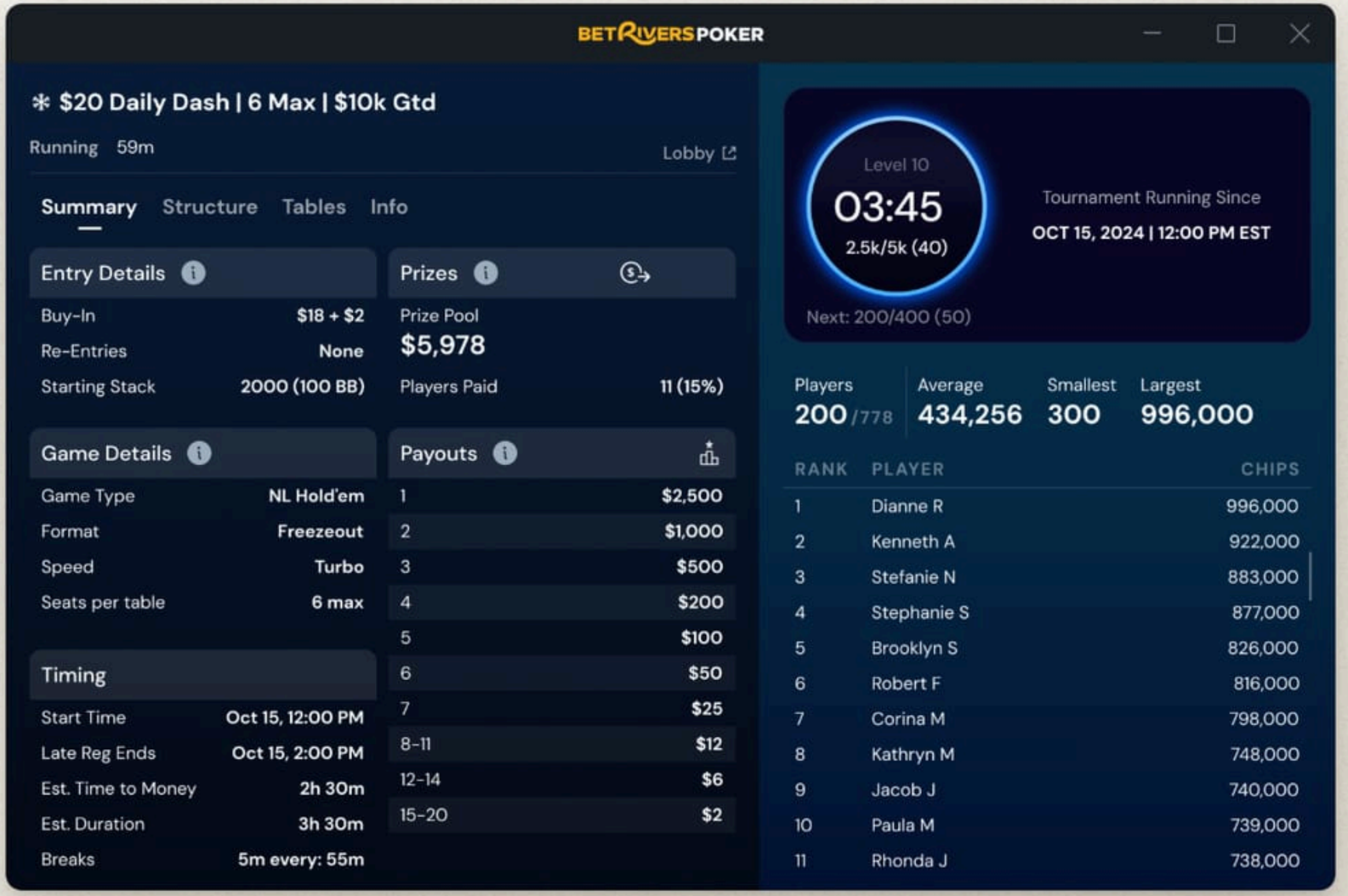
Navigation & information hierarchy

On screen

The MTT Detail Lobby of a running tournament: entry, prizes, payouts, blind level and ranking in 1 window.

What changed

Rankings, buy-ins and blind levels were poorly structured. I rebuilt the content from the Rank & Chip Summary and the Tournament Summary, with reusable display patterns.



MTT Detail Lobby · Freezeout · desktop, 900 × 600

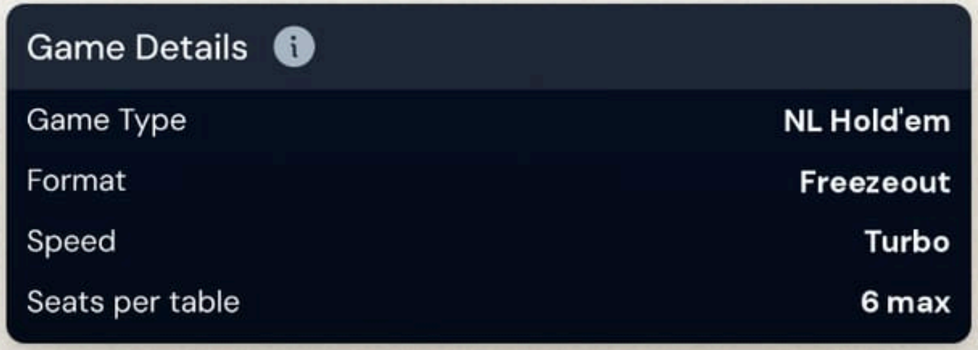
Adaptability between desktop & mobile

On screen

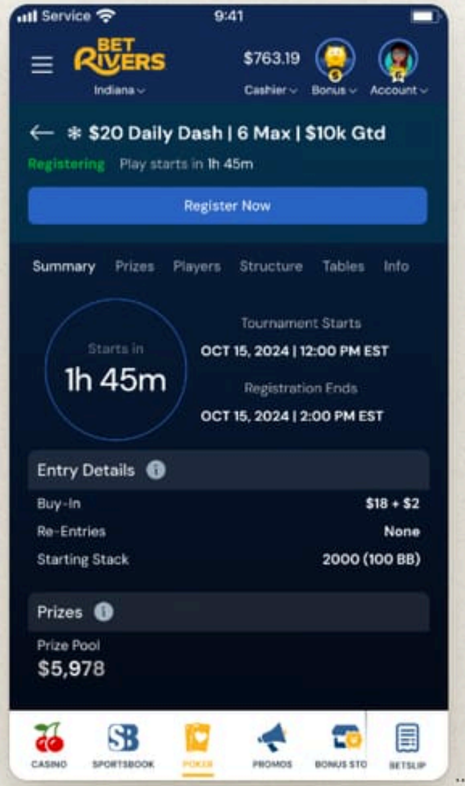
The Mobile Adaptive Layout Breakdown: the same Detail Lobby on 2 viewports, at the same scale.

What changed

No adaptation logic had been defined. I documented the rules and built modular structures, with scrollable behaviors and sub-navigation tab handling on mobile.



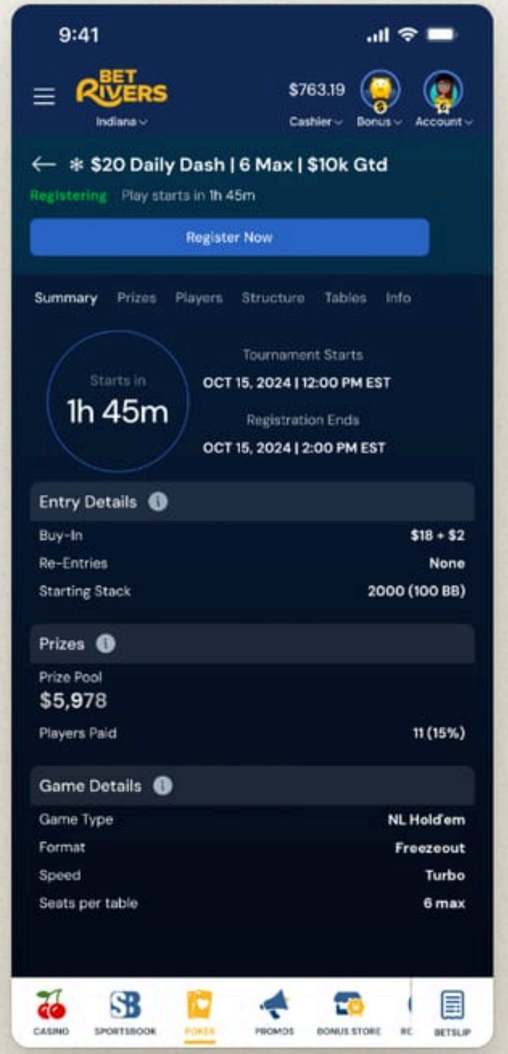
Close-up · the Game Details block, gained on the taller viewport



iPhone 8 · 375 × 667

667

+ 229 px

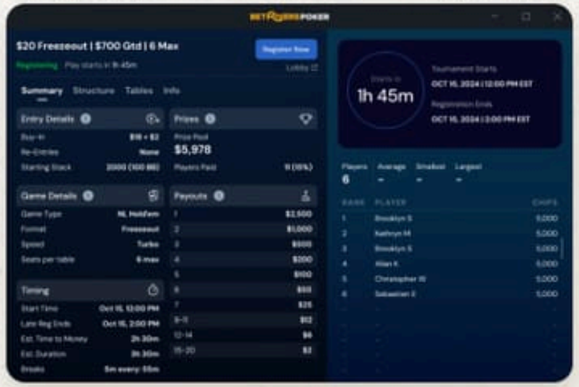


iPhone 11 · 414 × 896

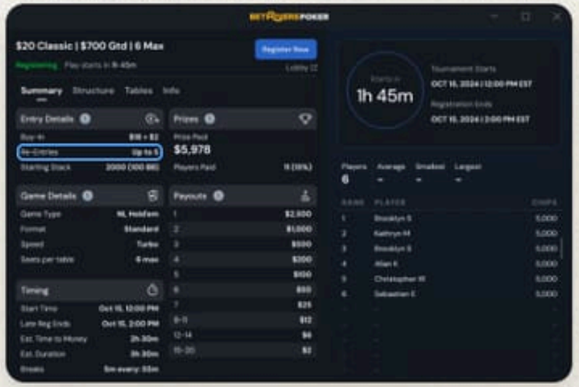
1 template, 8 tournament formats

I created the MTT Template Lobby: a modular framework that standardized tournament layouts while maintaining flexibility. 1 component, 8 variants: each format changes a skin and a few modules, framed in blue.

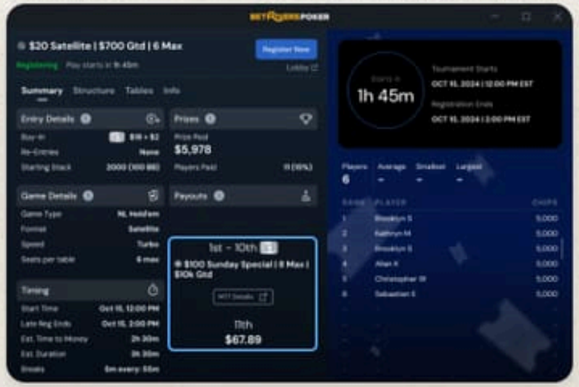
Freezeout
the base



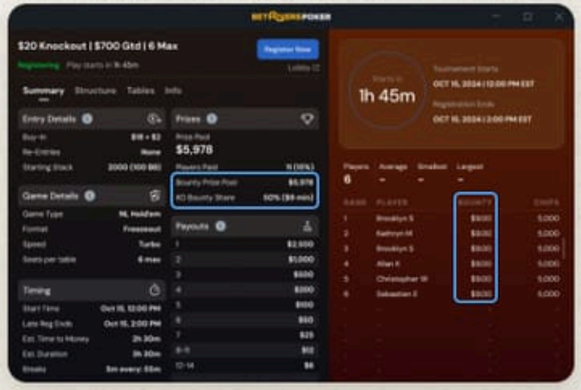
Standard / Re-Entry
up to 5



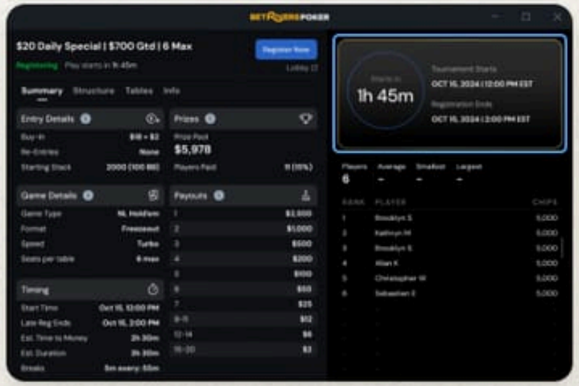
Satellite
satellite block



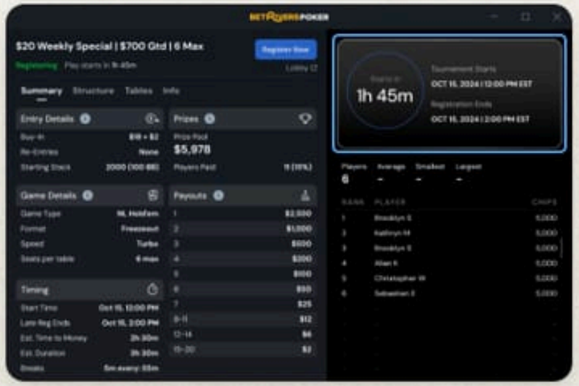
Knockout
bounty prizes + column



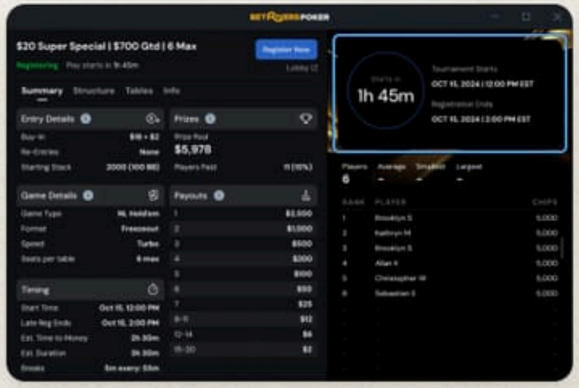
Daily Special
gold frame



Weekly Special
silver frame



Super Special
gold artwork



Freeroll
$0 buy-in, silver skin



1 page per format, every state on 1 board

On screen

The Freezeout page of the MTT Detail Lobby Template file, rearranged as a table: 5 tournament statuses, the player's situation, and its 27 screens, 13 mobile and 14 desktop.

File organization

The Re-Entry, Satellite and Knockouts pages are built the same way, each with its overview of all states.

Status	Player situation	Mobile	Desktop
Announced			
Registering	Reg starts in 1 hour+ Not registered · Registered		
	Reg starts in <1 hour Not registered · Registered		
Running	Late reg Not playing · Playing		
	Reg closed Not playing · Playing, not in the money · Playing, in the money		
Concluded	Not registered · Registered, not in the money · Registered, in the money		
Cancelled			

Micro-Interactions & Gesture Design

In online poker, every second counts. Micro-interactions and gestures must be seamless, intuitive, and non-disruptive to maintain the flow of gameplay. My role was to craft intuitive micro-interactions and gesture-based controls, ensuring platform-specific optimization for both desktop (click & hover) and mobile (tap & long-press), while preventing feature conflicts.

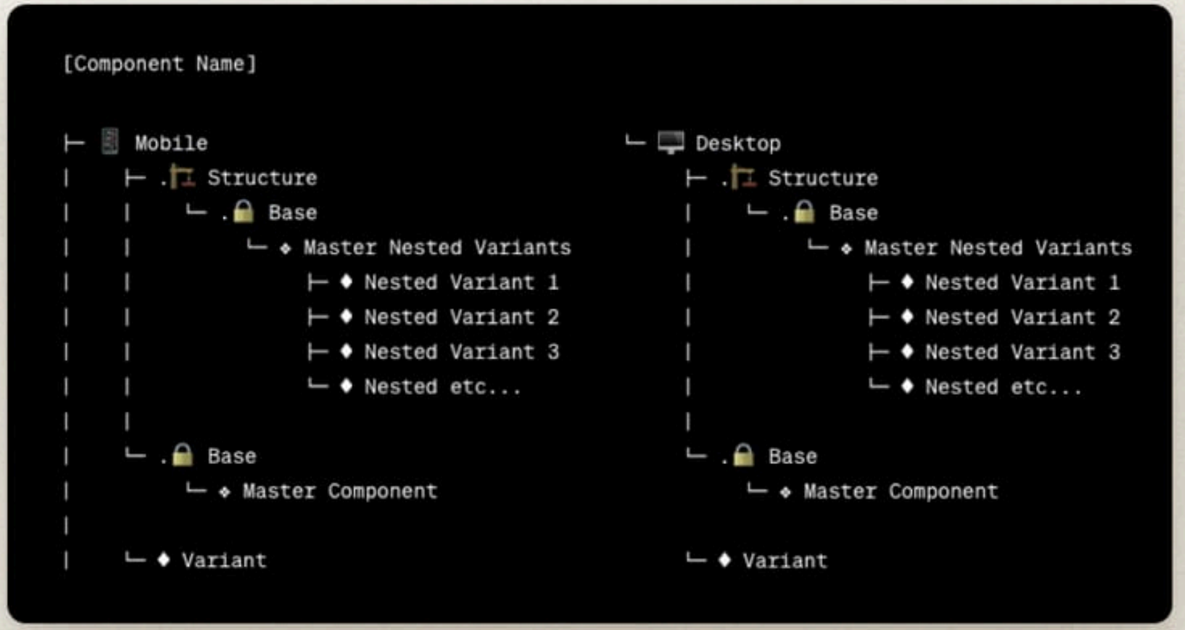
Intuitive gestures	Designed per platform, desktop and mobile.
Interaction hierarchy	No overlaps between multiple features.
Motion & transitions	Hover, selection and transition behaviors.
High-fidelity handoff	Figma branching, Rive motion specs, Asana.
Real-time feedback	Adaptive tooltips, non-intrusive toasts.



Structuring Documentation & Handoff

Given the complexity of poker UI interactions, precise and detailed documentation was essential to ensure accurate implementation. Since Figma Dev Mode was not an option, I developed a manual yet highly structured system to provide developers with all necessary design specs efficiently.

Centralized documentation	Figma files, templates and UI specs organized.
Naming & file structure	Standardized conventions.
Feature isolation	Implemented via Figma branches.
Annotations & specs	Color-coded guides, spacing rules, behaviors.
Interaction & motion	Rive and Asana for transitions and gestures.
Cross-team collaboration	Regular check-ins with developers.



Naming guideline · component hierarchy, mobile and desktop

Detailed visual handoff

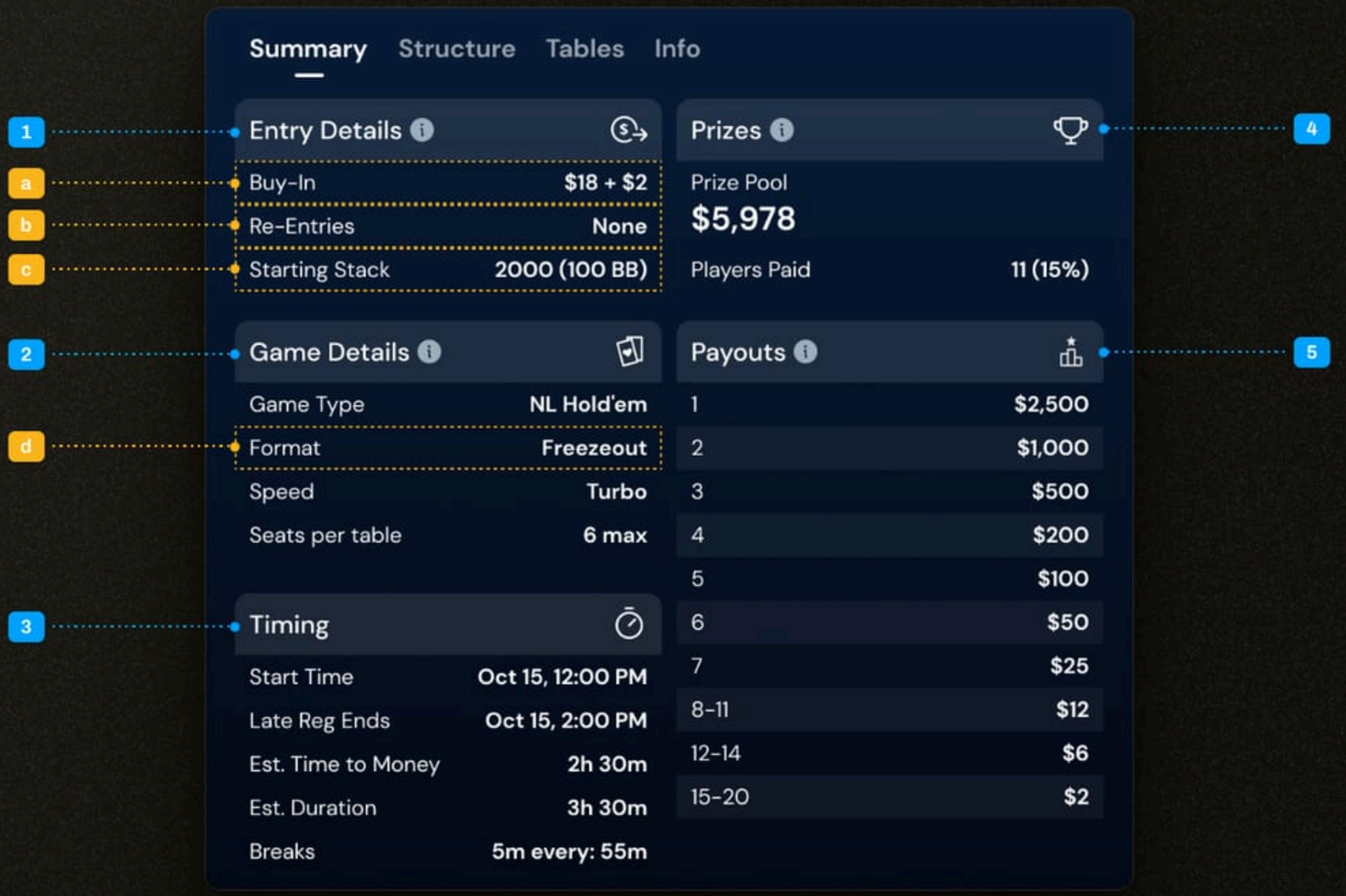
On screen

The anatomy of the Detail Summary, as handed to developers, with color-coded annotations.

- 1 1 to 5, the blocks of the component
- a a to d, the rows specified one by one

What changed

Specs were scattered across Slack, Asana and Figma. Every component, feature and interaction got a dedicated reference.



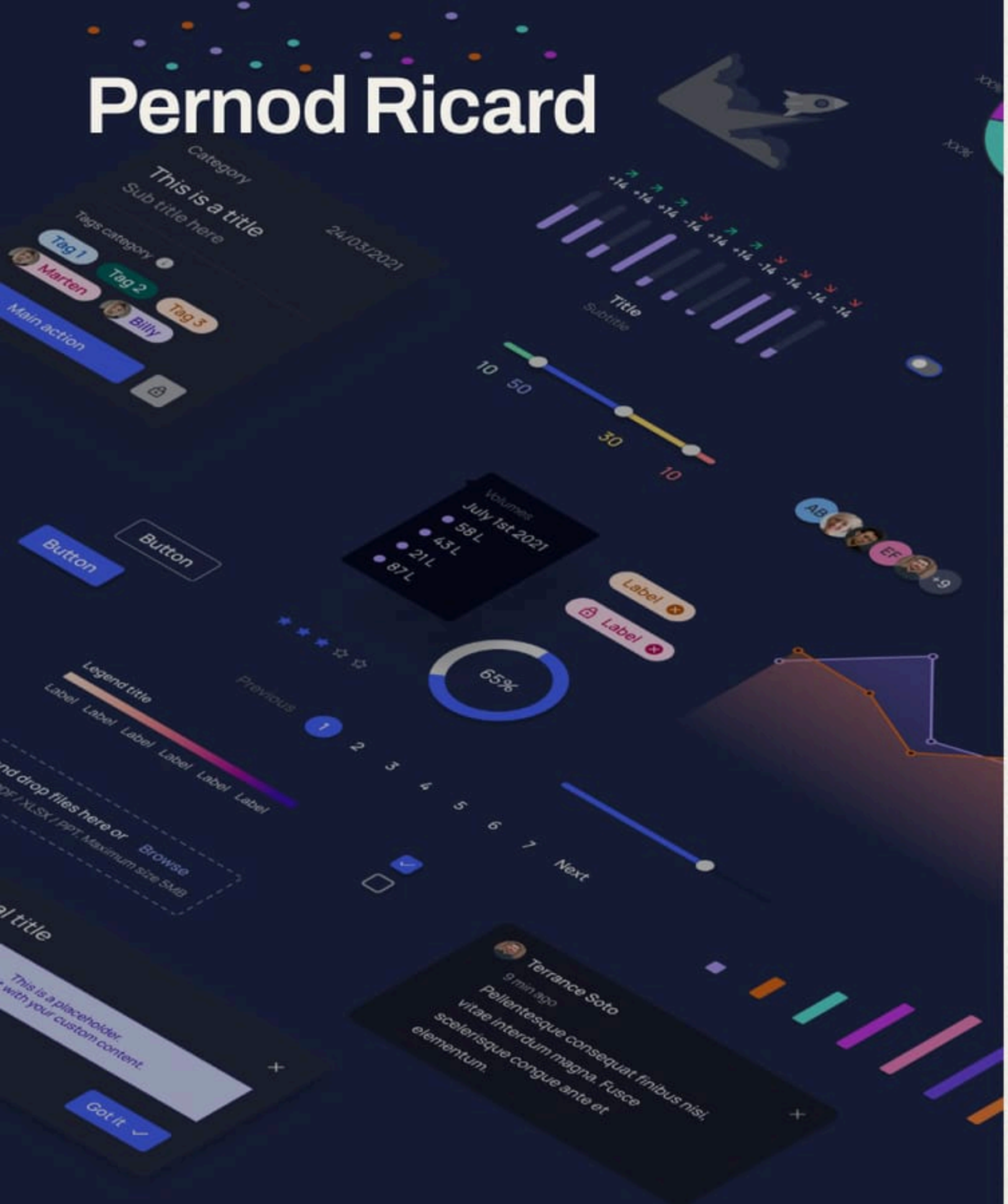
MTT Detail Lobby Documentation · Detail Summary

4 states live within 8 months of launch

Pennsylvania in October 2024, then Michigan, Delaware and West Virginia

■ Delivered	MTT Template Lobby fully adopted. Design System stabilized across 5 files, naming conventions adopted team-wide.
▨ Estimated	−90% tech feedback on handoff, −35% handoff cycle time. My estimates, not tracked figures.
□ Not resolved	No real user tracking. Accessibility never prioritized. No full merge with RSI's Design System.
○ Outside voice	Pokerfuse editorial pick: 1st US poker app at time of writing.





Case study · Archipelago, design system and SaaS

Design System Governance

Pernod Ricard is the world's #2 wines and spirits group, operating in 160+ countries. I joined the UI team to co-build Archipelago, the unified design system for 7 B2B SaaS products, and to design Extract, the first product released on it after migration.

My role	Senior Product Designer, in charge of design system and product
Crew	6 UI, 1 PO, 1 PM, 4 PRD, 6 DEV, 5 UX, 1 DA
Problem	Co-build and scale the Archipelago Design System across 7 SaaS products, establish a contribution journey for product teams, migrate the first live product from a third-party DS (Ant Design) to Archipelago, and publish external documentation via ZeroHeight.

Seven products. One system.

Senior Product Designer and design system reference for 7 squads. Led the Extract pilot to production, the AA audit and the pattern program. Co-built Le Cercle from scratch.

The library, live on Zeroheight

31

core components

One page format each: usage, do's and don'ts, anatomy, and a link to a live prototype.

11

chart types

A first-class layer, each with design and code specs.

9

pattern families

How actions, configuration, tables and comparison behave across all 7 products.

The outcomes

3 weeks

Extract pilot, design migration

The proof point sponsors needed.

7/7

products off Ant Design

Each migration faster than the pilot.

AA

WCAG 2.1 in governance

Checked on every release.

Seven products, seven islands

Same action, different solution
Filtering and editing solved differently in each product.

Every product on Ant Design
A generic framework, not the Archipelago system.

Funding on the line
Sponsors needed a first live migration to keep investing.

7 products: 6 SaaS run by their squads, and the system itself

RevUp
Retail promotion efficiency
Ant Design

Extract
A&P spend collection
The pilot

Matrix
A&P investment allocation
Ant Design

D-Star
Sales insights
Ant Design

Maestria
Portfolio strategy
Ant Design

Direct Trade
Marketplace and trade
Ant Design

Archipelago The design system, a product in its own right — DS team

Goal 01
One system
Every product builds from one library.

Goal 02
One proven migration
Extract live on Archipelago, roadmap intact.

Goal 03
One source of docs
Usable without a designer in the room.

Who decided what

DS lead + DS PM

Roadmap, priorities, staffing. A dedicated dev team built the coded library in Stencil and VueJs.

Me · Senior Product Designer, design system reference

Expert reference on components and patterns. Final calls on squad challenges, with the lead.

7 product squads

Product designer, PM and devs each. Build features, challenge and contribute.

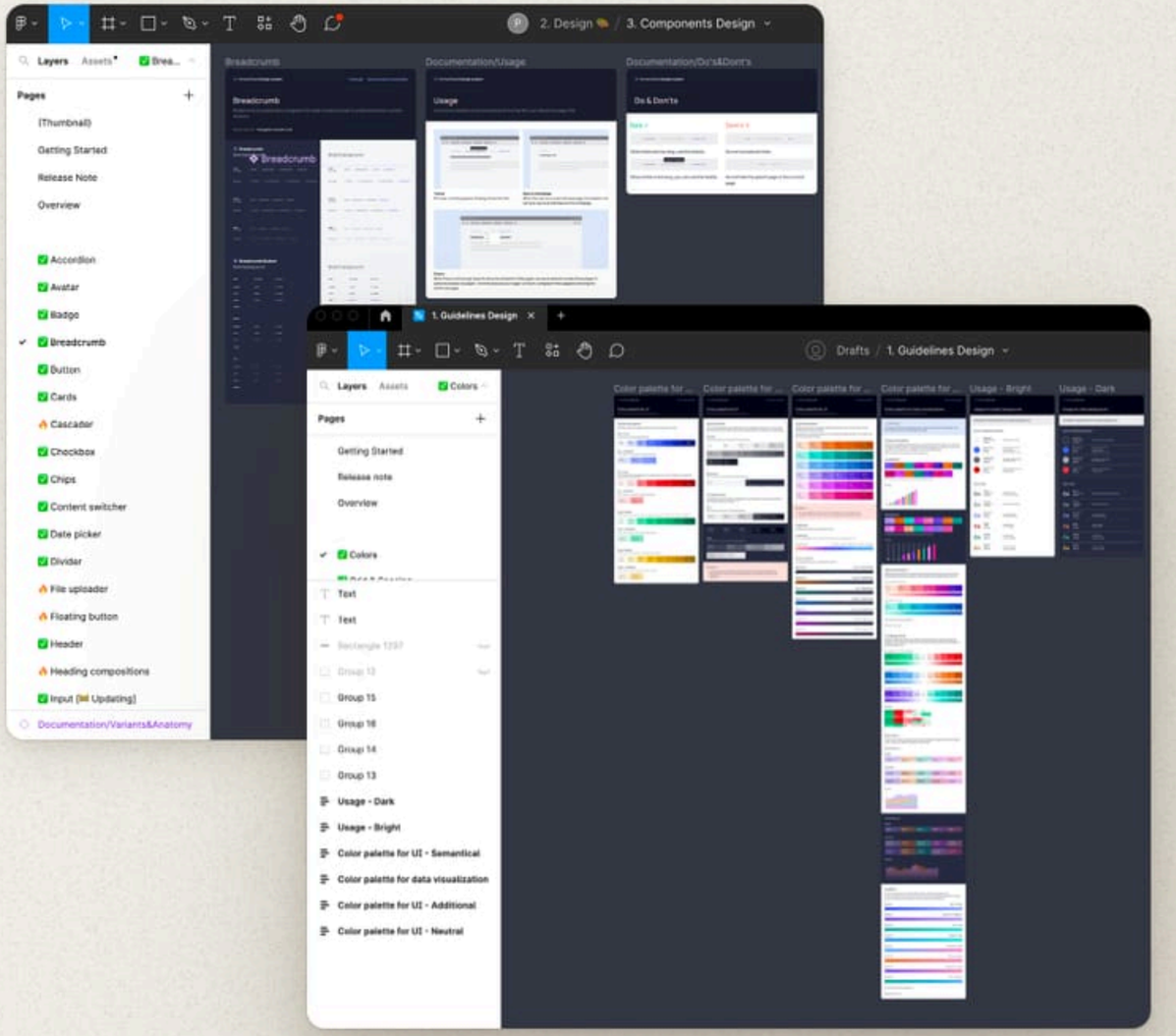
Workstream	Ownership	Level
Extract pilot migration to production	■■■	Led
Accessibility audit WCAG 2.1 AA	■■■	Led
Cross-product patterns audit of 7 products	■■■	Led
Le Cercle, B2C system from scratch, as a pair	■■■	Co-built
Component quality, docs Zeroheight	■■□	Owned
Squad arbitration with the DS lead	■■□	Co-decided
Roll-out to 6 products PD in squads	■■□	Supported
Foundations, v1 kit reworked	■□□	Inherited

Mission 01	**Building the DS B2B2C**	4 sub-missions
Mission 02	**Support product teams** migrating Extract	6 sub-missions
Mission 03	**Support product teams** product design on Extract & Maestria	5 sub-missions
Mission 04	**Create patterns and templates**	5 sub-missions
Mission 05	**Construction of Zeroheight**	4 sub-missions
Mission 06	**DS B2C design**	4 sub-missions

Building the DS B2B2C

I joined mid-build: the design system had been started by an external agency, then handed over to the UI team, which needed reinforcement because many projects were still to be done. It carried design debt: a failed merge caused by a heavy component, and no accessibility governance. The scope was to complete the system, resolve the debt, and establish contribution workflows.

Co-built and maintained	The Archipelago DS, across 6 Figma folders.
Resolved design debt	Failed merge (heavy component), accessibility.
DS contribution journey	Notion intake, sandbox, arbitration with the lead.
Onboarded new UI designers	Onto DS standards and file governance.



Guidelines and Components files, co-built with the UI team

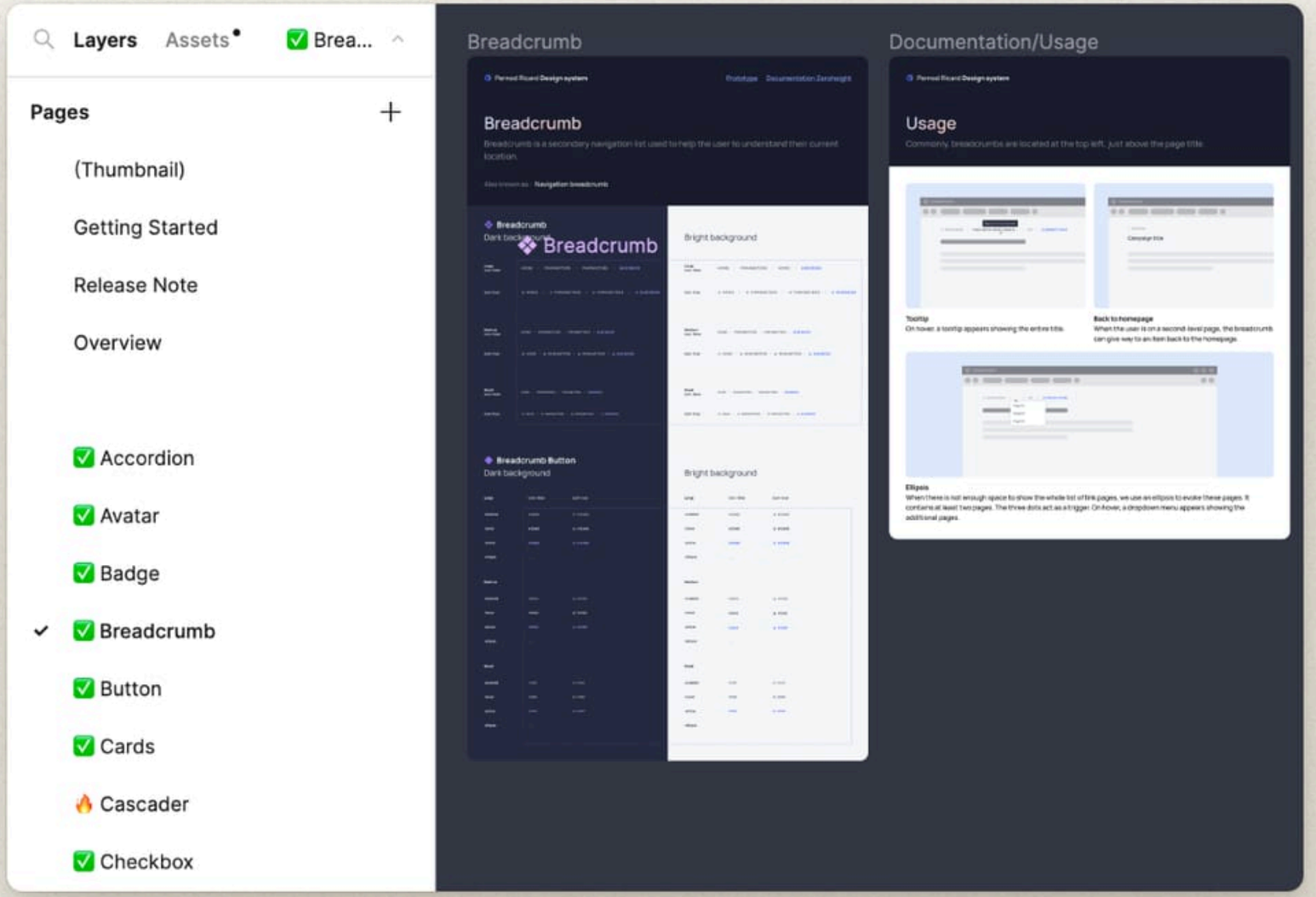
1 file, 1 page per component

On screen

The Components file in Figma: Getting Started, Release Note and Overview, then 1 page per component. Each page holds the component and its documentation, starting with Usage.

File organization

The Archipelago Design System is composed of 6 folders, co-built with the UI team. The Guidelines file opens on the same 3 pages.



Components file · pages panel, Breadcrumb page · co-built with the UI team

One token layer, by constraint

Primitives only 2022

Color, type, spacing, radius, shadow, grid and data viz palettes, as one primitive layer.

No semantic token layer: the concept was early and the products were not mature enough to name intent.

No modes: Figma could not yet switch variable modes for themes or density.

Blue 05 #F2F5FF	Blue 10 #D6DEFF	Blue 20 #96A8FA	Blue 30 #4765FF	Blue 40 #2F49F4	Blue 50 #0B2BCB	Blue 60 #01129C	Blue 70 #0C1875
Green 05 #F2FFF9	Green 10 #CDF7E4	Green 20 #72E4B3	Green 30 #2BD68E	Green 40 #13BF7A	Green 50 #07A669	Green 60 #008756	Green 70 #0A6141
Orange 05 #FFF6F2	Orange 10 #FFEEE5	Orange 20 #FFAE85	Orange 30 #FA854B	Orange 40 #E56B2E	Orange 50 #CC5418	Orange 60 #B23B00	Orange 70 #8C380E
Red 05 #FFF2F3	Red 10 #FFE5E6	Red 20 #FF8585	Red 30 #FA4B4B	Red 40 #E5312E	Red 50 #CC1E18	Red 60 #B20900	Red 70 #8C0E0E

Colors, public documentation · primitive steps and hex values

The layers I would add Today

01 **Primitive** — what existed
`red.500 = #E93D44`

02 **Semantic** — intent
`action.danger = red.500`

03 **Component** — usage
`button.danger.bg = action.danger`

Plus modes for themes and density. I ran multi-tier tokens in production with Tokens Studio at ClickOut Media, and run them in code today with Bridge.

31 core components

One page format each: usage, do’s and don’ts, anatomy, and a link to a live prototype that developers relied on at handoff.

Accordion · Avatar · Badge · Breadcrumb · Button · Card · Checkbox · Chip · Content switcher · Date picker · Divider · Header · Input · Menu · Modal · Notification · Pagination · Panel · Placeholder · Progress · Radio · Select · Skeleton · Slider · Stepper · Subheader · Table · Tabs · Toggle · Tooltip · Tree

11 chart types

Decision tools for A&P spend and portfolio strategy: charts are a first-class layer, each with design and code specs.

Bar · Line · Area · Bubble · Butterfly · Donut · Flat gauge · Progress bar · Proportional area · Icon and number · Demand map



Data Visualization file · bar chart, usage and anatomy

From Figma to production

	Stage	What it is	Mine
01	**Figma library** Design · UI team	Guidelines, components, patterns, data viz, templates, linked prototypes.	Specs, anatomy, component quality.
02	**Frozen dev copy** Handoff · DS team	A locked snapshot of what devs build. Design moves ahead safely.	Changelogs, user-story handoff.
03	**Stencil + VueJs** Code · DS devs, DS PM	The coded library, built by the dedicated DS dev team.	Behavior specs, prototypes.
04	**7 product squads** Products · PD, PM, devs	Each squad migrates off Ant Design and builds on the library.	Extract pilot, PD in squads.
	Zeroheight Documentation · public	One public doc hub across all four stages, for design and code.	Structure, content.

Support product teams

Migrating Extract

Extract, a SaaS tool for data collection and validation with a worldwide user base, had been built on Ant Design because Archipelago was not mature enough at the time. Archipelago had never been migrated to any product: Extract was the first migration, and the funding of the design system depended on it. The migration was total, and a success.

Build models from Pernod Ricard's DS

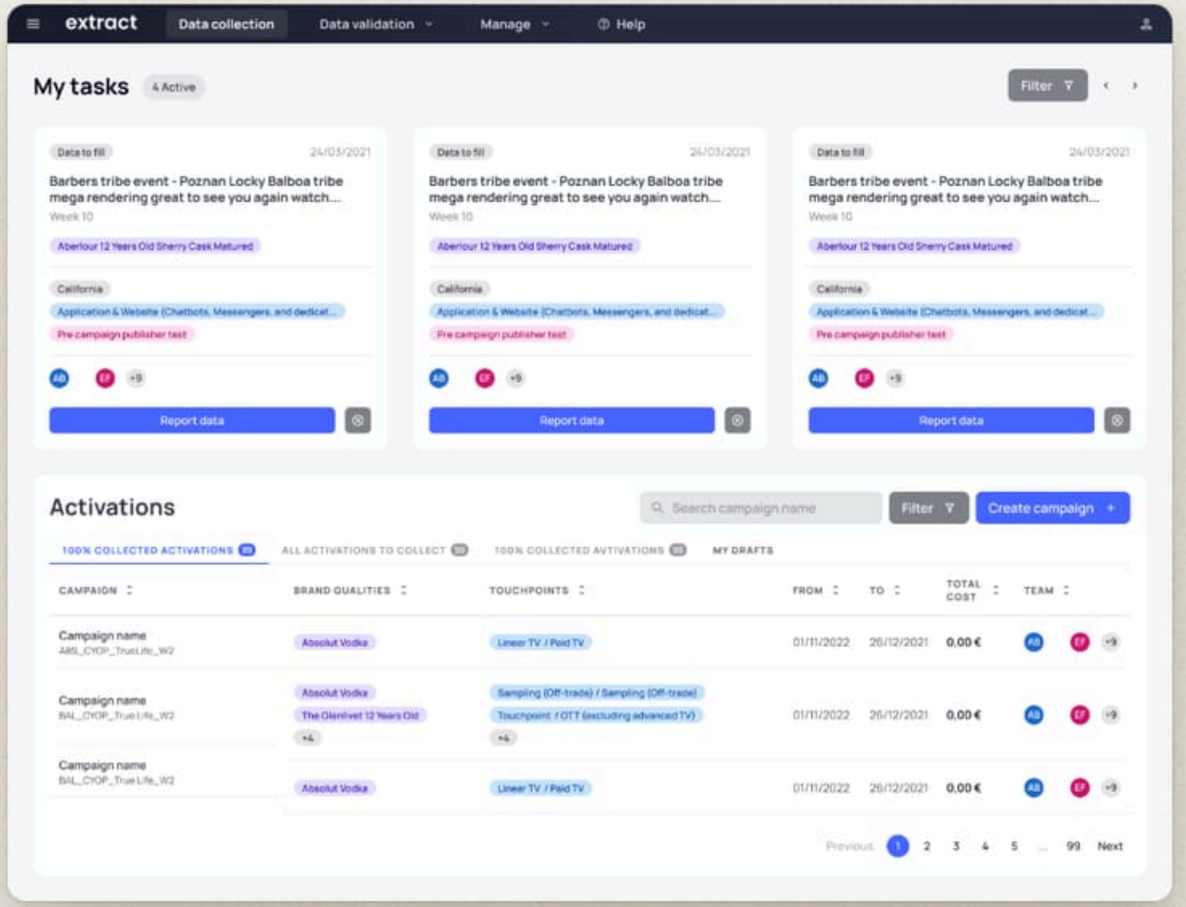
Fueling the Contribution Journey

Correct, improve or create components as needed

Document the change log at each update

Propose UX recommendations to improve the product

Ensure the Handoff

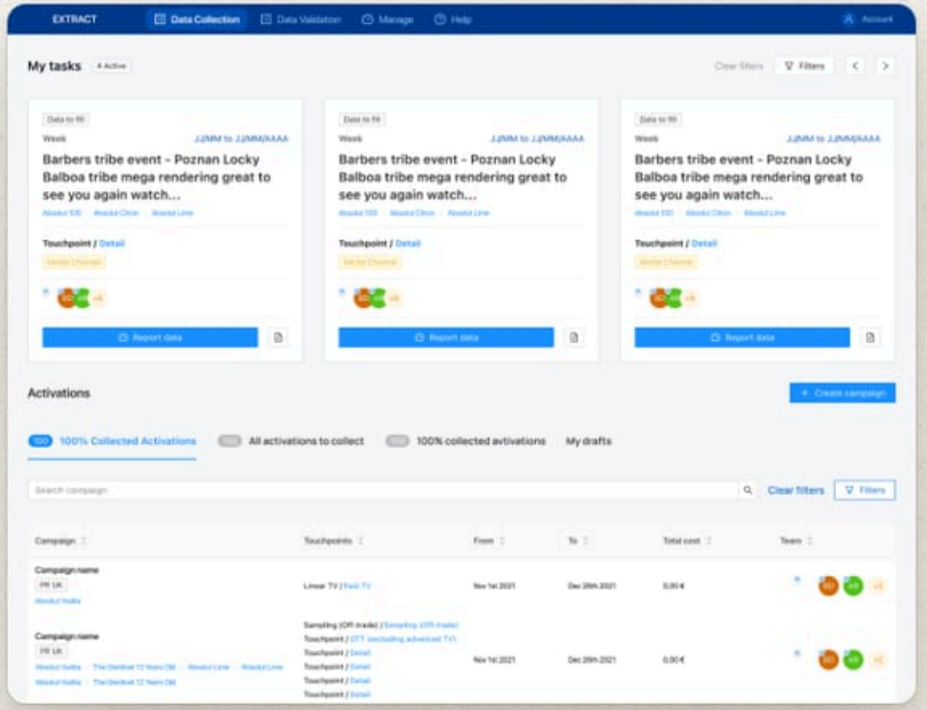


Extract · My tasks, rebuilt with Archipelago components

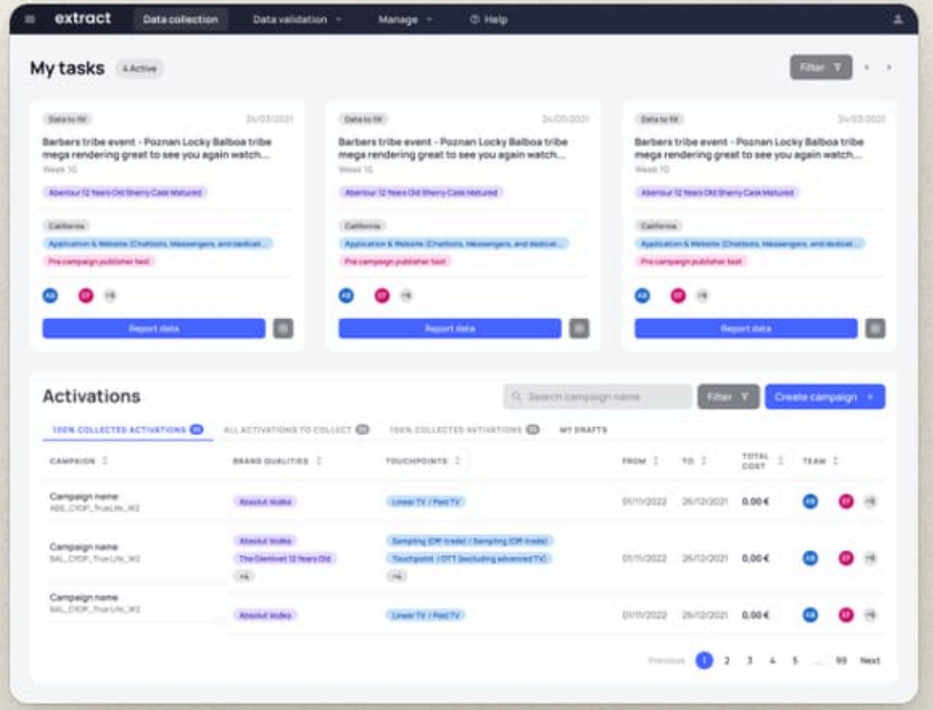
From Ant Design to Archipelago

The first product migrated to the design system, in full: the same page, My tasks then Activations, rebuilt with Archipelago components. No UX change, simply a swap of components to Archipelago. I led the design side.

Original build Ant Design · not my work



Migrated Archipelago components



Extract · My tasks and Activations, same scale

A panel, not a custom component

On screen

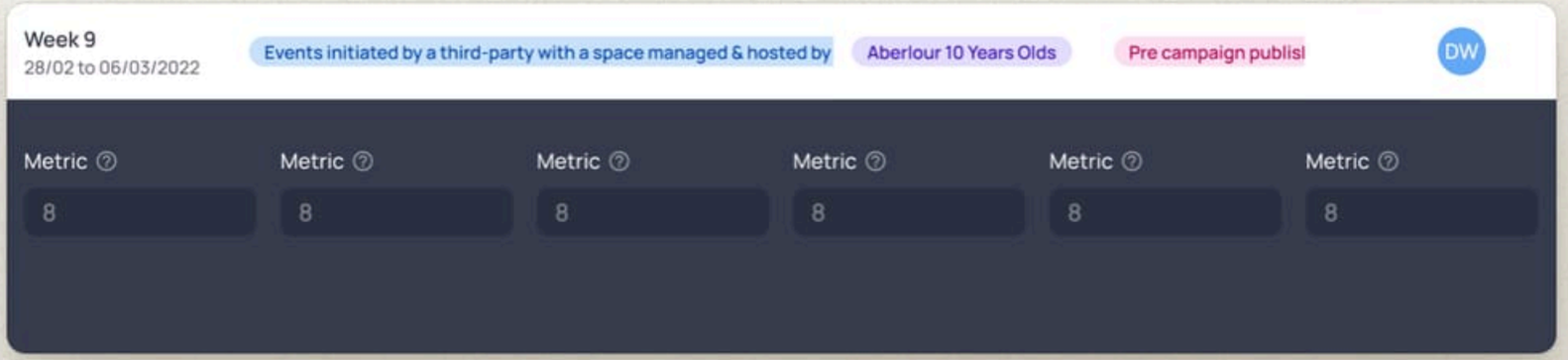
Editing data in Extract, 2 ways. ISO version: the row unfolds inside the table. Target version: a panel opens when the user clicks on a row.

What changed

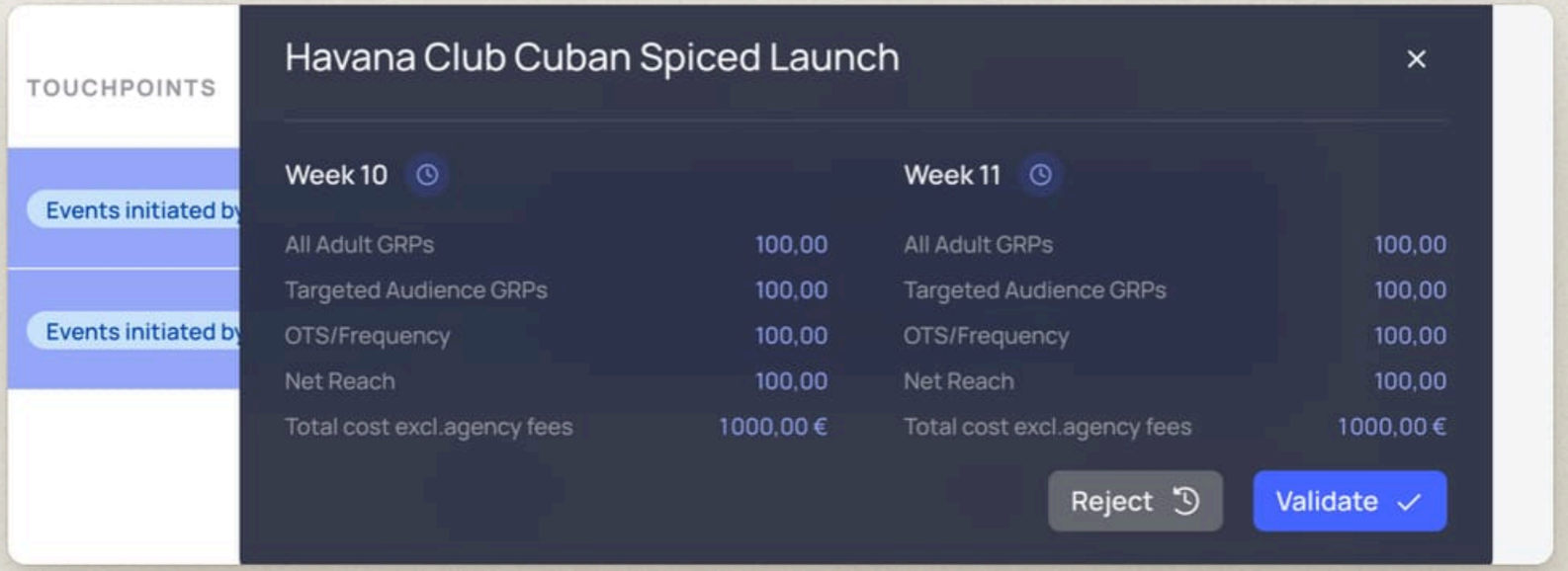
Extract wanted a dropdown behavior integrated into a basic cell. The DS lead and I arbitrated: no custom component, a pattern instead. I designed it: the panel.

Then

The whole target version was developed, then extended to all 7 SaaS products once the behavior, action and interaction patterns were in place.



ISO version · the row unfolds inside the table



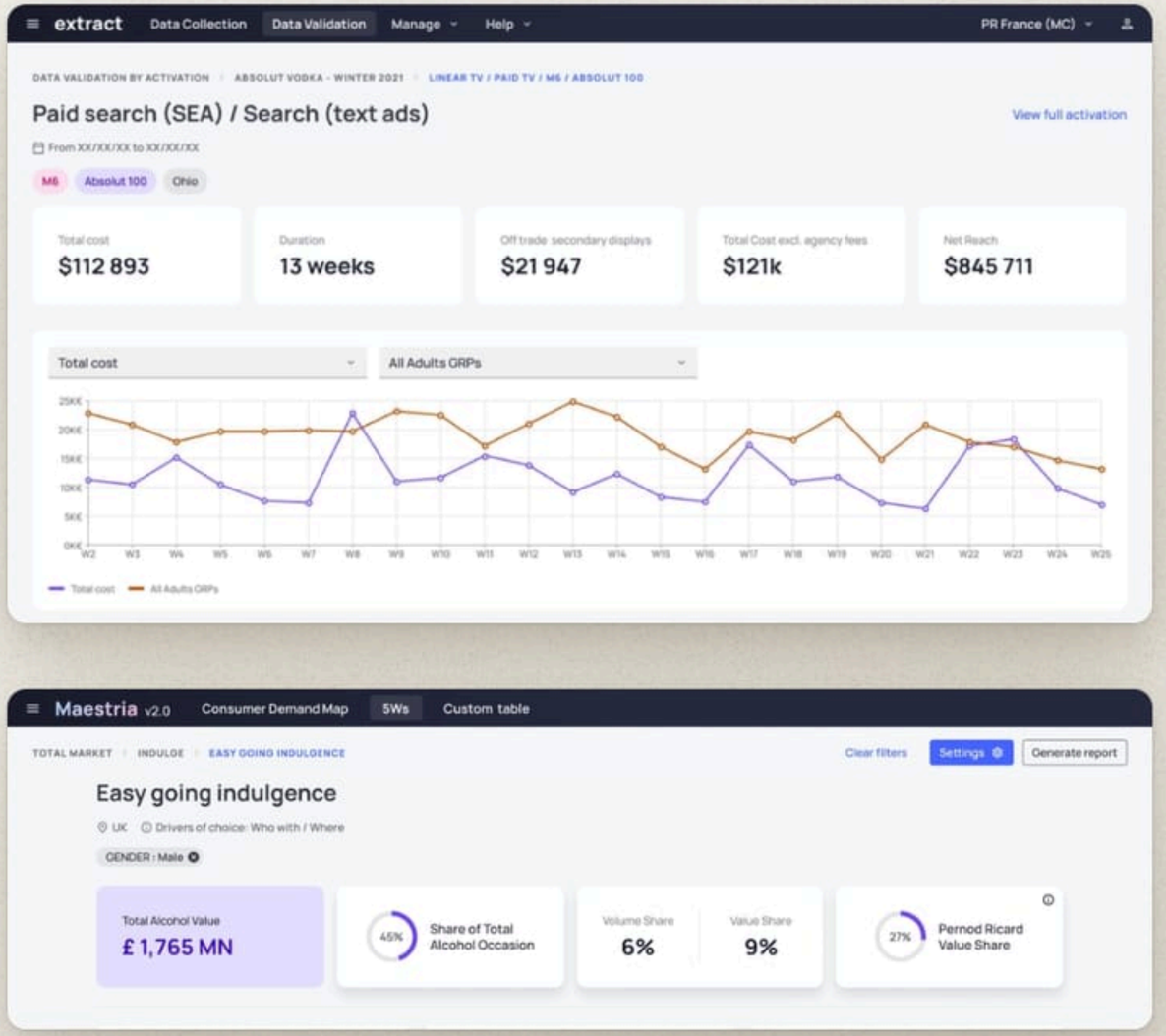
Target version · a panel opens beside the table

Support product teams

Product design on Extract & Maestria

Alongside Archipelago, I worked directly with the Extract and Maestria product teams on data-heavy B2B products used across Pernod Ricard's international markets. With Product Owners, engineers and the UXR team, I designed new capabilities, restructured critical workflows and validated decisions with contributors from key markets.

New features & evolutions	Owned the design across Extract and Maestria.
Critical workflows	Redesigned with Product, Engineering and UXR.
Iterative research	Improved data quality and completion rates.
Business rules	Translated into production-ready UX and specs.
Aligned with the DS	New product patterns kept aligned with it.



Extract · Data validation, then Maestria · 5Ws, 1 occasion

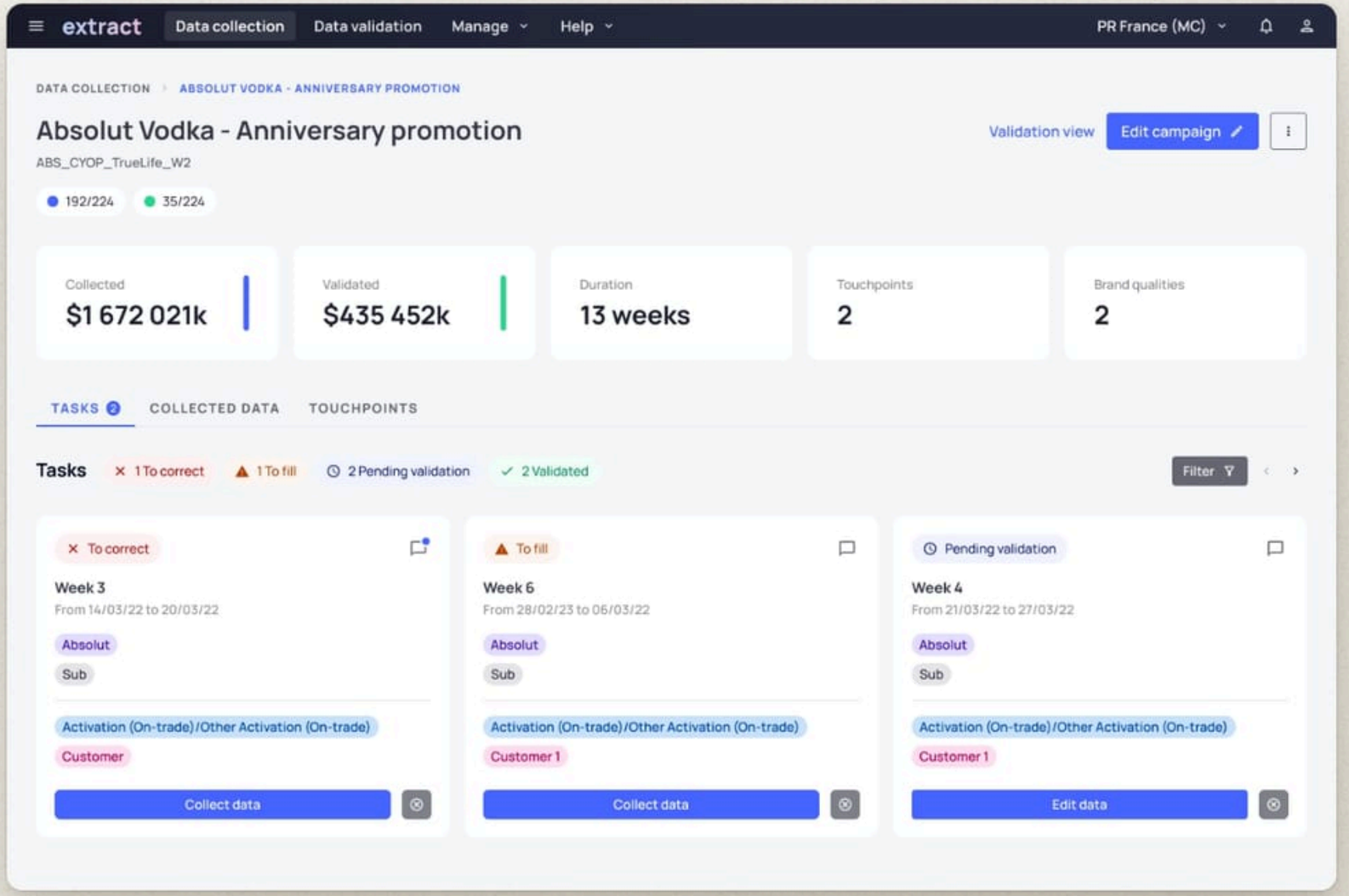
Campaign progress, at a glance

On screen

Extract, the Absolut Vodka campaign: collected and validated spend, then tasks week by week, each with its status: to correct, to fill, pending validation, validated.

What changed

Contributors have little intrinsic incentive to spend time in the tool. I designed clearer workflow and validation states, to make progress and validation requirements easier to understand for contributors and validators.



Extract · Data collection · campaign tasks

Three calls that shaped the system

Patterns, not one-off components

Extract asked for a dropdown inside table cells.

We refused a component for 1% of cases and shipped an edit panel pattern instead.

Trade-off	A less literal answer for one product.
Gain	One edit behavior for all seven.

User stories, not every screen

The team wanted a mockup for every Extract state.

I scoped handoff to key screens per user story instead.

Trade-off	Fewer static specs.
Gain	Faster iterations, developers worked from flows.

An ISO version and a target version

The PM feared UX changes would slip the roadmap.

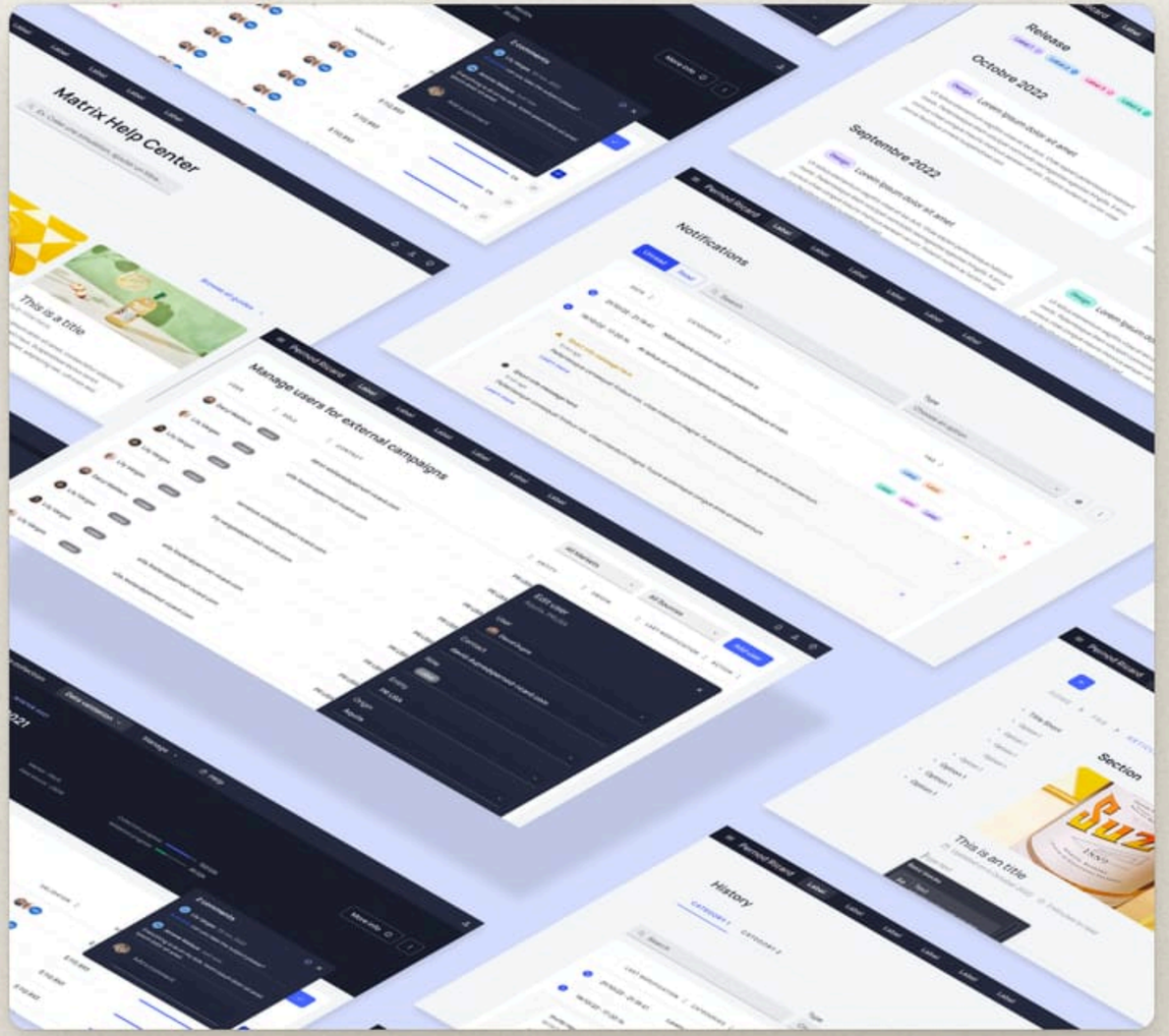
I split the work: a pure component swap, plus the recommended UX next to it.

Trade-off	Two files to maintain.
Gain	On-time migration, and the target fed cross-product patterns.

Create patterns and templates

Pernod Ricard's 7 SaaS products shared common patterns (data filtering, campaign creation, data editing), but each product designer had built an independent solution for the same action. I led the harmonization of patterns and page templates with the UI team. Once in place, they carried Archipelago to all 7 products.

Inventory	All existing patterns, on the existing flows.
Benchmarking	Of the competition, key screen by key screen.
New patterns	Assembled into 5 flows.
Page templates	Generic pages common to all products.
Evangelism	Presented to the designers, iterated by vote.



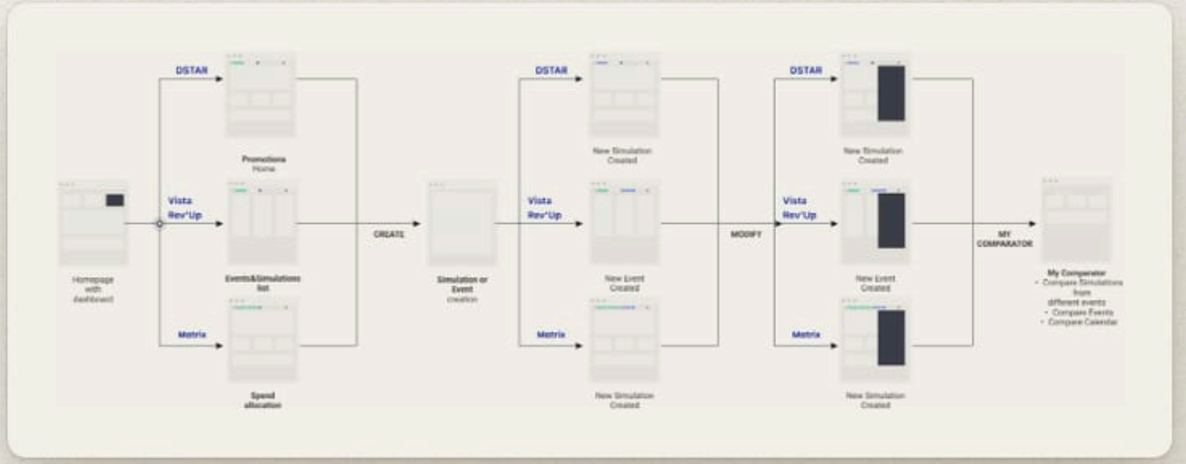
Page templates · presentation visual

Two kinds of patterns, one review loop

Action patterns Type 01

The same task, done the same way everywhere.

01	**Audit**	Live journeys of 7 products, every pattern listed.
02	**Homogenize**	Flows doing the same action grouped.
03	**Template**	Create, edit, filter, compare, manage.



Action patterns, step 02 · 3 of the 7 products shown

Page templates Type 02

Pages every product needs, designed once.

01	**Benchmark**	Best-in-class products, screen by screen.
02	**Zoning**	Quick freehand zoning of each page.
03	**Mockups**	Notifications, history, access, FAQ…

Both types

Review, vote, iterate

Presented to each product's designer for feedback. Votes in Figma set what ships first.

How it behaves, not just looks

9 interaction pattern families, documented

Status indicators
Semantic colour, icon and label, and which component carries the state

Common actions
16 verbs, each with its own rule

Navigation
3 priority levels, 3 directions, object-oriented

Configuration
Selection, filtering, settings, forms, partial or full-screen panels

Simulation
2 levels; the user stays the final decision maker, not the AI

Notifications
Task-generated or system-generated

Tables
Add a line, edit a row or a single cell inline

Comparison
Attributes grouped to compare before deciding

Modularity
Dashboards built from bricks

Components make products look alike; patterns make them work alike.

Action patterns

16 verbs, one rule each

Add	Export
Add to	Manage
Apply	Remove
Cancel	Reset
Create	Run
Delete	Save
Duplicate	Save as draft
Edit	See

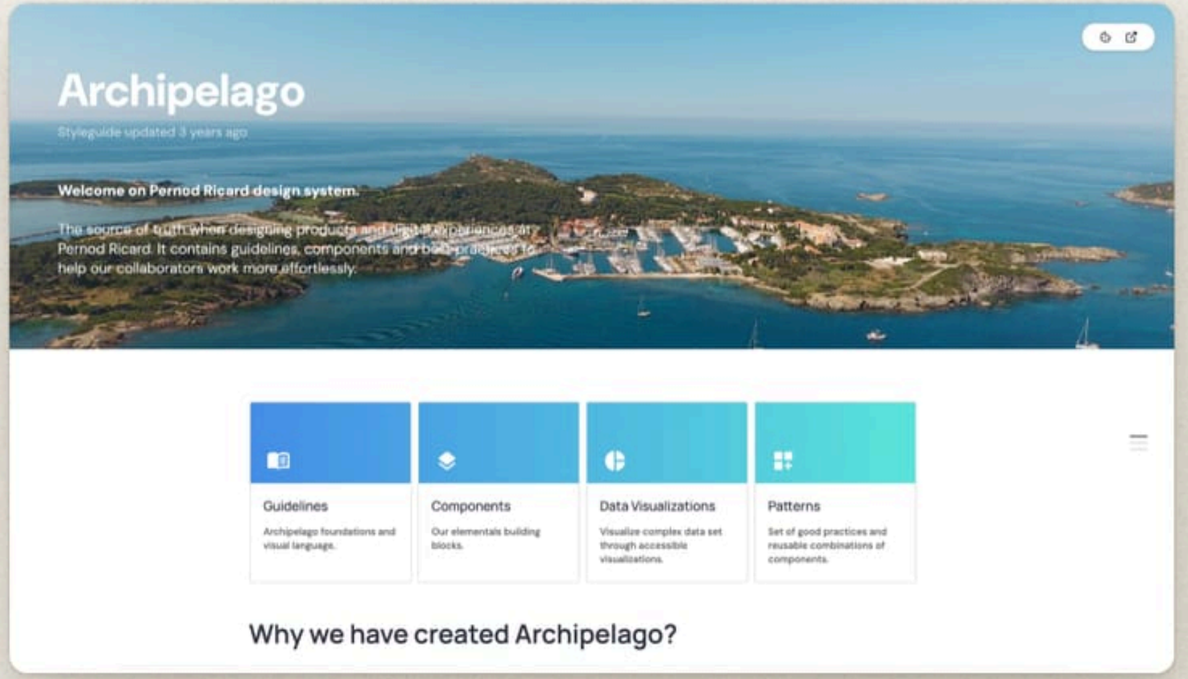
Create opens a creation funnel; every later change is **Save**.

Delete always uses the danger button and asks for confirmation.

Construction of Zeroheight

To communicate its design system internally and externally, Pernod Ricard needed a documentation source presenting guidelines, patterns, components and best practices. I built and published the Zeroheight hub. It has been released externally.

Page content	Structure and content of the documentation.
Illustrations	Built in a dedicated Figma file.
Animated gifs	Animated demonstrations of the components.
Patterns	Defined and added to the documentation.





Guideline name	Design	Stencil	VueJs
Brand assets	–	–	–
Colors	✅	✅	✅
Grid & Spacing	✅	✅	☐

Zeroheight · home, then Guidelines status per platform

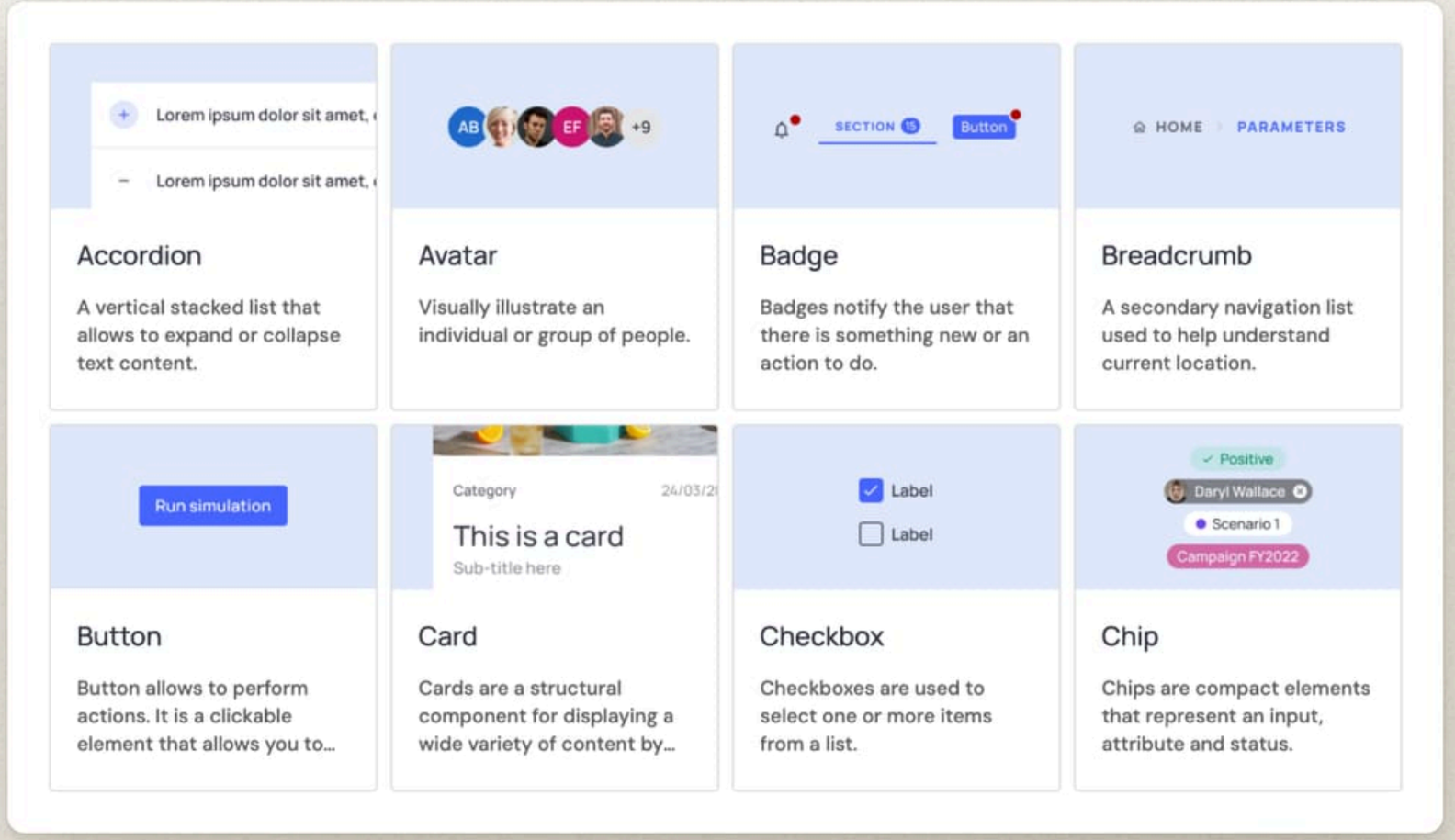
The documentation went public

On screen

The public Archipelago documentation, page All components: 1 tile per component, with its preview and its definition. Here the first 8.

What changed

Pernod Ricard needed a documentation source to communicate its design system internally and externally. The Zeroheight has been released externally.



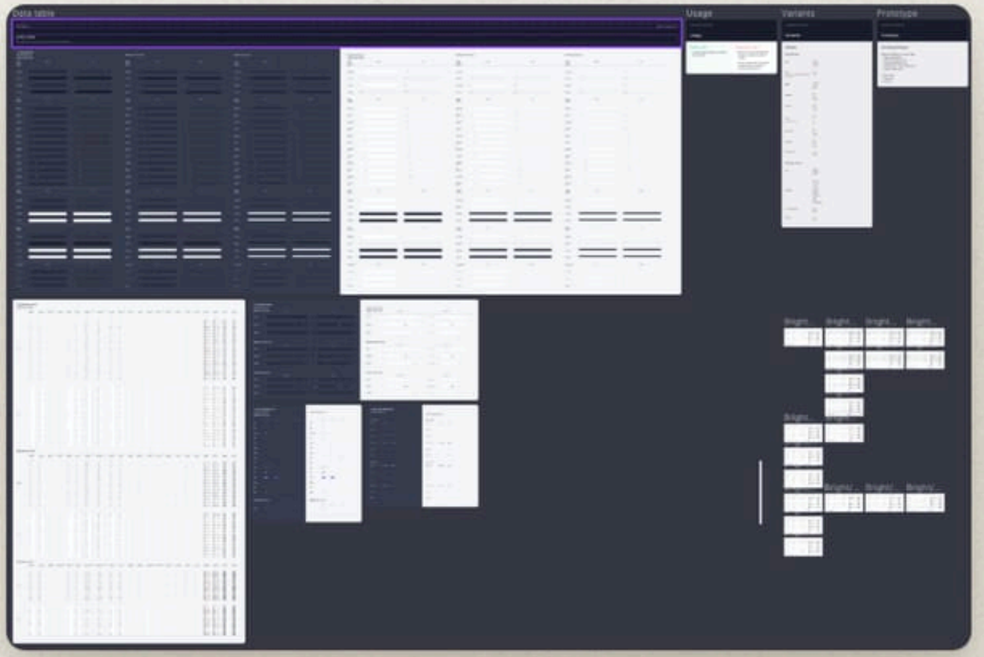
My content in Zeroheight, a third-party tool · captured Nov 2026

When an update broke trust

Problem	Figma released component properties. We refactored to cut variants, and the refactor broke components inside live product mockups.
Impact	Product designers stopped accepting library updates. A design system nobody updates is a design system in decline.
What we changed	We limited structural edits, and for the major ones we repaired the broken components inside each product ourselves.
Lesson	**Adoption is a trust metric.** The system team absorbs the cost of change, never the product teams.

When 6 weeks of work would not merge

Problem	UI designers worked in one child branch, released at developer pace. After 1.5 months of changes, Figma could no longer merge it.
Cause	The data table. Given the number of possible combinations of variants, we had to wait 10 seconds for each change.
Impact	Product designers waited for fixes already designed. A second library had to be created and every product mockup relinked by hand.
What we changed	The table restructured: unnecessary variants removed, divided into 3 Figma pages. And a locked copy of what is coded for developers.
Lesson	**Version design and code separately.** Drift is cheap to prevent and expensive to repair.



Components file · Data table page, before the restructure

Accessibility, one year late

Users could not tell some component states apart. Target: WCAG 2.1 AA. Four designers, one file.

Table | Work in Progress | Review | Changelog +

Fail 38 ··· +

Aa Component	Complia...	Obj. vs Bg	Texte	Obj. vs Bg	Texte	État
[Dataviz] Axis	Fail					Updating
[Dataviz] Colors	Fail					First Review
[Dataviz] Legend	Fail					Updating
Accordion	Fail	< 3:1	< 3:1	4.5:1	4.5:1	Figma ✅
Avatar (Initials)	Fail	< 3:1	< 3:1	< 3:1	< 3:1	Figma ✅

What changed

Accessibility stopped being a project and became part of every release.



Audit table in Notion, contrast ratios per component · then the validation workflow I set up, applied to every component

Consider color deficiencies

Seen with deuteranopia, Curaçao is confused with Aqua and Blueberry: we deleted it.

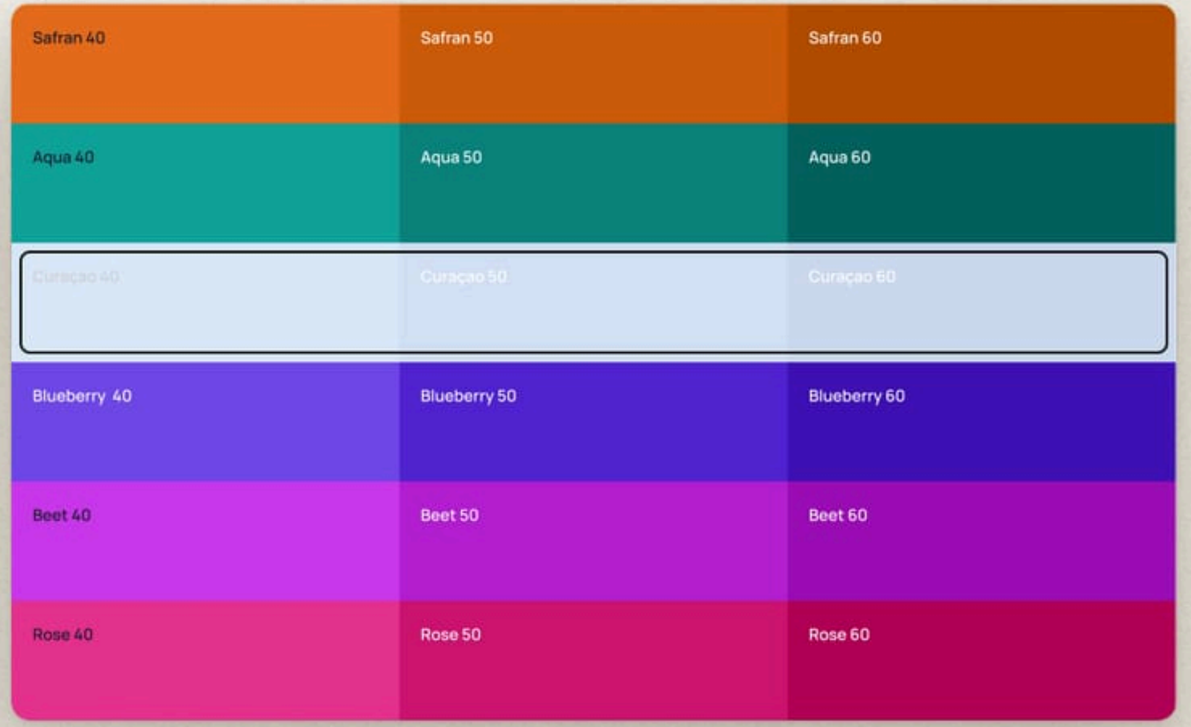
Deuteranopia

Red-green confusion · 3 rows look alike



The palette

Curaçao, deleted



Guidelines file · Consider color deficiencies · co-built with the UI team

How a change gets in

Arbitration is the key step: most requests became patterns, not new components.

3 rules, born from the challenges

No component for 1% of cases
Product-specific needs become patterns or stay in the product.

No breaking change without a path
Structural edits ship with a migration, or not at all.

Devs build from a frozen copy
Design moves ahead; code never chases a moving file.

01	**Request**	Need logged in the contribution journey
02	**Discuss**	Feedback and errors in the Figma sandbox
03	**Arbitrate**	DS lead and I decide: system, pattern, or product?
04	**Build**	In a branch, every edit in a changelog
05	**Review**	Peer review, then accessibility check
06	**Release**	Release note, dev copy, Stencil and VueJs release

DS B2C design

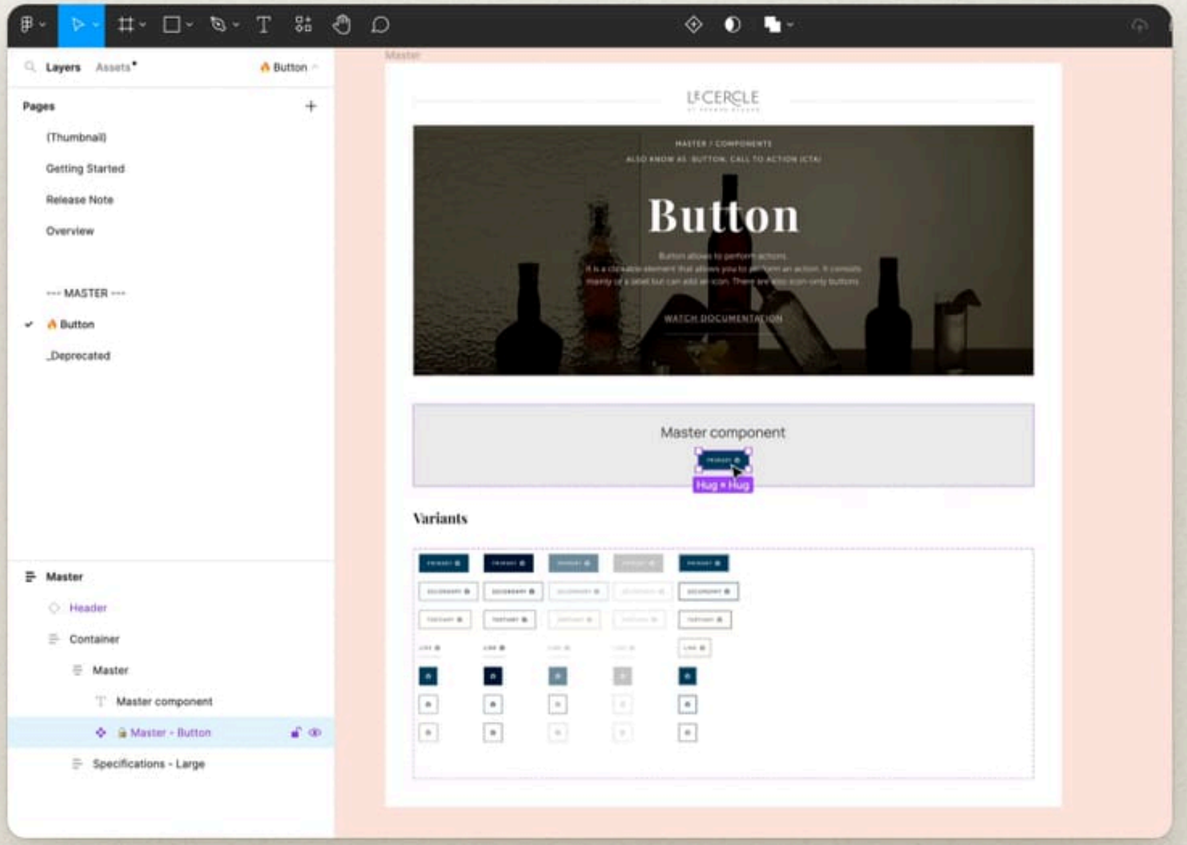
Le Cercle is Pernod Ricard's B2C site for its premium customers. Its graphic charter is based on the codes of luxury, so Archipelago could not be used. The site was already online, run by another design team working in Sketch. Co-built as a pair with another designer, the design system was made from scratch in Figma and handed off to the Hong Kong design team. It fixed Archipelago's pains: master components update every variant at once, docs captions linked to components.

Audit of the site and its graphic charter

Construction and structure of the Design System

Creating Components

Documentation and Evangelization



Le Cercle design system file · Button, master component and variants

Who needed what, and how I served it

Partner	What they needed	How I worked with them
Product squads PD, PM, devs	Answers, not waiting	PD in squads, reviews, final calls
DS dev team and DS PM	Stable specs	Linked prototypes, frozen files, changelogs
PO and PM	No roadmap risk	ISO vs target screens, user-story scope
UX research	Market validation	Interviews and tests with key markets
External dev agency	Buildable specs	Production-ready flows for Maestria
New UI designers	Fast ramp-up	Onboarding on DS standards
Hubvisory consultants	DS literacy	Training sessions I designed and ran

How I start in a new team

One week to listen. Then a loop.

Week 1

Listen, measure

Adoption and drift per product, an AA baseline, talks with squads and DS devs.

Weeks 2 to 3

Fix the flow

A clear path from squad request to release, agreed with engineering.

Then, in a loop

Ship, measure, iterate

Run the DS as a product: track adoption, contribution lead time and AA pass rate, improve each release.

Every call balances

End-user needs + Business strategy + Tech constraints

Key result · Archipelago, Pernod Ricard

7 SaaS products on Archipelago

Extract first, migrated in full. Then all 7.

- Extract migrated in full and released internationally, validating the system for the entire digital transformation program.
- The whole target version developed, then extended to all 7 SaaS products. The program kept its funding.
- Zeroheight, the documentation source, released externally.
- Accessibility audit completed and integrated into DS governance after 1 year of debt accumulation.
- Extract's data quality and completion rates improved through iterative research. Maestria's data-visualization experience shipped live.
- Le Cercle B2C DS handed off to Hong Kong design teams.



At Pernod Ricard, by hand		Now, in Bridge
Six weeks of drift, then a failed merge	→	Each exploration in its own copy, reviewed change by change
Needs collected in Notion, then a sandbox	→	New tokens, variants and components listed automatically
Accessibility checked by hand, component by component	→	Contrast checked in light and dark before approval
Figma and code drifting apart	→	Figma kit generated from code, with a fidelity check
Documentation written by hand	→	Docs drafted from code by AI, reviewed by the DS lead

The problems I fought on Archipelago, by hand, with a 24-person program.

Bridge sits between AI explorations and the design system, so what works comes back in, reviewed.

Bridge

Founder & Design Engineer

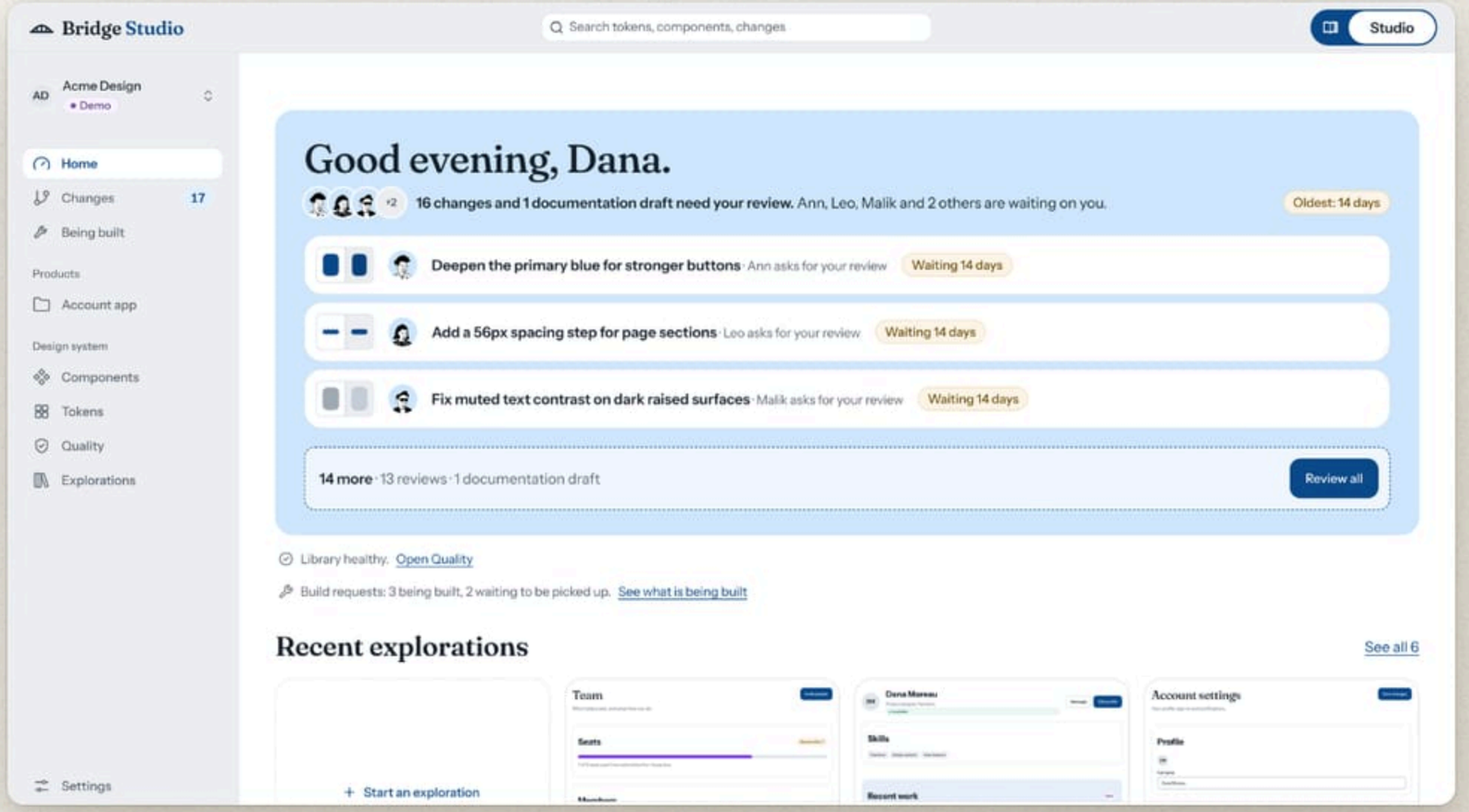
Governance for design systems, now that every team explores with AI.

Teams prototype with AI outside the design system. Bridge spots what each exploration adds, the DS lead reviews it, then code, Figma and docs update together.

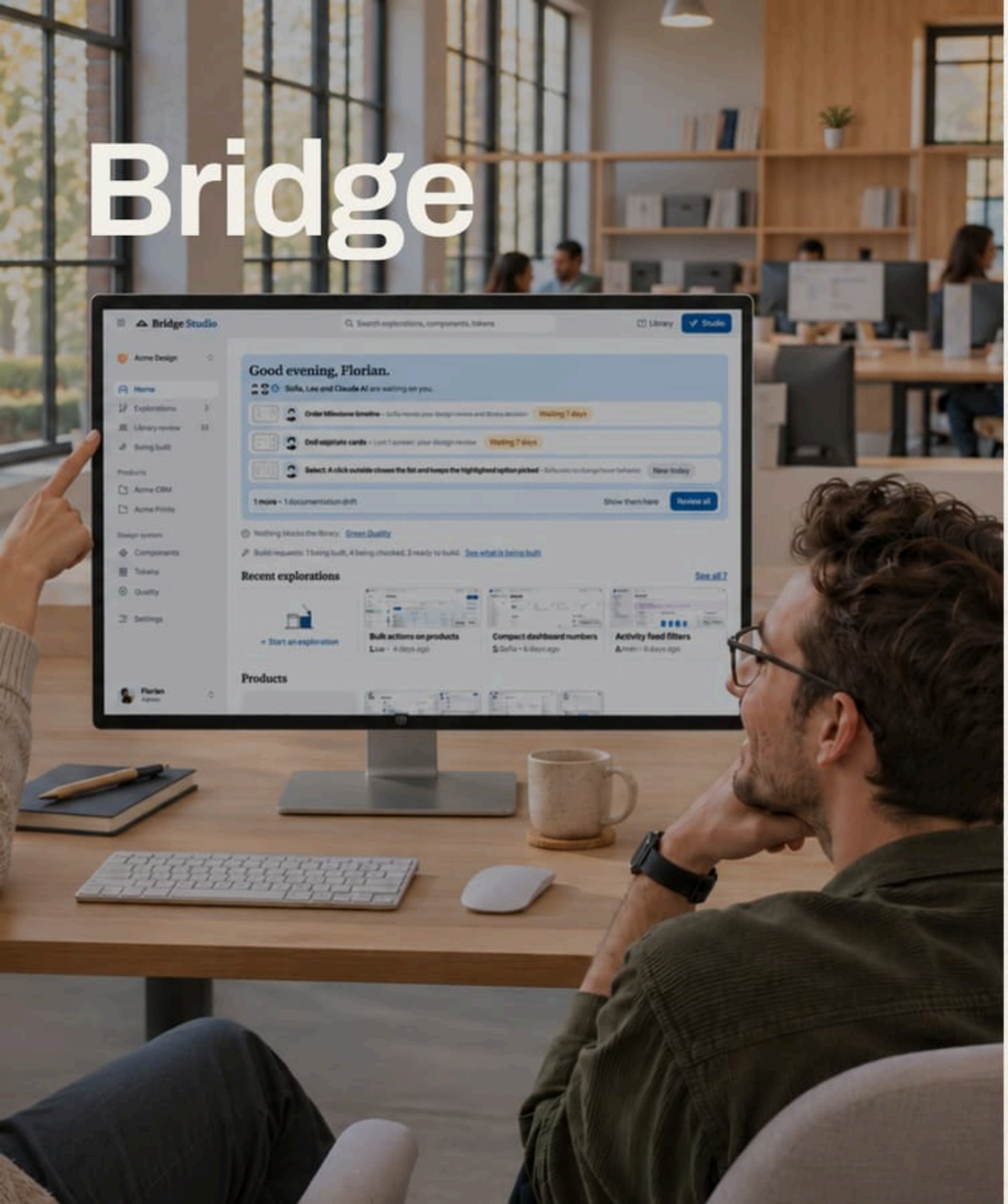
I built every part

- The Studio
- The React library, with tokens
- The Figma plugin
- The Claude connector

bridgedesignsystem.com



Bridge Studio · home of the demo workspace, the DS lead review queue



Case study · Bridge, own product, built in public

AI exploration platform for product teams

Every product team I spoke to in 2026 had started exploring with AI, and none could tell me why they kept what they kept. The exploration happened in a chat window, the good idea was copied by hand into the product, and the reasoning stayed in someone's head. Bridge closes that gap: you explore with AI in whatever tool you already use, Bridge tracks what the exploration changed, you review it, and the code, Figma kit and docs follow.

My role	Founder & Design Engineer
Crew	Solo, building with AI agents
Problem	Teams explore with AI faster than they decide.

Mission 01 — Establishing the problem before building anything — 5 sub-missions

Mission 02 — 5 pains, and the workaround that proves each one — 6 sub-missions

Mission 03 — What the corpus contradicted — 2 sub-missions

Mission 04 — Designing the product — 3 sub-missions

Mission 05 — The design system it runs on — 1 sub-mission

Establishing the problem before building anything

A product built on its founder's own frustration is a guess with better lighting. So the first thing I built was not a screen. It was a corpus, kept as data, one row per source.

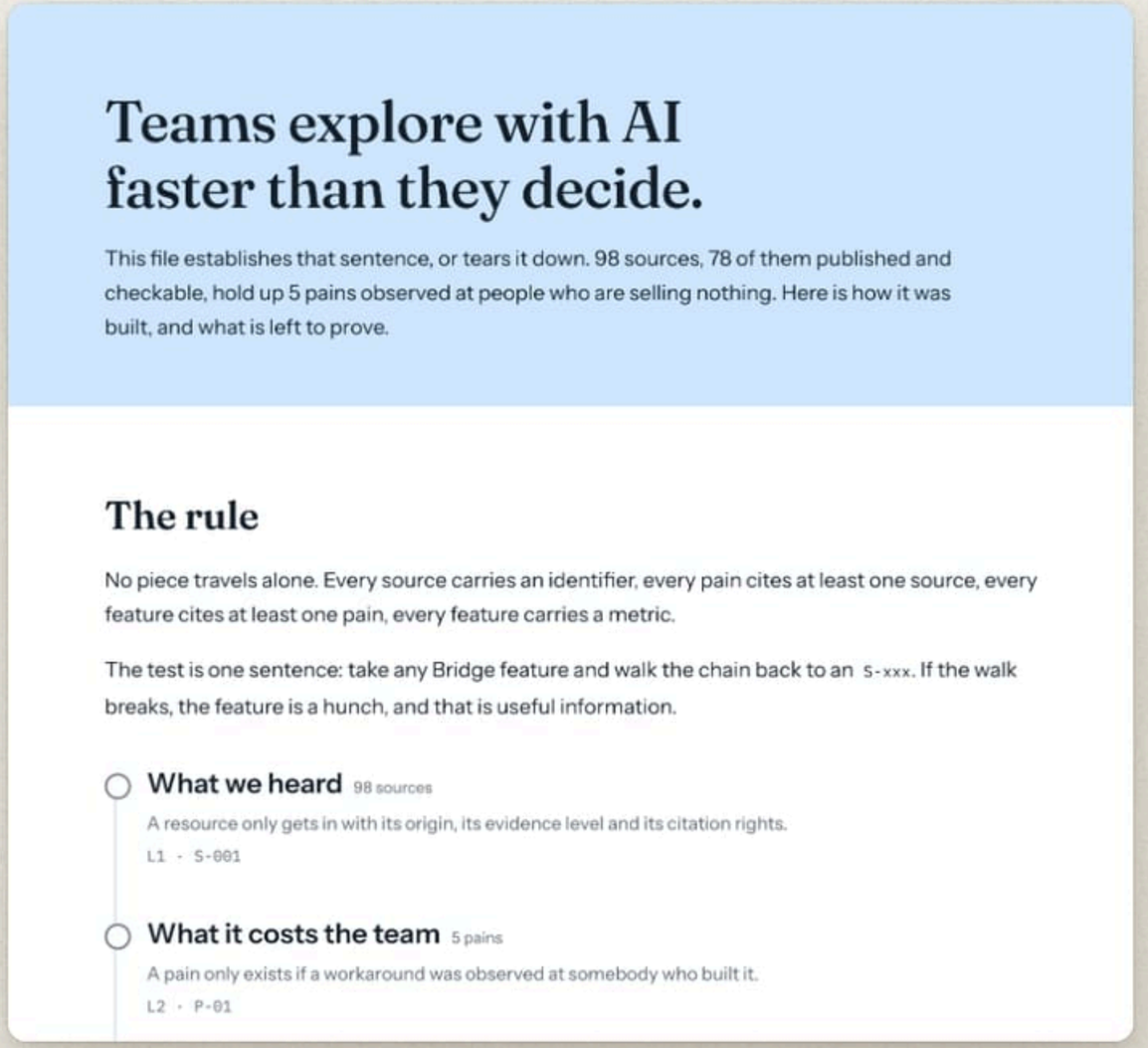
Why a workaround and not an opinion
No pain kept without the workaround people had built.

The pain the rule kept in
Fewest sources, strongest signal.

The channels, marked on what they returned
44 LinkedIn, 38 open web, 11 my tracker, 5 direct.

The four evidence levels
Read before any source is cited.

What this research does not prove
4 capabilities rest on judgement, not evidence.



Research file · The method

Where the 98 sources came from

Channels · measured, not estimated

LinkedIn posts and comments	44
Open web: Hacker News, studies, vendor docs	38
My own tracker: job specs, interviews, calls	11
Direct exchanges: calls, private messages	5

Evidence levels · read before a source is cited

A Published: study, report, article	78
B Written first-hand: message, job spec	16
C Recalled from memory: a lead, not proof	3
D Inferred: to test, not to cite	1

Four fifths is public and checkable. 79 can be named, 17 are anonymised, 2 stay private.

5 pains, and the workaround that proves each one

A pain only exists if a workaround was observed at somebody who built it.

Explorations never come back to the system
35 sources, the best documented.

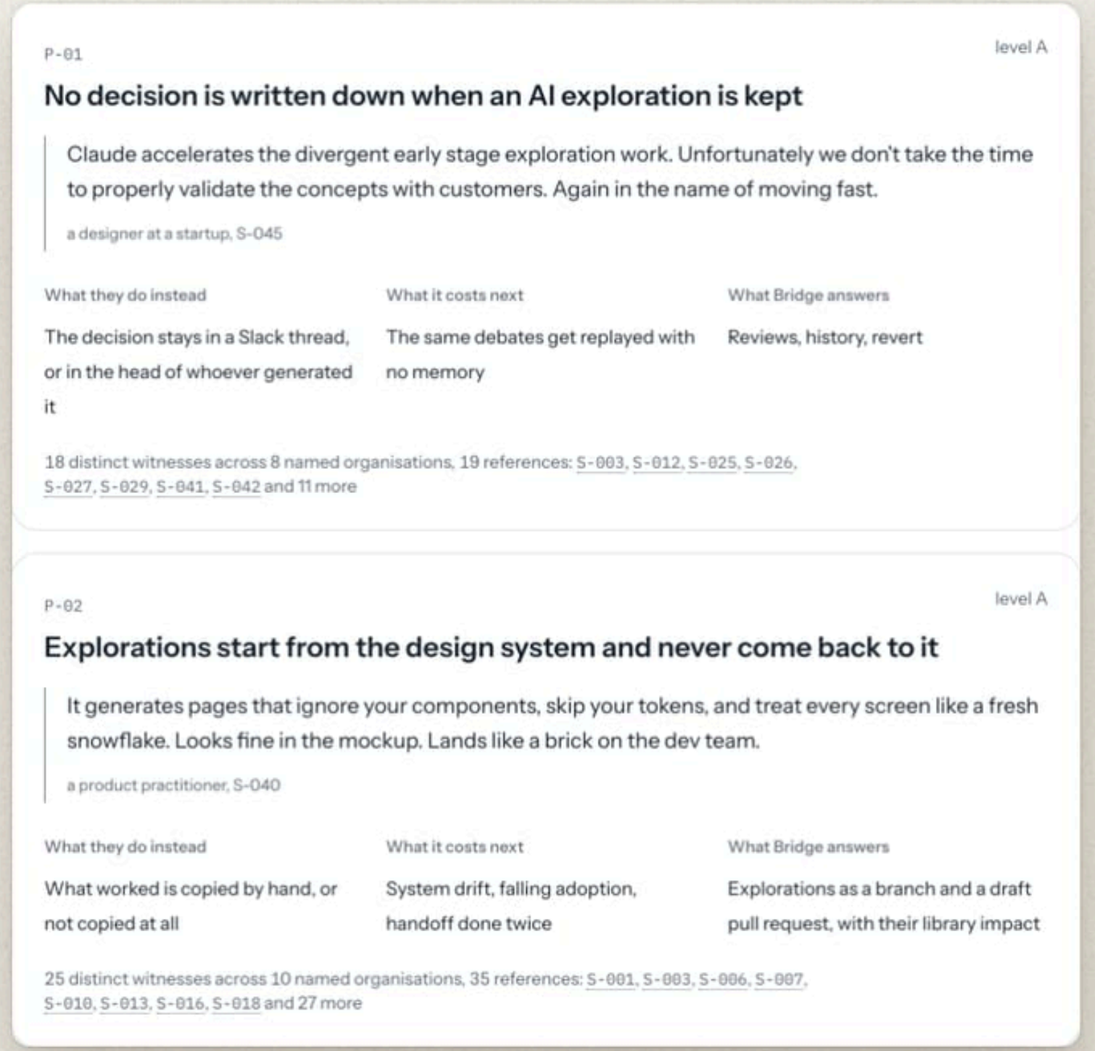
No decision is written down
Argued in a place built for reviewing code.

Nobody knows who approved what
19 sources. The cost surfaces at the moment of the bug.

Volume outruns review capacity
The review work moves to the structure.

Six teams that had already built it themselves
6 sources, and every one had built the tool.

What that means for what Bridge has to beat
An afternoon of internal tinkering, not a vendor.



P-01 level A

No decision is written down when an AI exploration is kept

> Claude accelerates the divergent early stage exploration work. Unfortunately we don't take the time to properly validate the concepts with customers. Again in the name of moving fast.
>
> a designer at a startup, S-045

What they do instead	What it costs next	What Bridge answers
The decision stays in a Slack thread, or in the head of whoever generated it	The same debates get replayed with no memory	Reviews, history, revert

18 distinct witnesses across 8 named organisations, 19 references: S-003, S-012, S-025, S-026, S-027, S-029, S-041, S-042 and 11 more

P-02 level A

Explorations start from the design system and never come back to it

> It generates pages that ignore your components, skip your tokens, and treat every screen like a fresh snowflake. Looks fine in the mockup. Lands like a brick on the dev team.
>
> a product practitioner, S-040

What they do instead	What it costs next	What Bridge answers
What worked is copied by hand, or not copied at all	System drift, falling adoption, handoff done twice	Explorations as a branch and a draft pull request, with their library impact

25 distinct witnesses across 10 named organisations, 35 references: S-001, S-003, S-006, S-007, S-010, S-013, S-016, S-018 and 27 more

Research file · pains P-01 and P-02

The four best documented, and what teams do instead

Pain	What teams do instead	What it costs downstream	Sources
Explorations never come back to the system	What worked is copied by hand, or not copied at all	System drift, falling adoption, handoff done twice	**35**
No decision is written down	The decision stays in a Slack thread, or in the head of whoever generated it	The same debates get replayed with no memory	**19**
Nobody knows who approved what	Ask the lead again, or just assume	Blurred responsibility at the moment of the bug	**19**
Volume outruns review capacity	Skim the review, or merge without reading	Review debt, repaid in production	**16**

06

sources only. The smallest group in the corpus, and the one I would have cut on volume alone.

Every one of them had already built the tool

An internal tool at an online betting operator. A hand built automation workflow at a car manufacturer. An AI agent wired to a team's own design system. A reference harness built over a year.

What it costs them: maintaining a tool that is not their product.

What that means for what Bridge has to beat

The competitor is not another vendor. It is an afternoon of internal tinkering that works well enough.

What the corpus contradicted

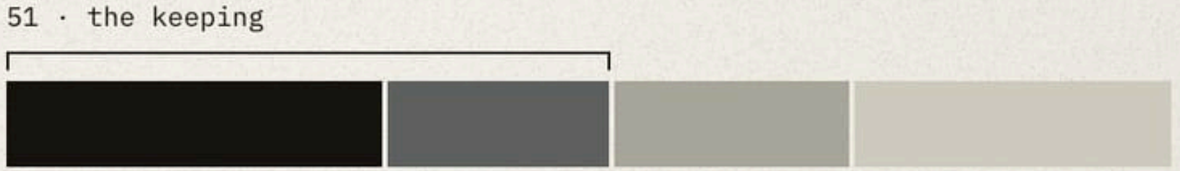
The public conversation in 2026 was about generation: which model, which prompt, how fast. That reframing is what turned a design system tool into a governance product.

The interesting problem was never the generating. It was the keeping.

Sorted by subject, the corpus disagreed with the conversation
51 of 98 sources are about the keeping, not the generating.

The same level in my own decision log
31 decisions re-read: 9 touch no screen at all.

Governance was a level nobody had named · my 98 sources



Governance of explorations	32
Design system adoption	19
New ways of working with AI	20
Business model, measurement, rest	27

51/98
sources are about the keeping.

The same level in my own decision log

I had been sorting my own decisions into three buckets by where they were stored, which classified nothing. Re-read against what each one actually commits, four levels appeared.

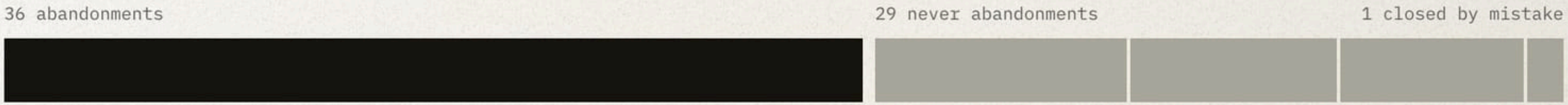
The taxonomy I needed to understand my own project turned out to be the product's own subject matter.

9/31

decisions touch no screen at all.

Product What a person sees and can do	14
Governance How the work gets done, reviewed and shown. Nothing about the product	9
Strategy What the product is, who it is for, what we will not build	7
Technical Implementation, reversible	1

Of the 65 explorations I closed without shipping while building Bridge, 29 were never abandonments



36
abandoned and archived

11
tests closed on purpose

9
ran their course

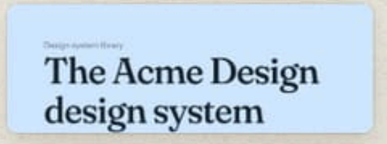
8
replaced by a newer attempt

32 of the 36 abandonments are one decision: Figma drawing fixes stay in the team’s Figma.

Set that batch aside and 4 pieces of work were abandoned one by one.

A closed test can be worth more than a shipped feature: one surfaced 2 defects.

What each pain became

The pain, in the research	What it became, in the product	On screen
Explorations never come back to the system	They open as a branch and a draft pull request	
No decision is written down	Reviews, history and revert: the decision is the artefact	
Nobody knows who approved what	A named review, and a page that says who decided and when	
Volume outruns review capacity	One queue, and an audit on the rendered product	
Six teams that had already built it themselves	Bridge itself, and the build-versus-buy argument it has to win	

Designing the product

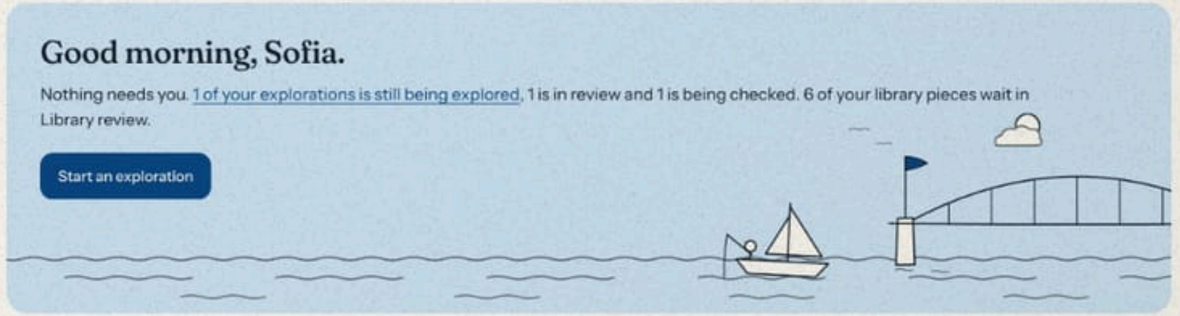
A contributor sees only what is waiting on them. A reviewer gets the queue. A lead gets adoption and drift. Nobody is shown a fact they cannot act on, which sounds obvious and is the rule I broke most often before writing it down.

The contributor, the reviewer and the lead
Each role opens on what it can act on.

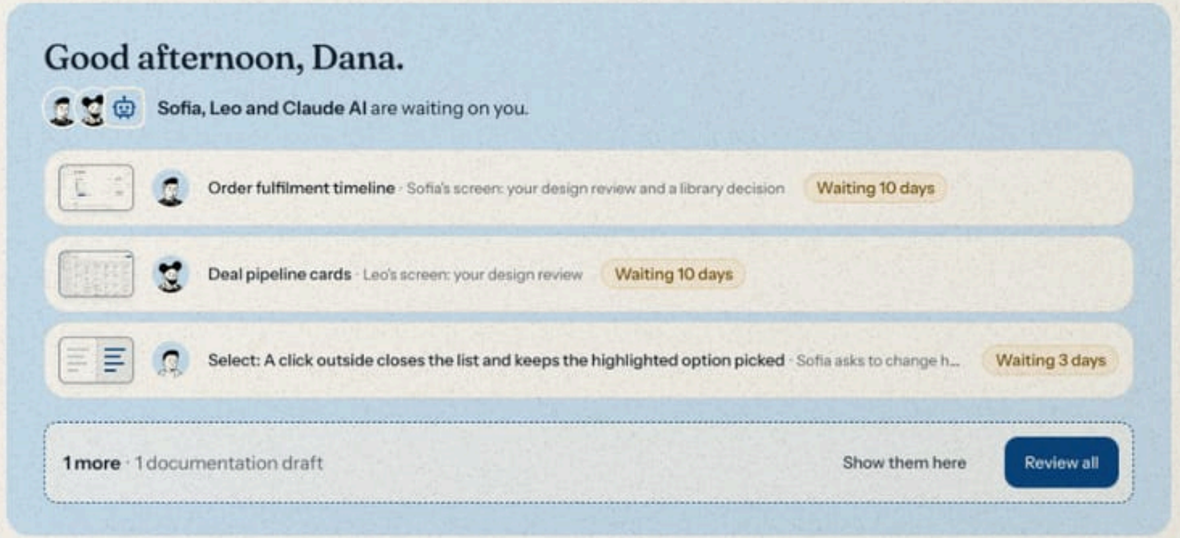
The review is the product
A screen, an open question, and 2 answers.

Measuring adoption on the rendered product
Adoption and drift measured on what the product renders, not on what the code declares.

Contributor · Sofia, designer



Reviewer and lead · Dana, design lead



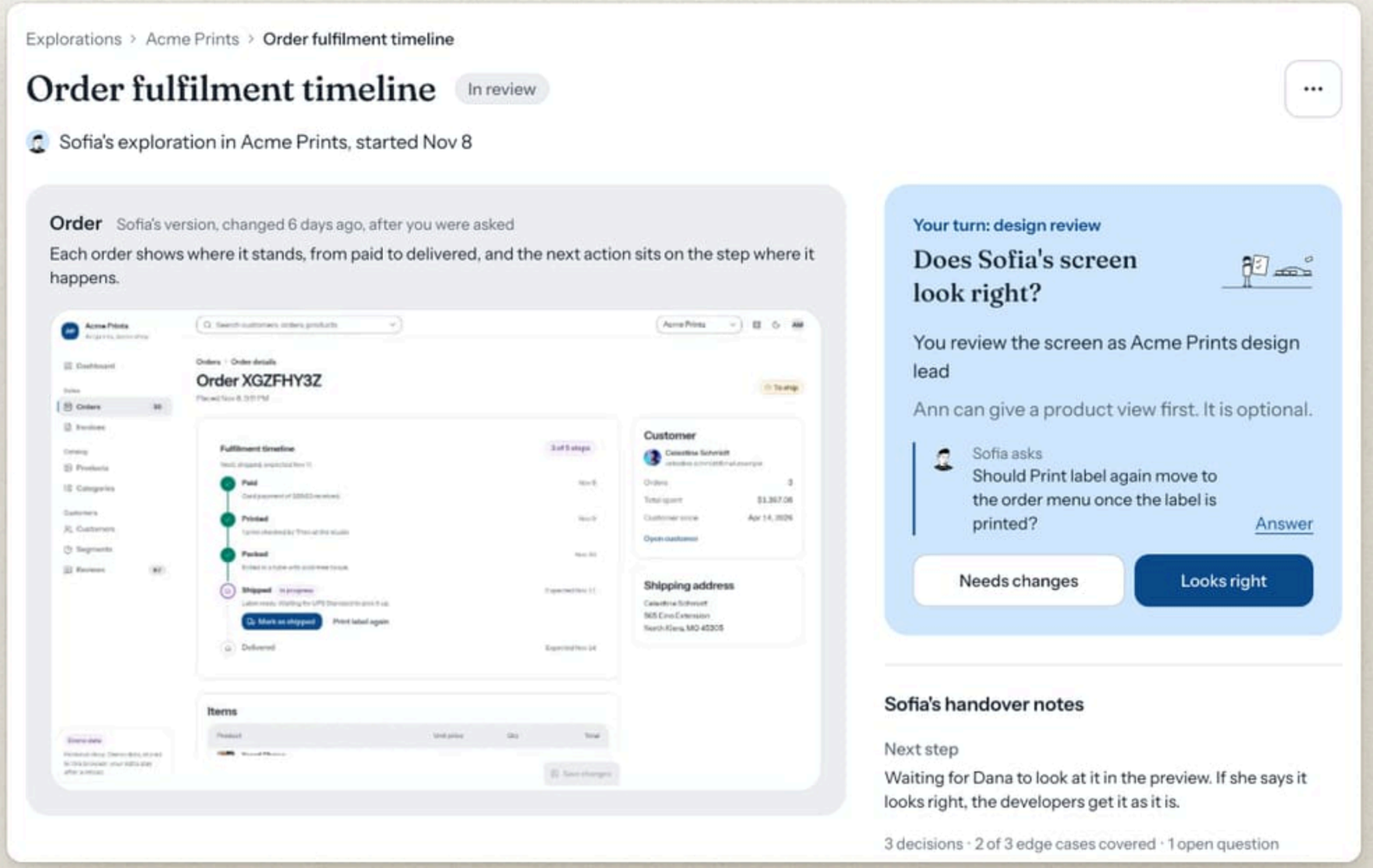
The review is the product

On screen

Sofia's exploration under review (demo team data): her screen, her open question, and 2 answers to give, needs changes or looks right.

What changed

No diff, no pull request, no code. The developers get it as it is. A designer should never be shown a failing test or a merge conflict in Studio; that belongs in GitHub.



Studio · Explorations, Order fulfilment timeline, seen by the design lead

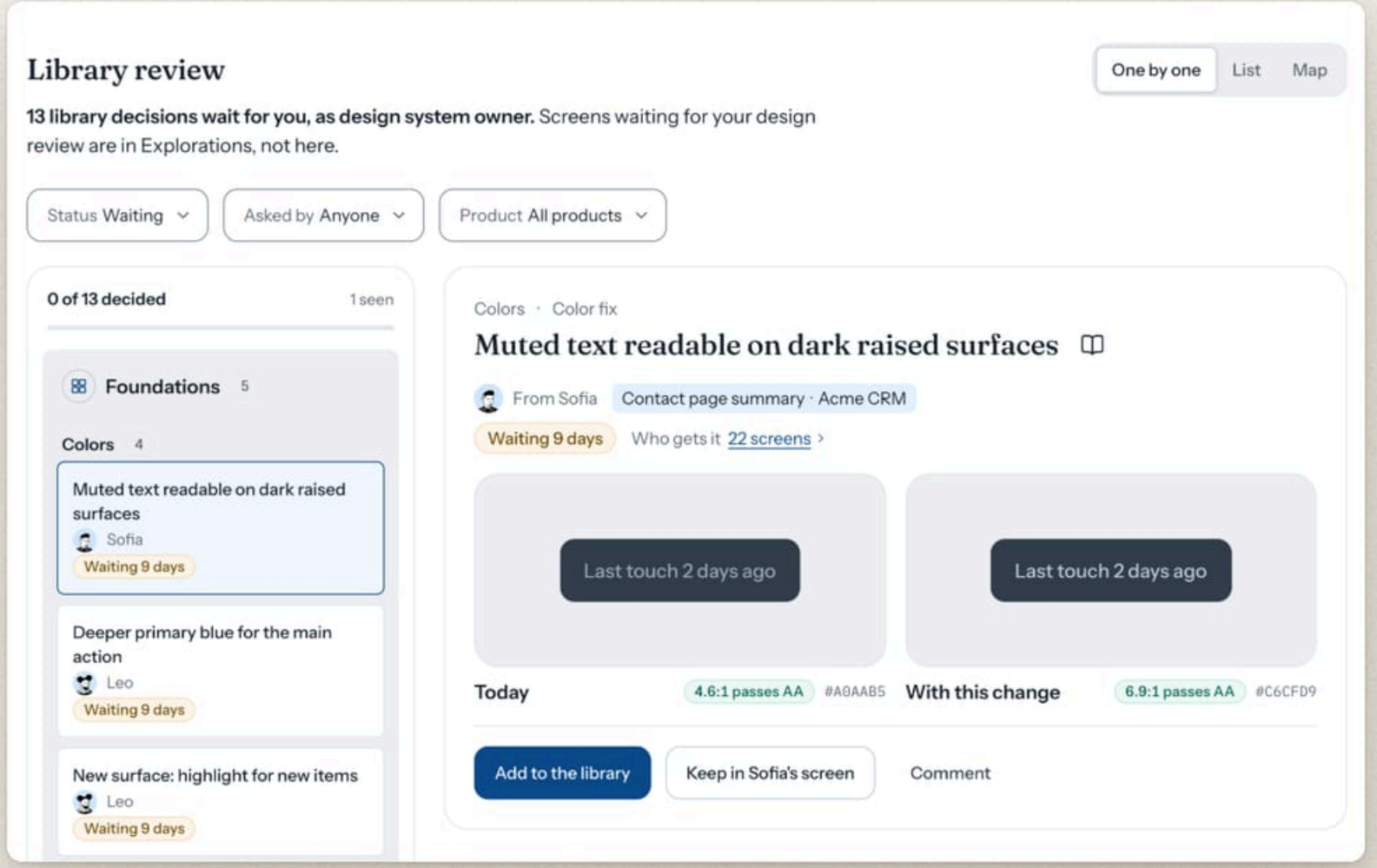
The library review queue

On screen

The library review queue (demo team data): changes to the design system itself, grouped so the grouping says the state and each row shows only what differs.

What changed

A design system change is decided piece by piece, next to the screen that asked for it: add it to the library, or keep it in that screen.



Studio · Library review, one by one, seen by the design system owner

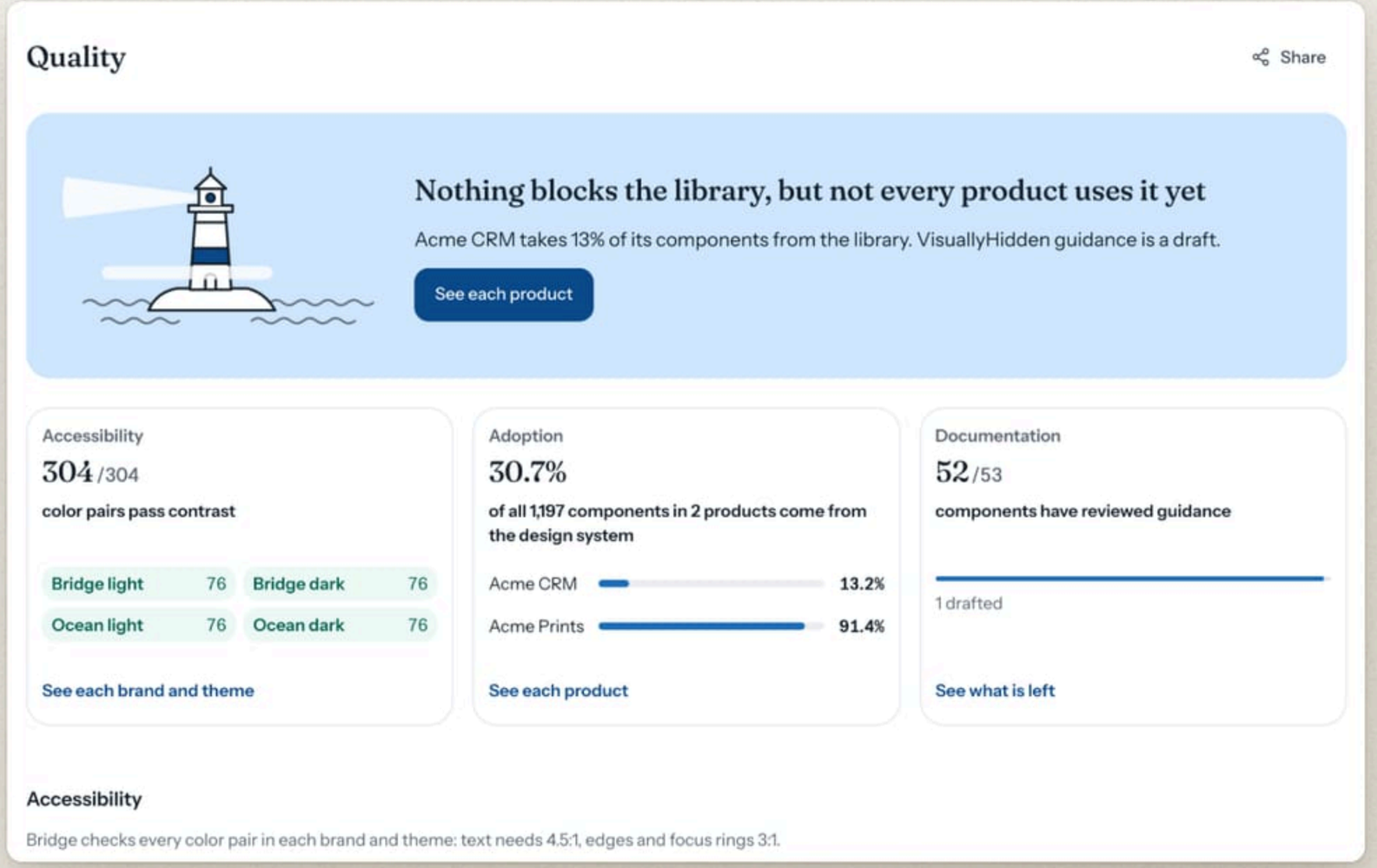
Measuring adoption on the rendered product

On screen

The quality screen (demo team data): design system adoption and drift, measured on the rendered product rather than declared.

What changed

An audit that trusts a declaration measures intentions. The quality screen measures adoption and drift on what the product actually renders, not on what the code declares.



Studio · Quality: accessibility, adoption, documentation

The design system it runs on

Bridge runs on its own design system, and its interface uses only real components from it. No hand made imitations: an audit that counts an imitation as adoption is a governance tool lying about governance.

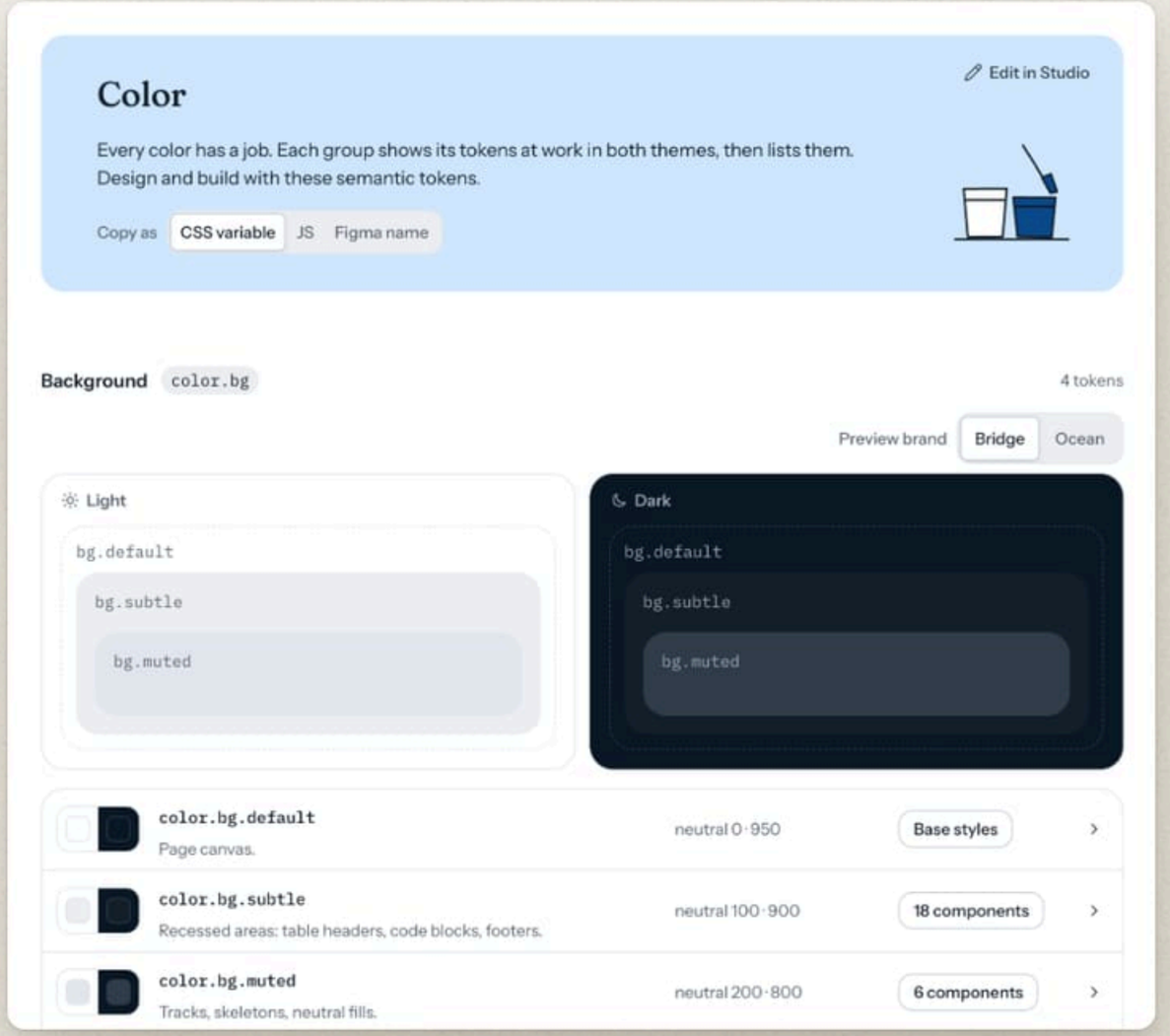
The rule I held to throughout: never take a capability away from someone already using it.

53
React components

119
semantic tokens

2
brands, Bridge and Ocean, light and dark

Only real components, never imitations
An early audit counted imitations as adoption: flattering and false.

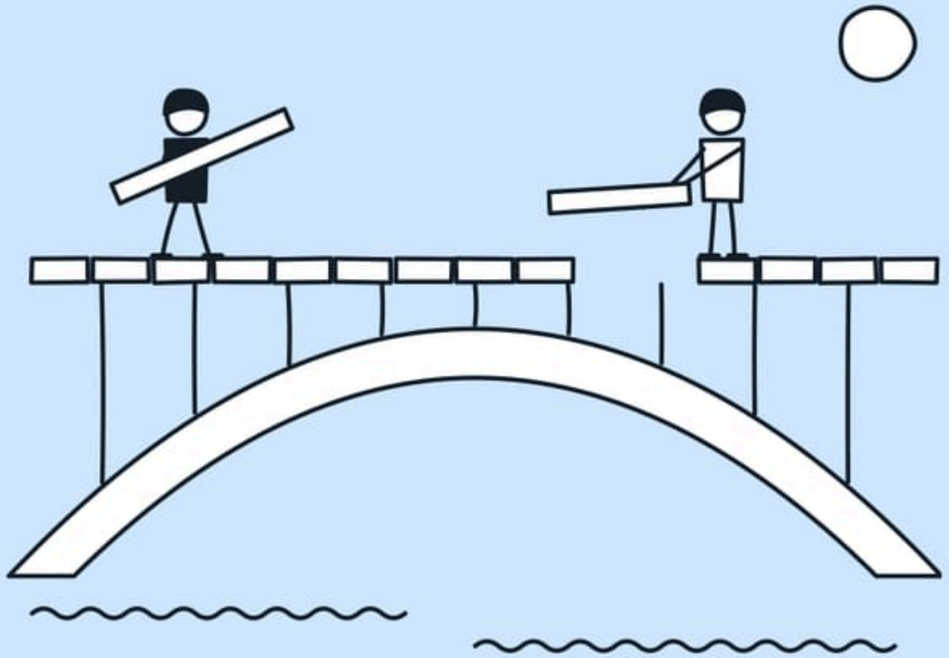


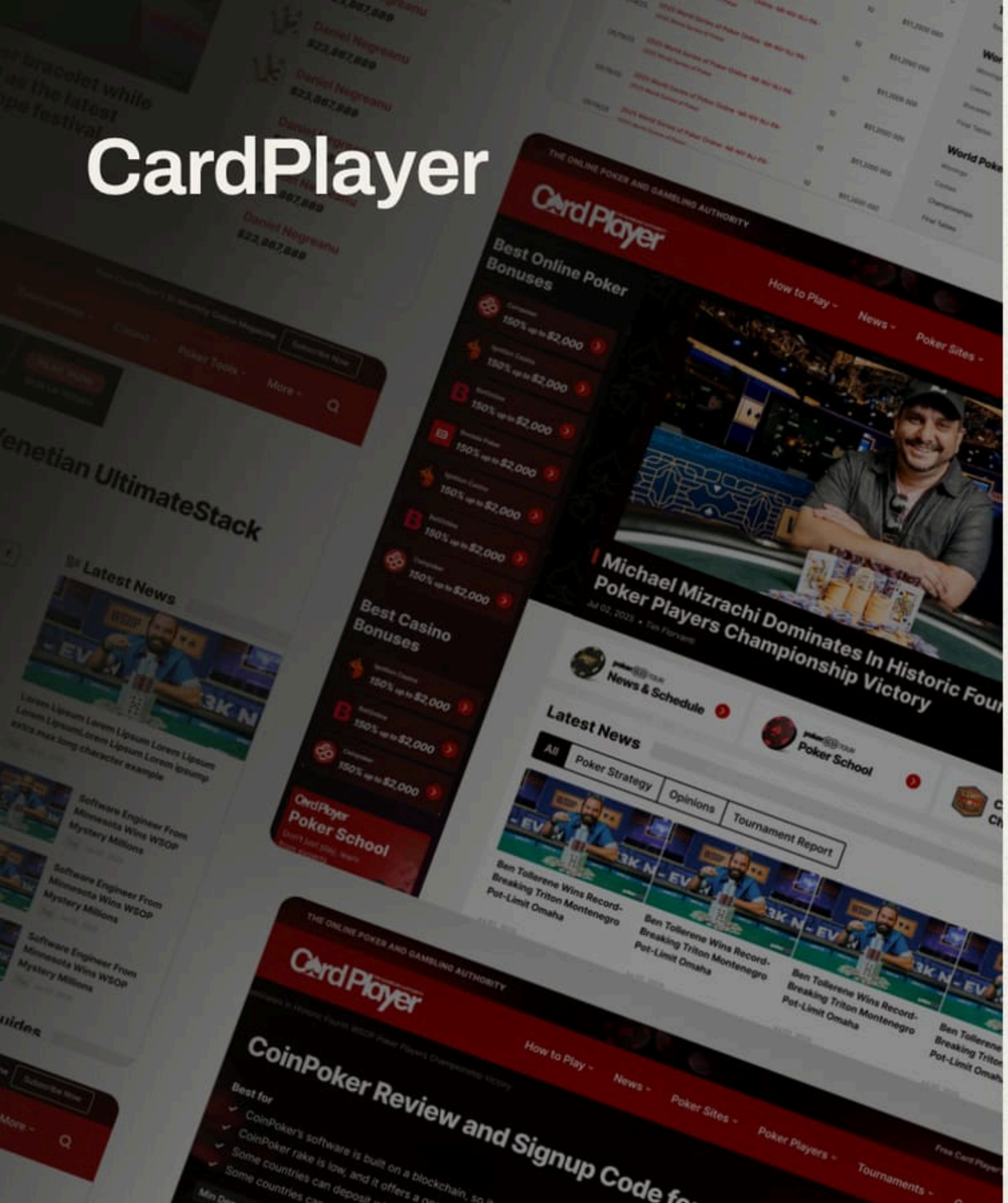
Library · Foundations, Color, light and dark

5 pains, each one answered in the product

I researched it, designed it, and built it. Live at bridgedesignsystem.com.

- 51 of the 98 sources are about the keeping, not the generating: the finding that turned a design system tool into a governance product.
- The product is live, free to use and public on GitHub, with components and tokens under MIT.
- The research is published in full: 98 sources, the 5 pains, the log of 65 closed explorations and the 31 decisions.
- Bridge runs on its own design system, and its interface uses only real components from it.





Case study · ClickOut Media, poker media website · 5 months

Template Ecosystem Redesign

CardPlayer is the world's oldest poker magazine, publishing since 1988, a flagship SEO and affiliate asset acquired by ClickOut Media. Years of incremental growth had left template sprawl and design debt that made every iteration costly. The redesign goal was to re-architect the template ecosystem, not to ship a visual refresh.

My role	Senior Product Designer (Solo Designer)
Crew	1 SEO Director, 1 PM, Dev Team
Problem	143 templates with no canonical system, overlapping layouts serving the same user intent, no responsive design, no macro design governance. The team had no established method to manage a redesign at this scope.

Mission 01 — System Diagnosis & Scope Control — 5 sub-missions

Mission 02 — Canonical Layouts & Growth Levers — 5 sub-missions

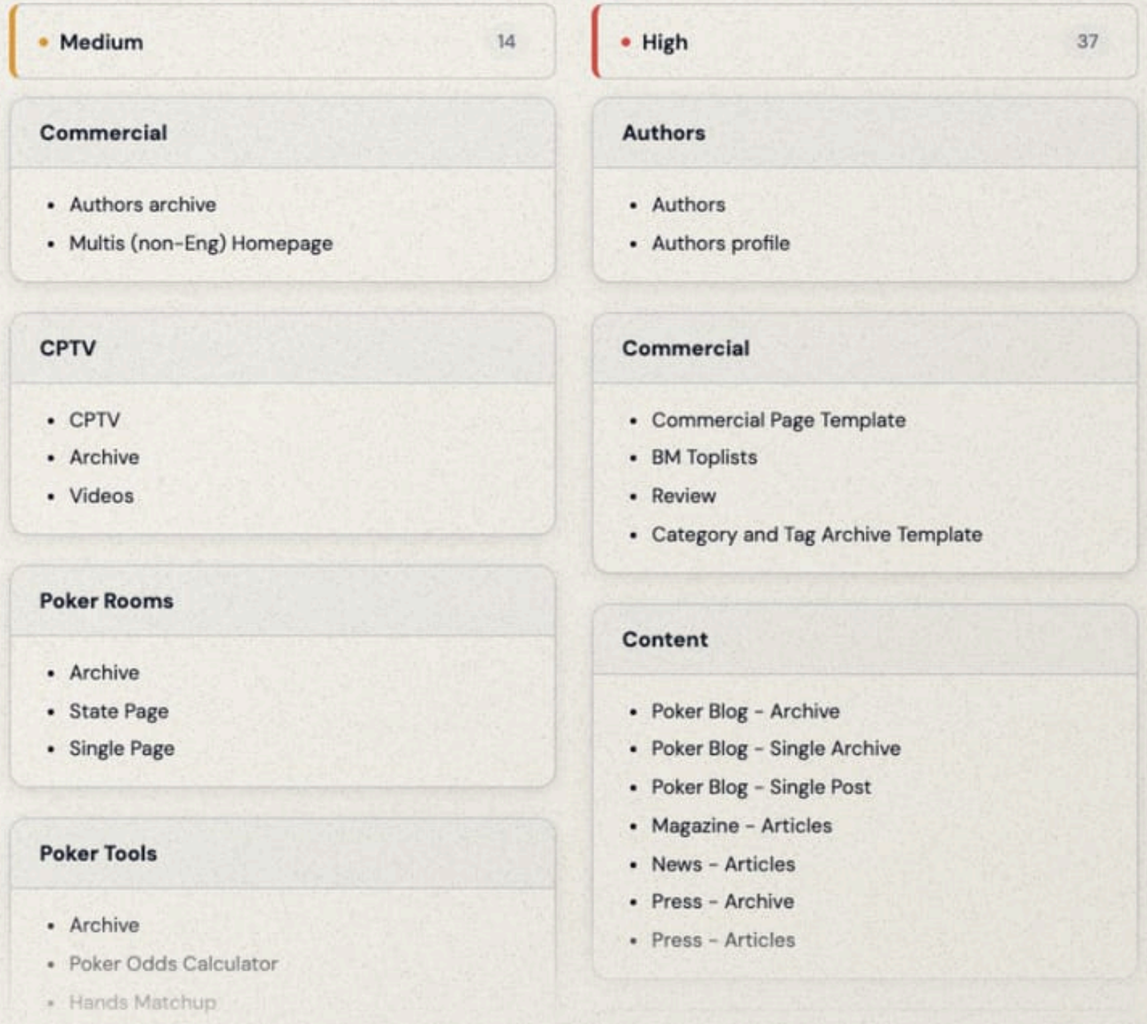
Mission 03 — Reworking Key Templates — 5 sub-missions

Mission 04 — Governance & Delivery Efficiency — 5 sub-missions

System Diagnosis & Scope Control

The redesign started with “redesign everything” but no prioritization, no sequencing, no definition of done. The product had become a patchwork of templates added over years to ship fast: the same user intent served by multiple layouts, and no one had a system view.
The real risk was an unfinishable project, not bad UI.

Full template inventory	Grouped by category, intents mapped.
Priority tiers	High/medium/low sequencing instead of parallelizing everything.
Desktop-first strategy	Primary traffic surface stabilized first, based on analytics data.
Navigation rework	IA rebuilt around high-intent surfaces.
Unique vs secondary	Canonical set defined, variants demoted.



Priority tiers · Low 20, Medium 14, High 37 · 71 desktop templates

Desktop-first strategy

On screen

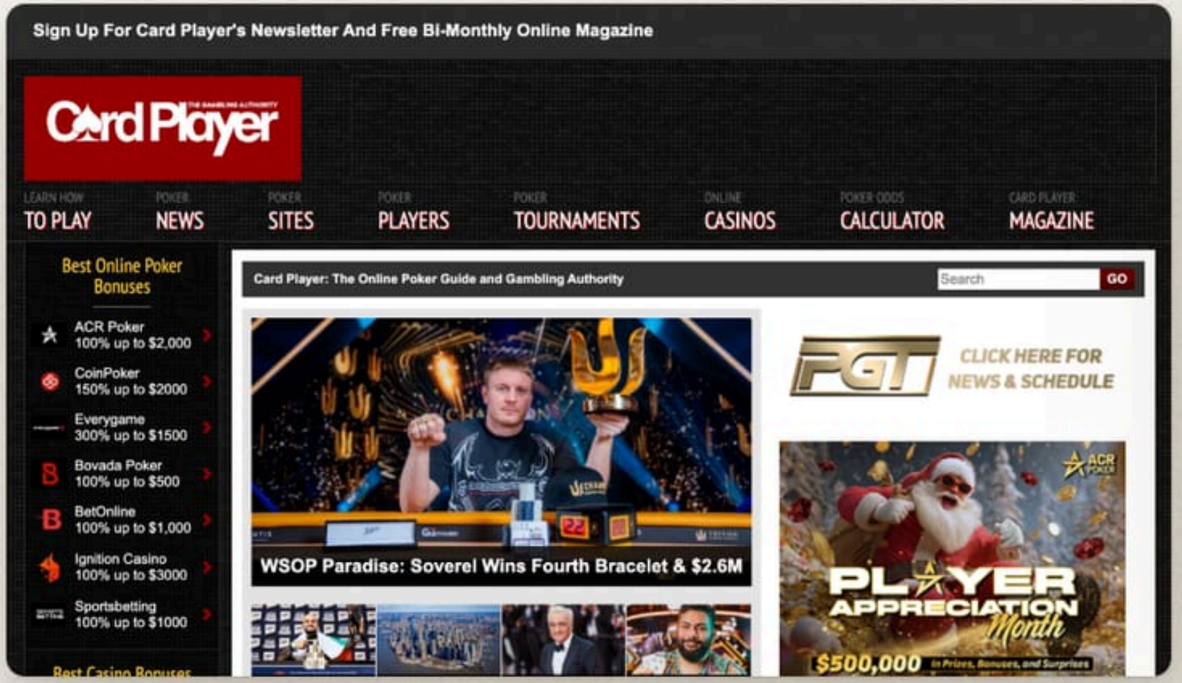
The original CardPlayer homepage on desktop, then one of my high-priority desktop templates: the news article, XL in my tracker.

What changed

At project start, the site wasn't responsive and analytics showed desktop as the primary traffic driver. I designed high-priority desktop templates first to protect the primary traffic surface: a data-driven sequencing decision, not a design preference.

Original homepage

Original site · not my work



News article

high priority, XL · desktop



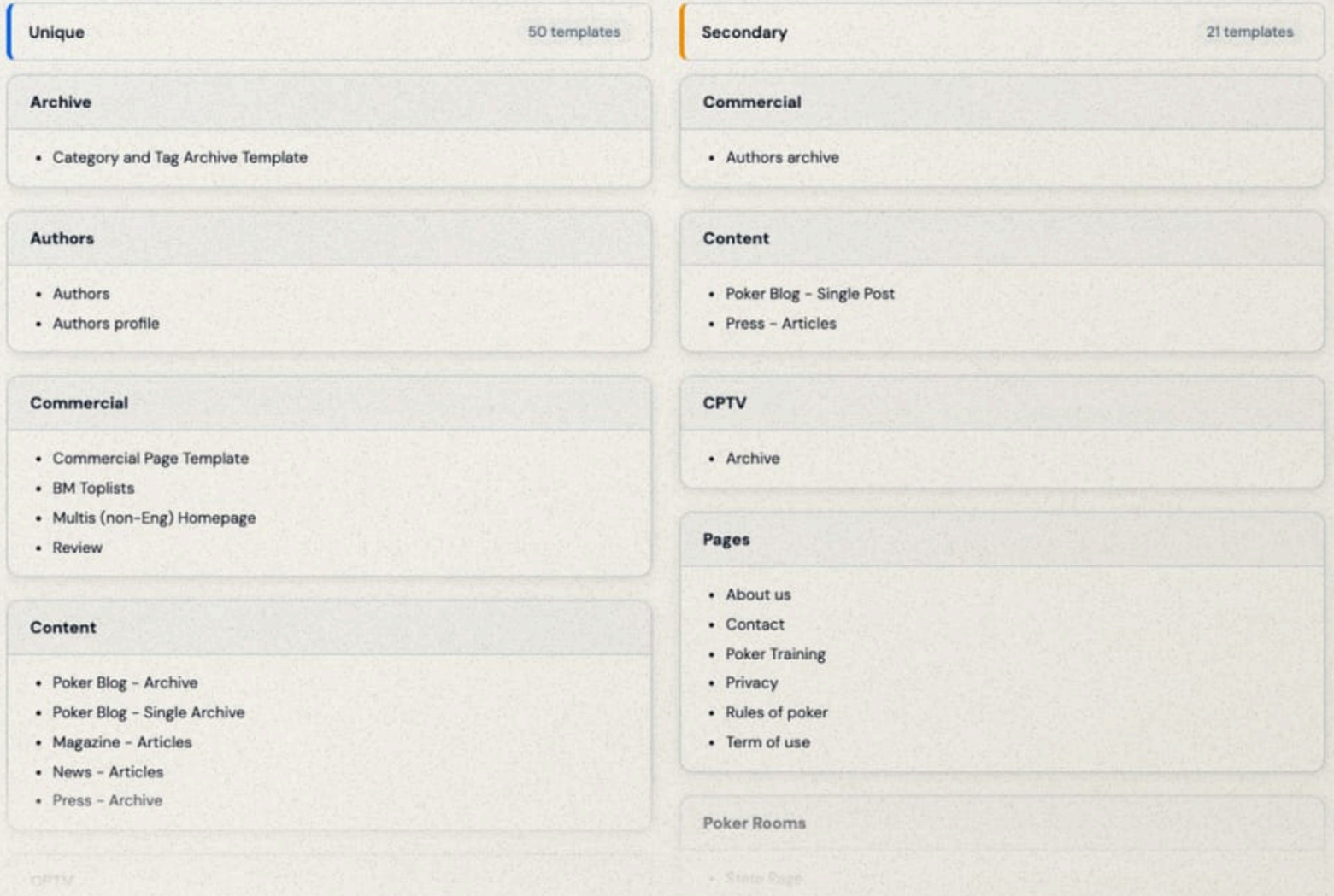
Unique vs secondary

On screen

My audit of the live templates in 2 columns, grouped by category: what is truly unique, what is a secondary variation.

What changed

I grouped templates by user intent and content hierarchy, defined a canonical set of core templates, demoted secondary templates into controlled variants, and merged accidental variants back into the canonical system.



Unique 50 · Secondary 21

Navigation rework

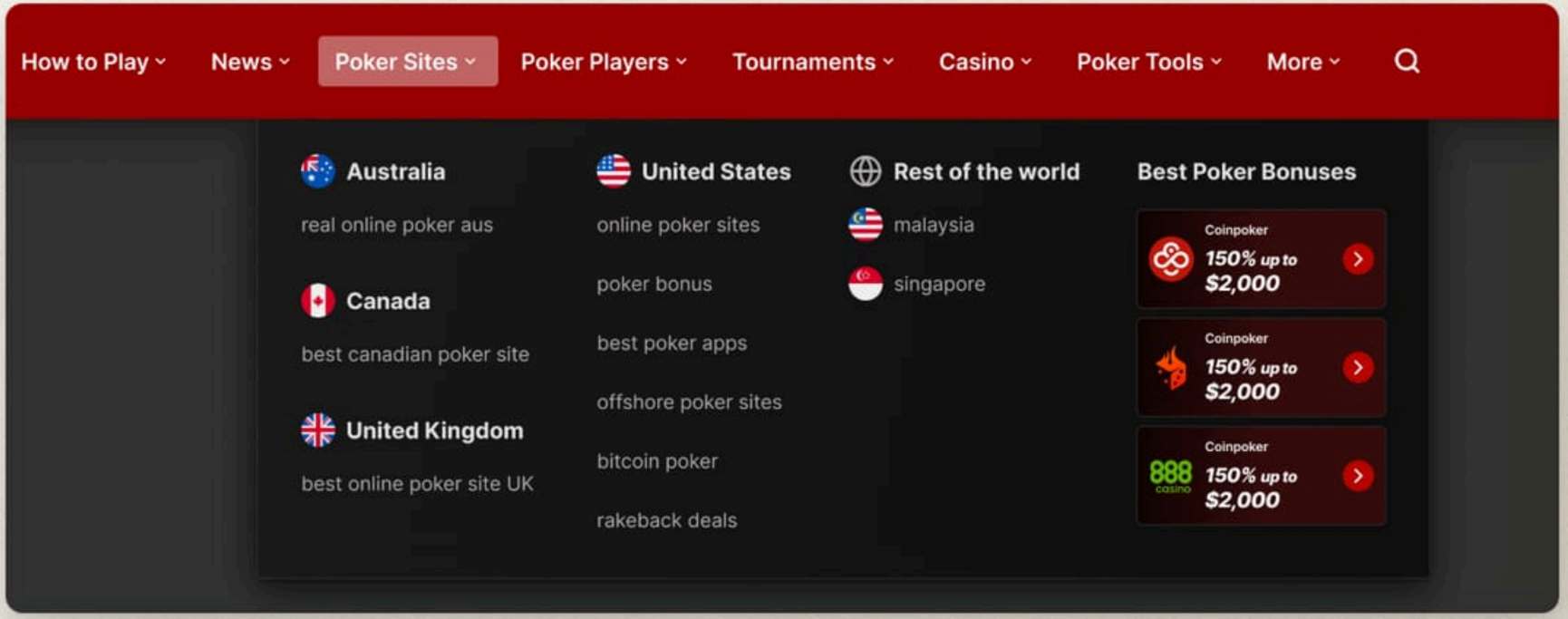
On screen

Two of the 8 mega-menus of my redesigned header: Poker Sites by country with the best bonuses, Poker Tools with the Odds Calculator.

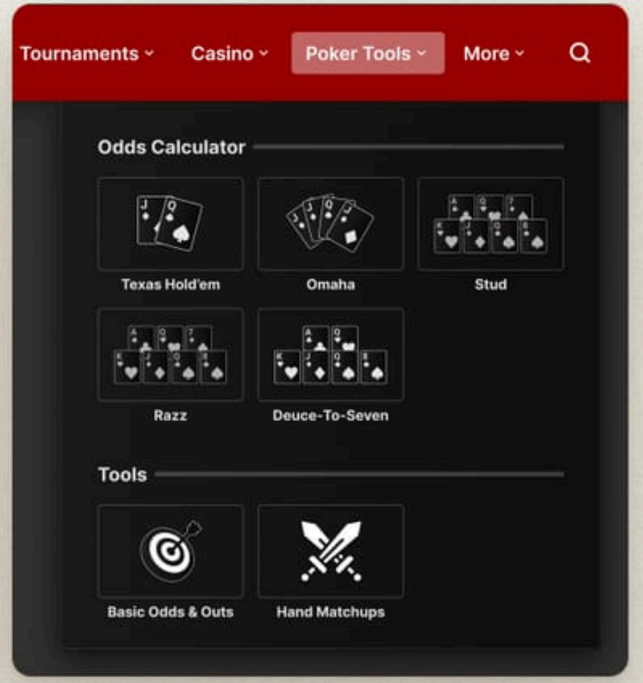
What changed

Content categories overlapped, and high-intent surfaces (Tools, Guides, key commercial categories) were not surfaced clearly. I rebuilt the information architecture, simplified and clarified top-level categories, and repositioned high-intent sections: a clearer user mental model and stronger internal linking for SEO.

Poker Sites by country, Best Poker Bonuses



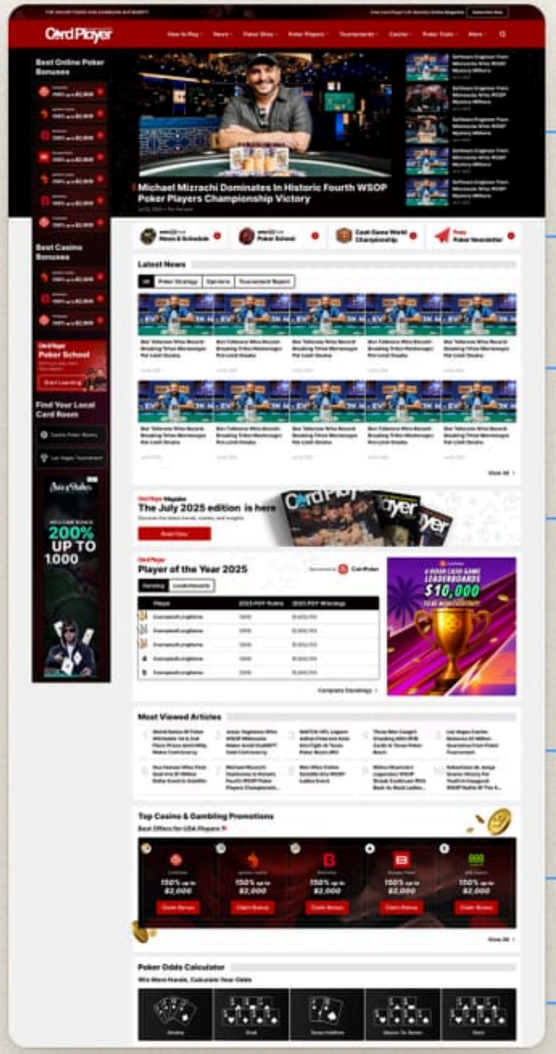
Poker Tools Odds Calculator



Canonical Layouts & Growth Levers

Multiple templates existed for the same user intent across archives and listings, fragmenting UX and multiplying maintenance cost. Monetization (ad columns) constrained layout decisions on every page. The goal: default to reuse, and reserve rework for cases where IA or navigation hierarchy was objectively inconsistent.

Canonical archive skeleton	9 archive families, 1 reusable pattern.
Negotiated layout rules under ad revenue constraints	Reduced homepage ad load, two-column monetization kept elsewhere.
Imported proven patterns from competitor affiliate sites	Google News CTA, newsletter hooks, country guides, trust blocks.
Missing templates identified	Taxonomy archives and search results.
Reuse vs rework rule	Reuse by default, rework if IA is broken.

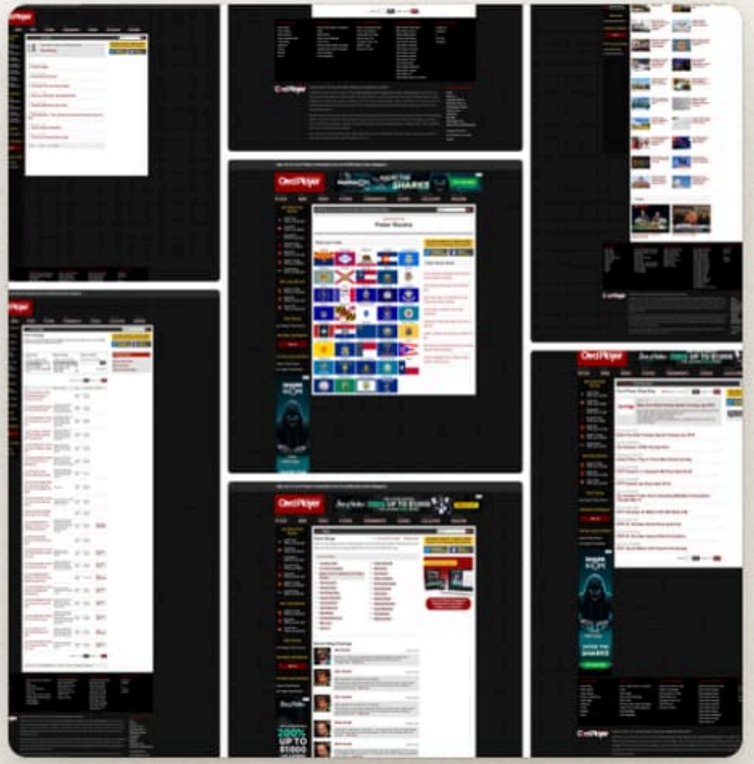


Redesigned homepage, desktop · down to the Odds Calculator

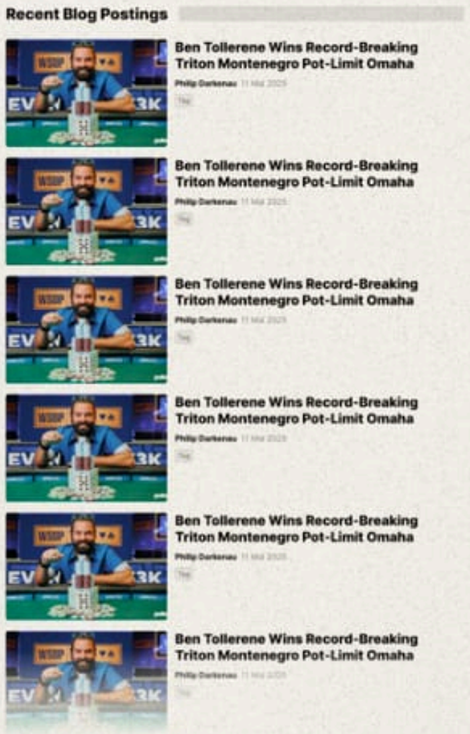
9 archive families, 1 skeleton

"Same intent, different template" across blog/news, press releases, topics, tournaments, authors, poker rooms, regions, CPTV, strategy. I standardized a canonical archive/list skeleton with predictable zones.

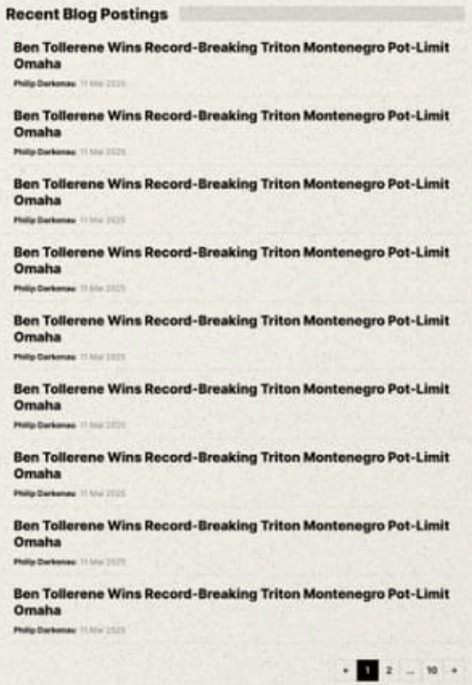
Original archive pages not my work



List with thumbnails



Text list



Card grid

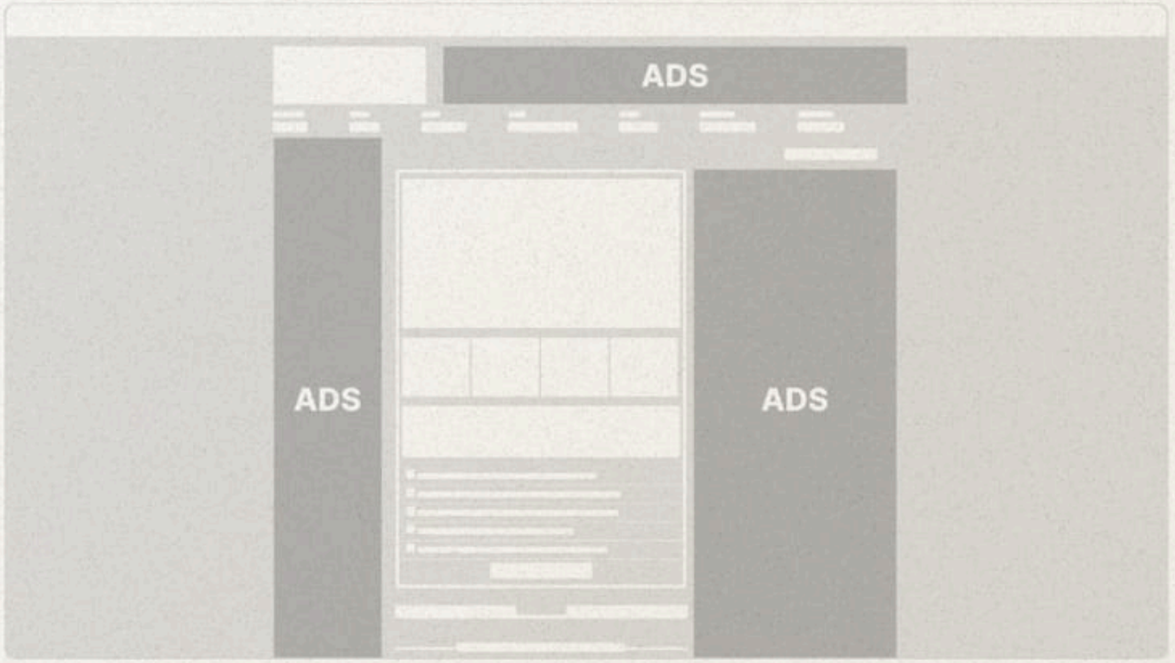


The generalised list components

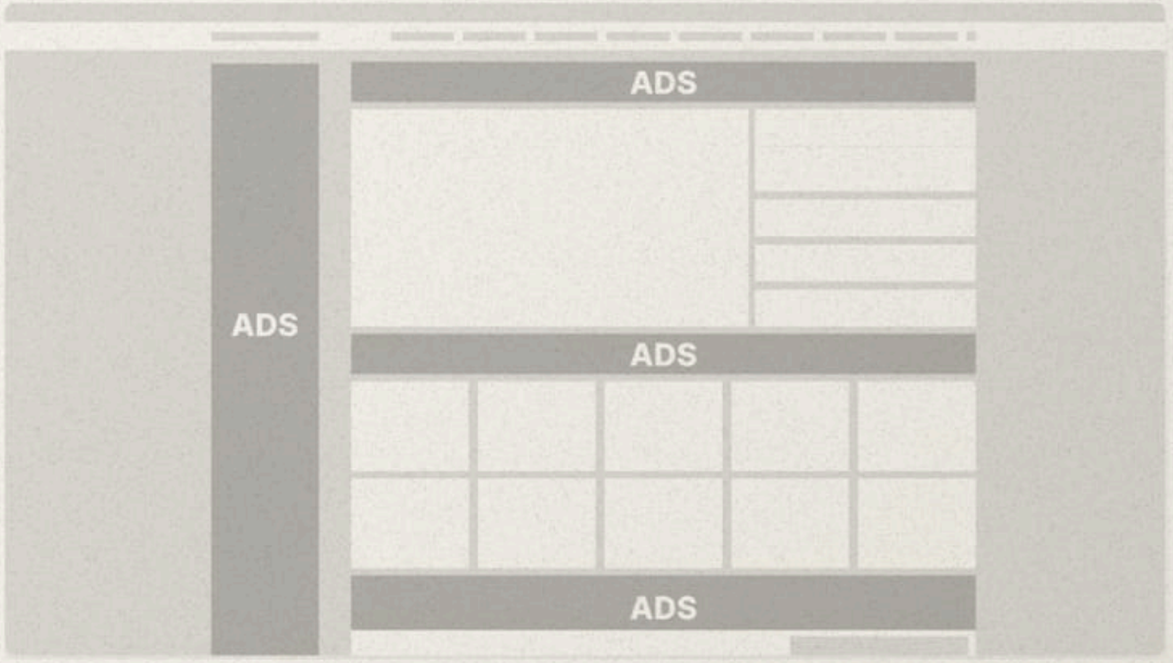
Homepage under ad constraints

I negotiated a homepage-specific compromise: reduced right-side ad load to reclaim central content focus. I kept a monetization column where it mattered (best bonus / key redirects) while moving secondary items into the content flow, and accepted full two-column constraints on other templates where the trade-off was less harmful.

Original homepage ad zoning



Redesigned homepage ad zoning



Toplists, designed to be A/B tested

On screen

The final toplist cards of the live site, desktop and mobile, in their 2 types: casino and poker / betting.

What changed

Before the redesign, on the live site, I explored new versions of the toplists, the operator rankings that carry the affiliate offers, to improve conversion: different ways to organize and present each offer. These are the final versions, designed to be A/B tested.

Desktop Casino



Mobile Casino



Desktop Poker / Betting



Mobile Poker / Betting



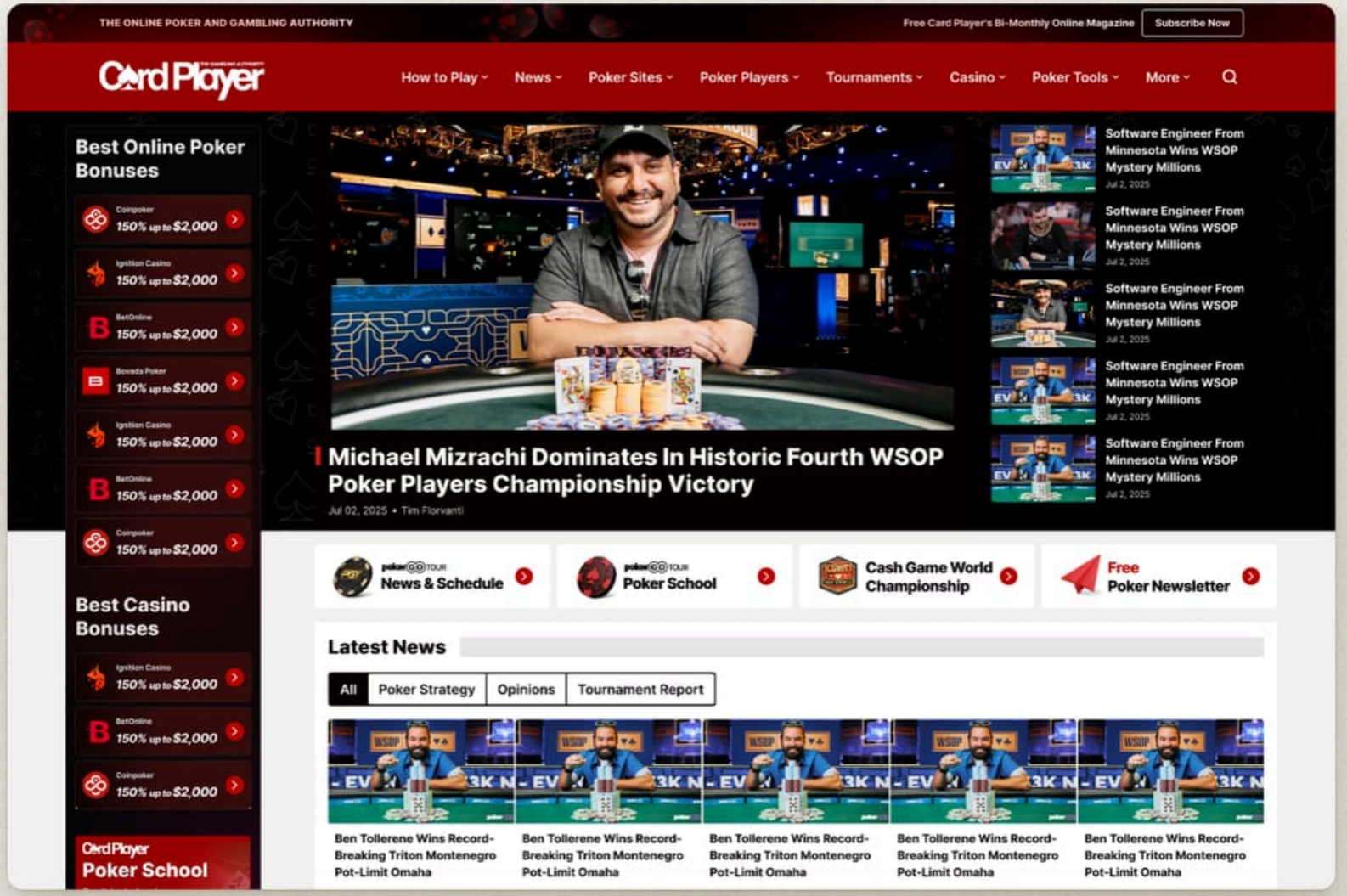
Redesigned homepage

On screen

The top of my desktop homepage: one bonus column on the left, the lead story and 5 headlines, 4 shortcuts, then Latest News by tab.

What changed

The homepage is the XXL template of my tracker, 11 design iterations. The objective: clearer editorial hierarchy and scanability, stronger brand authority and trust signals, without diluting CardPlayer's DNA.



Homepage, desktop · top of page

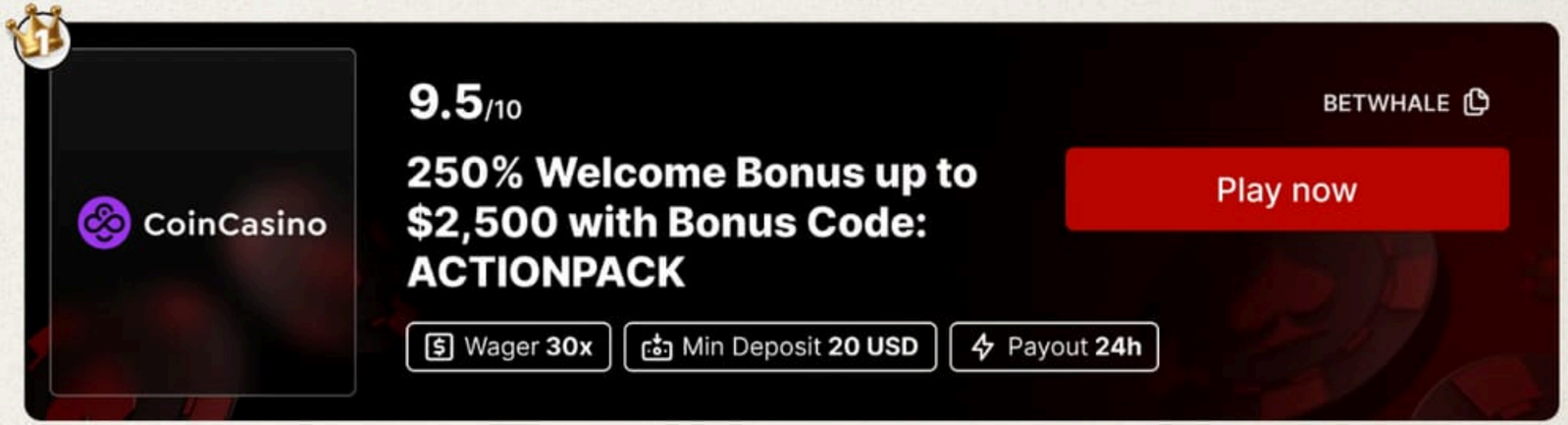
Dark and light themes, made to be A/B tested

On screen

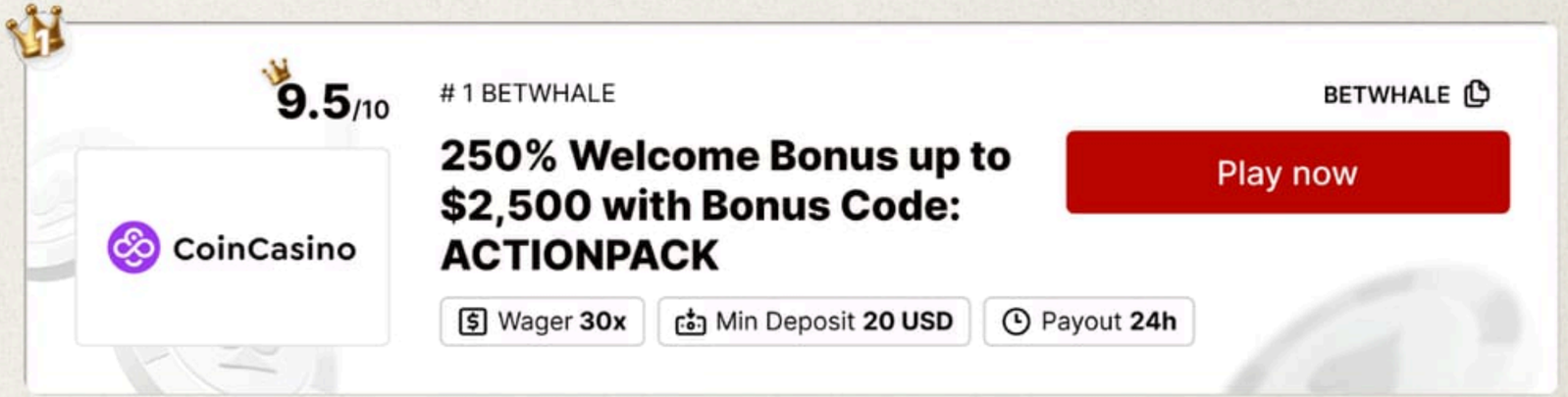
The toplist card of my redesign in its 2 themes: rating, welcome bonus, terms tags and the Play now button.

What changed

For the redesign, I explored a dark theme and a light theme, designed to be A/B tested.



Dark

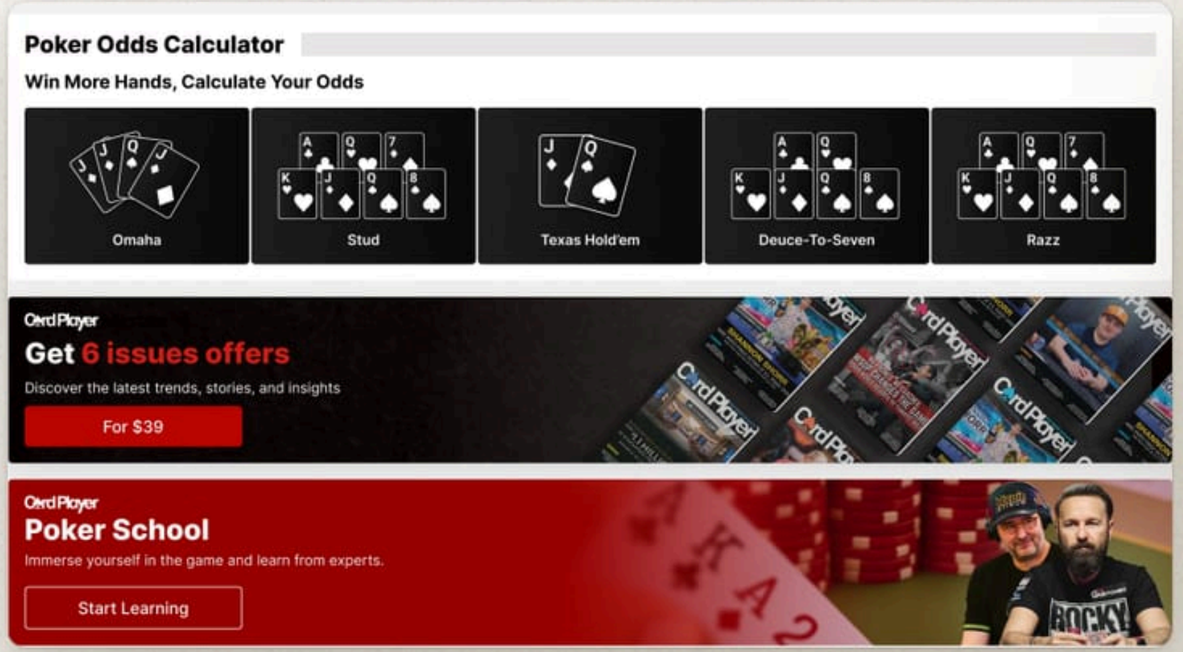


Light

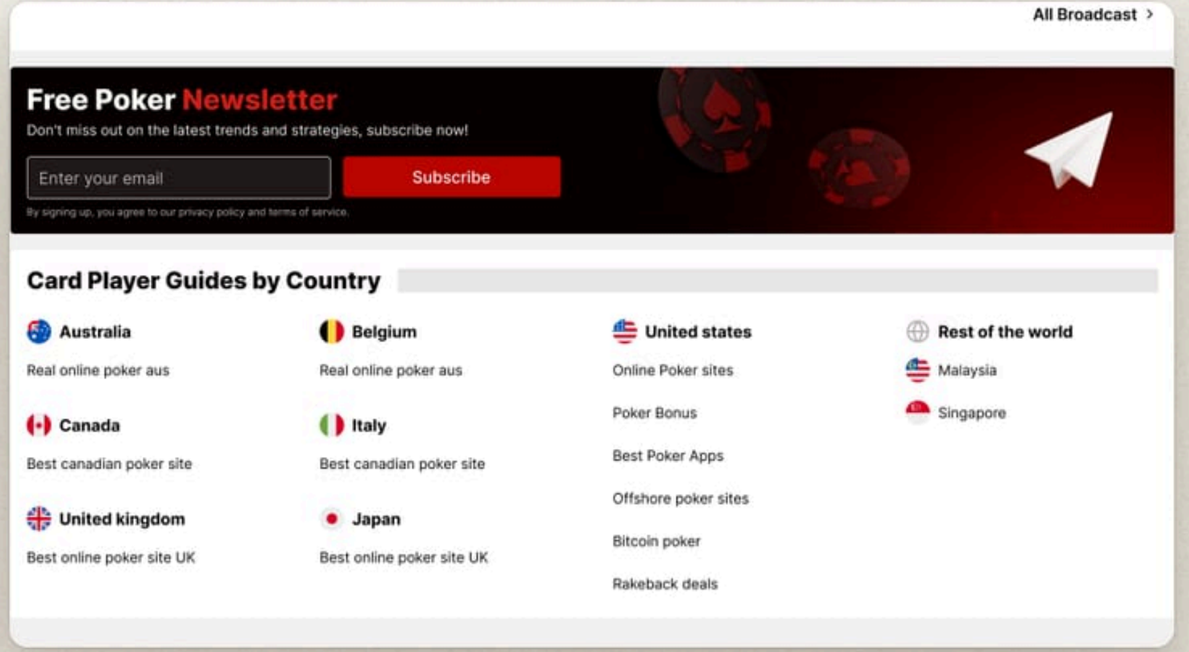
Imported proven patterns

Patterns proven on competitor affiliate sites, added beyond the initial brief: country guides, most viewed articles and the Poker Odds Calculator on the homepage, newsletter subscription surfaces, stronger framing for magazine offers, and authority/trust blocks with hard proof points.

Homepage, lower half Odds Calculator, magazine offer



Homepage, bottom newsletter, guides by country



Mobile adapts the system

On screen

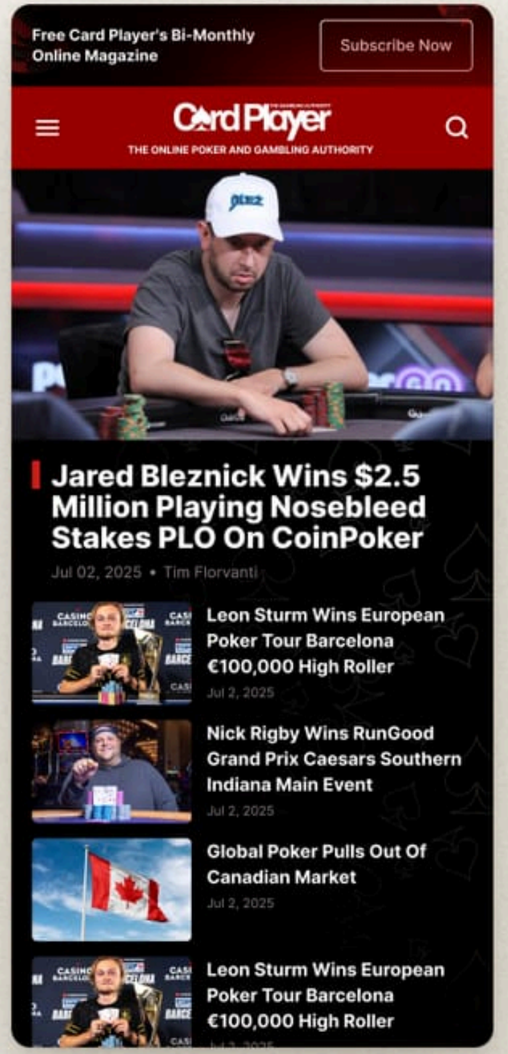
The original CardPlayer site on a phone, then my redesigned homepage at mobile width.

What changed

The original site wasn't responsive. I started the mobile adaptation only once core desktop patterns were standardized, so that mobile would adapt the system rather than become a separate redesign.



Original site · not my work



Redesigned homepage, mobile

Reworking Key Templates

Highest-complexity surfaces had layered page variants for the same intent, mostly due to legacy fast shipping and lack of global governance. The highest leverage was collapsing them into fewer scalable skeletons reusing canonical modules.

Magazine	3-page journey reduced to 1 template.
Tournament	Competing layouts consolidated into 1 hub with fixed pillars (News, Winners, Schedule, History).
Region	Mixed pages reduced to 1 template with tabs (News / Rooms) and region dropdown.
CPTV	Parallel layouts consolidated into 1 constrained skeleton with standardized entry points.
Maintenance reduced	Any future UI update applies once.

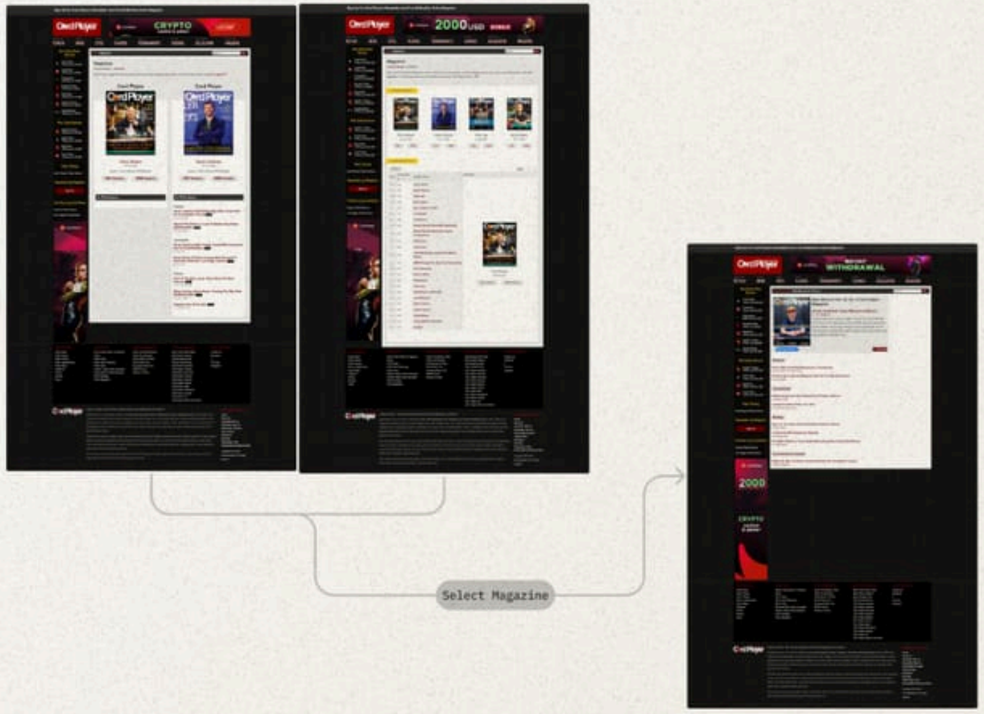


Magazine and WSOP hub, desktop · 2 of the 4 reworked templates

Magazine: 3 pages, 1 template

Latest issues and issue search lived in separate templates, with a third page for issue summary. I replaced the multi-template flow with one unified Magazine template: latest issues, search/browse and issue summary (chapters) in a single place, with the core actions standardized (read online, download PDF, print subscription).

Original site `not my work · 3 pages`



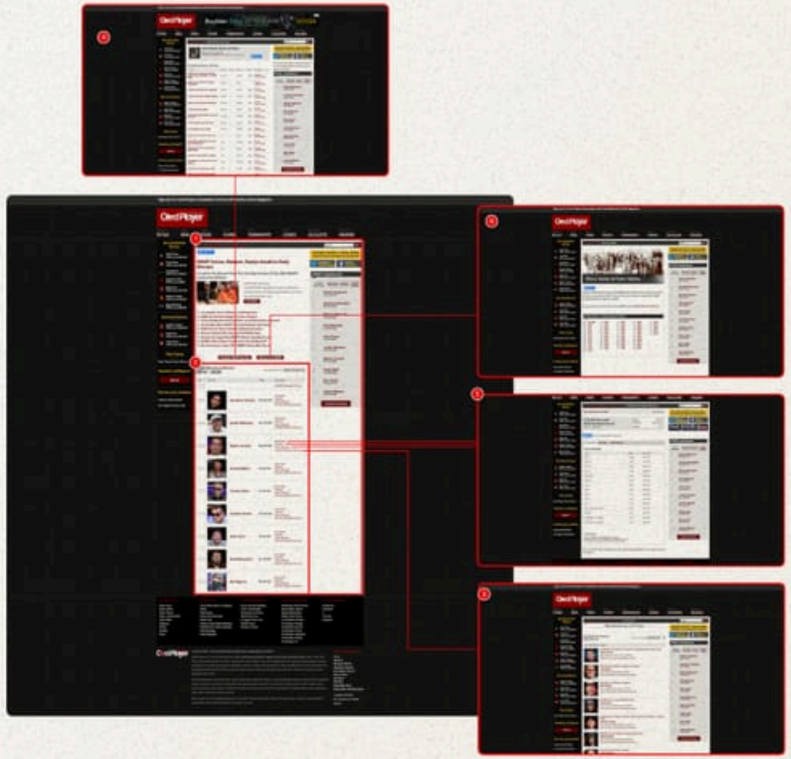
Magazine template `summary open`



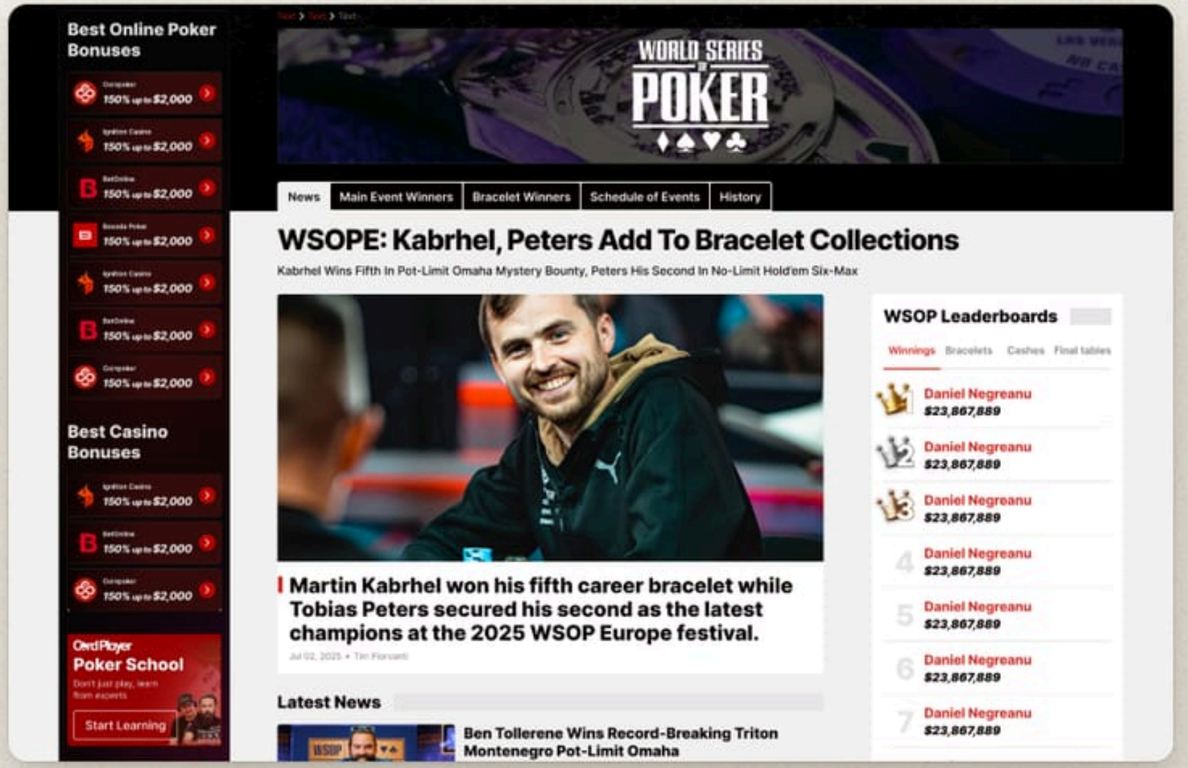
Tournament: 1 hub, fixed pillars

Tournament content was fragmented across multiple page types and navigation paths (news, winners, schedules, history), forcing users to bounce between templates to understand one event. I consolidated tournament behaviors into one hub-style template with explicit content pillars.

Original site not my work · 6 linked pages



WSOP hub 5 pillars as tabs



Region: 1 template, 2 tabs

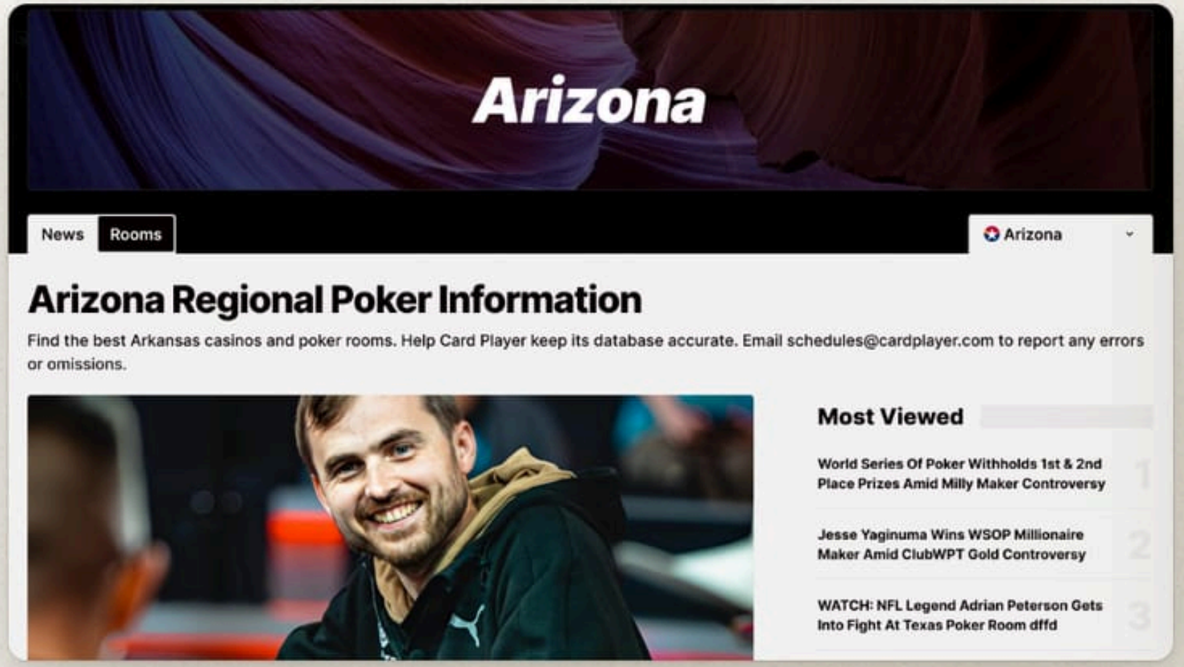
On screen

The Arizona page in its 2 states, News and Rooms: same banner, same tabs, same region dropdown.

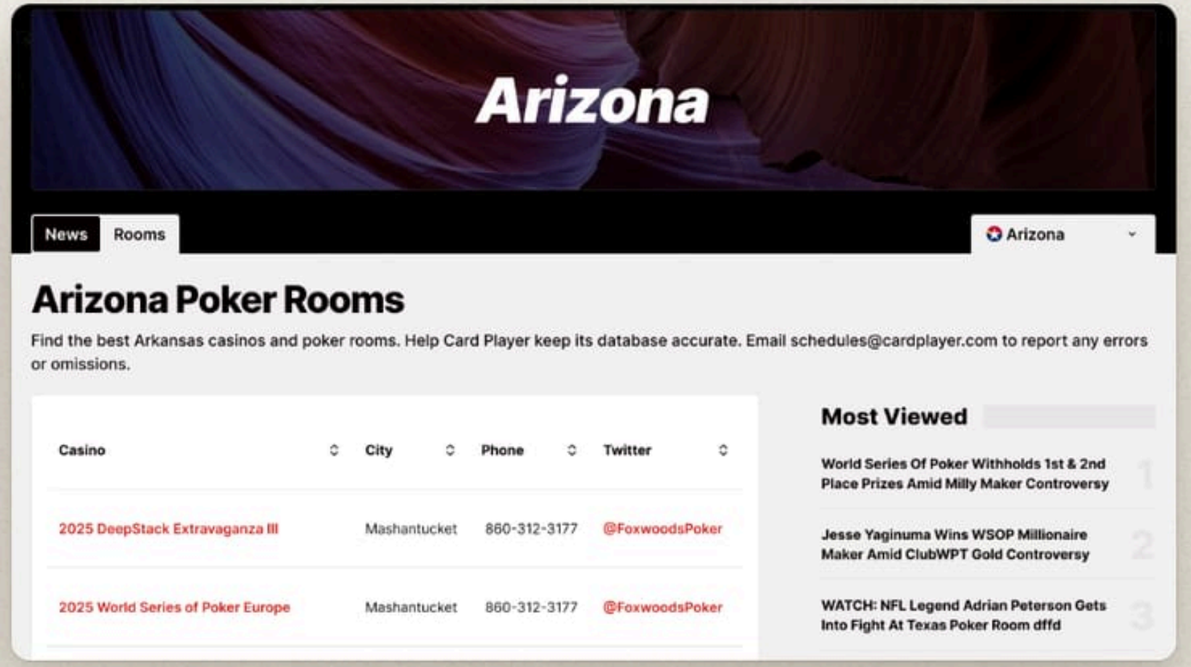
What changed

Region pages were treated as one-offs, mixing news and rooms without a clear structure. I rebuilt the region experience as a structured index: a pre-selected region landing, tabs to move between News and Rooms, and a single region dropdown for switching countries or states without creating new templates.

News tab regional news first



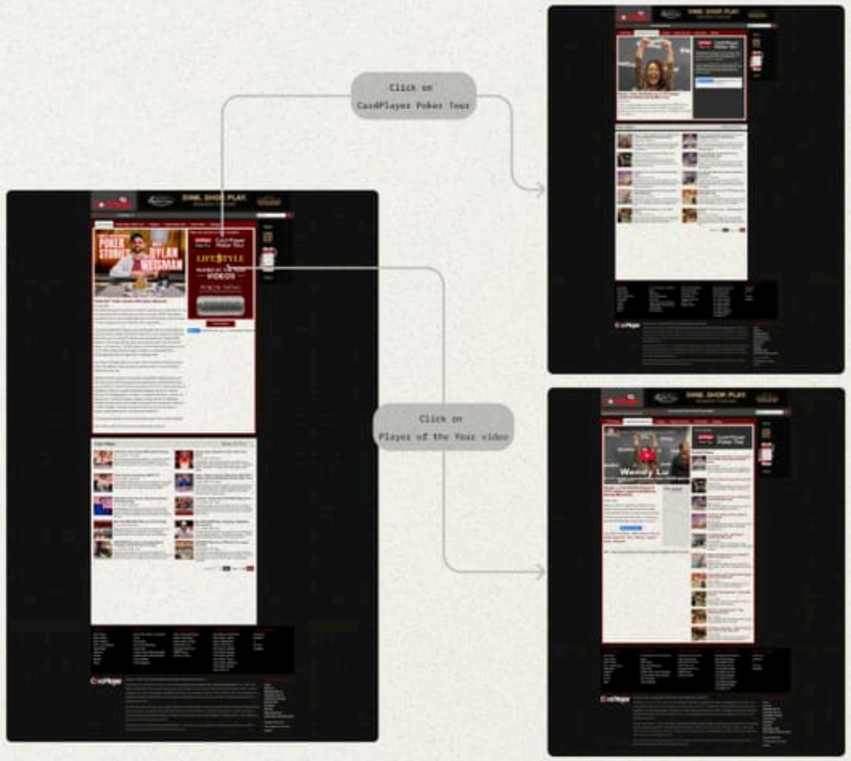
Rooms tab same header, same dropdown



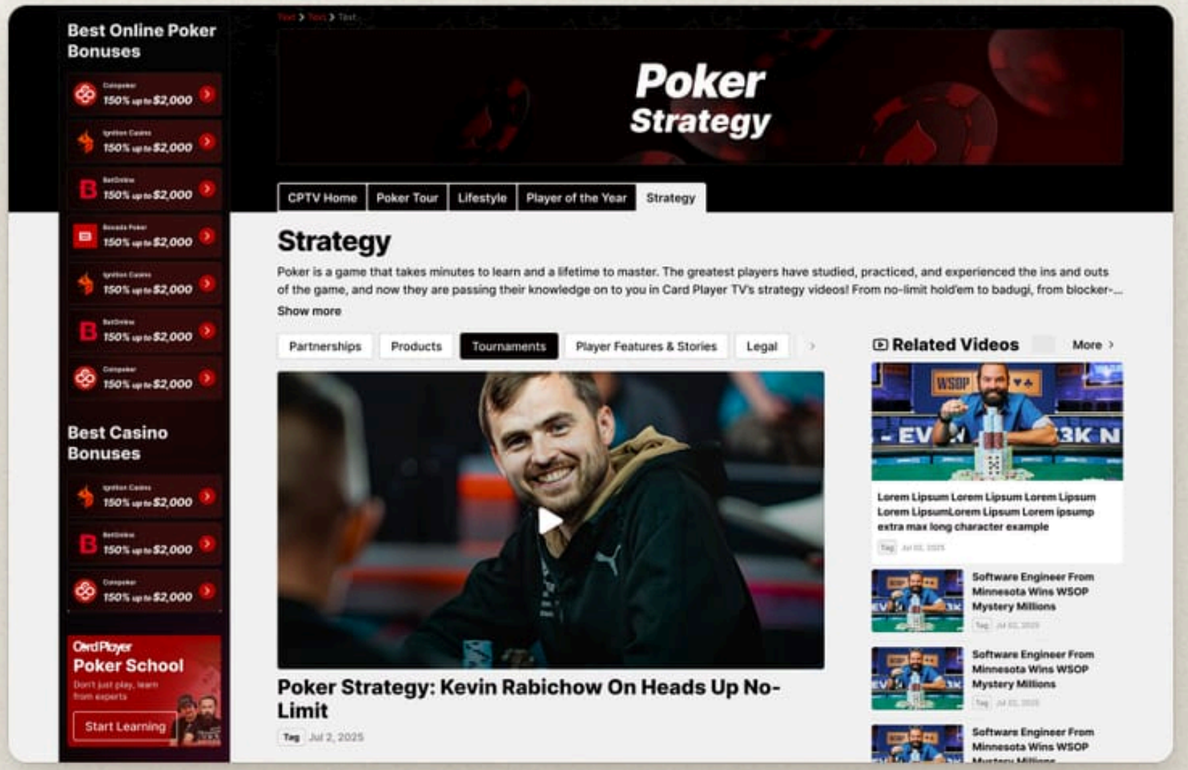
CPTV: 1 constrained skeleton

CPTV existed through several separate layouts, fragmenting the experience and creating a high maintenance surface. I consolidated CPTV into a constrained layout system that reuses canonical patterns, with key entry points and navigation triggers (tour, videos, highlights) standardized within one scalable model.

Original site not my work · 3 linked pages



CPTV template Strategy tab



Governance & Delivery Efficiency

Jira tracked dev tickets but no one had scoped the design workload: no t-shirt sizing, no effort points, no epic breakdown. This kind of program planning is standard for engineering but was never applied to the design side. Rather than creating a parallel epic in Jira, I built the tracking layer in Google Sheets, the format the SEO Director already used to review project progress.

Macro tracking	Google Sheets tracker, effort points XS-XXL.
Effort-based reporting	Progress measured by weighted workload, making re-iteration costs visible to stakeholders.
Figma file structure	Emoji-based naming convention for WIP.
Component scalability governance	3-4 organism variants = keep, explosion risk = atom/molecule swaps.
Alignment loops	Scope, priority and “definition of done” sessions with Product SEO and PM.

Page	Iterations	Size	Progress	Status
Homepage				
Homepage	11	XXL	70%	Rework after design review
Content				
Poker Blog - Archive	1	M	100%	Waiting for Validation
Poker Blog - Single Archive	2	S	80%	Rework after design review
Poker Blog - Single Post	1	XS	100%	Waiting for Validation
Magazine - Articles	2	M	100%	Waiting for Validation
News - Articles	4	XL	60%	Rework after design review
Press - Archive	3	S	70%	Rework after design review
Press - Articles	2	XS	100%	Waiting for Validation
Magazine				
Current Issues	2	XL	50%	Rework after design review
Archive	2	XL	50%	Rework after design review
Single Magazine Page	2	XL	50%	Rework after design review

Templates	Effort points	Progress	High priority
71	**227**	**66.2%**	**76.7%**

Desktop tracker, weighted progress · rebuilt from my tracking sheet

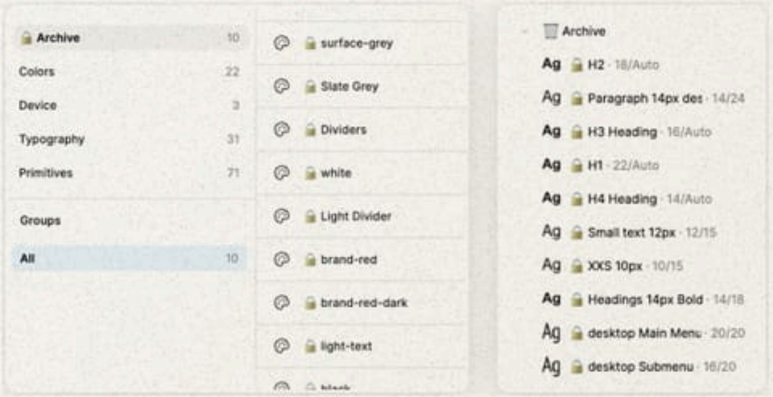
Figma file structure

On screen

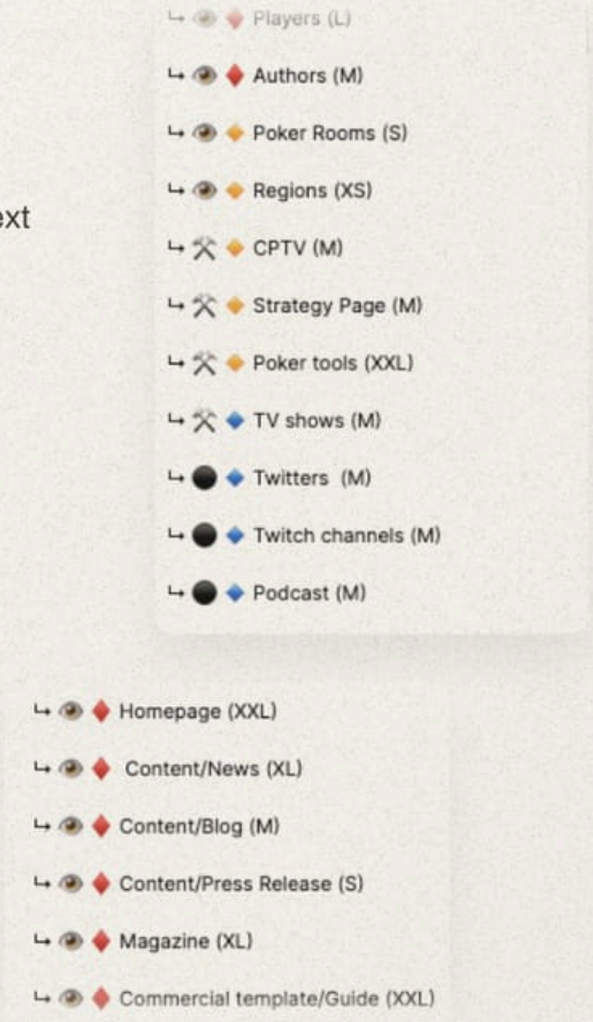
The page list of my Figma file and its legend: status, priority and t-shirt size of every page. Below, legacy variables and text styles, locked.

What changed

I implemented an emoji-based naming convention to make WIP instantly readable in the page list, and locked legacy styles and variables to signal “do not touch / not for redesign use”.



Legacy collection · locked



Figma page list · status, priority, size

Status

NOT STARTED

IN PROGRESS

WAITING FOR VALIDATION

Priority

HIGH LEVEL

MEDIUM LEVEL

LOW LEVEL

Component scalability governance

On screen

One table of my Figma file, from the atom up: 2 Cell types, the header and row molecules, then the Master BigTable on a finished page.

What changed

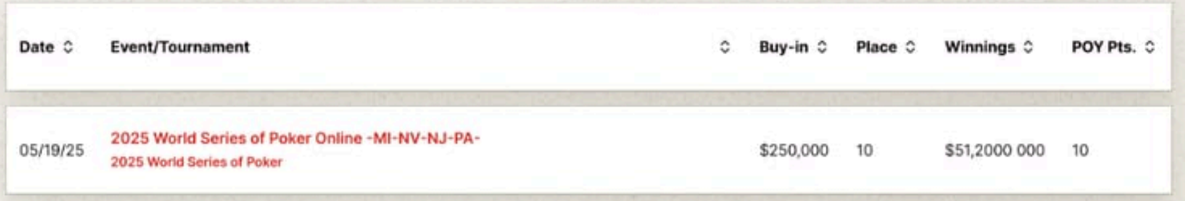
At CardPlayer scale, organism design can explode into variants and slow down the entire file. If 3-4 variants cover the need: keep controlled organism variants. If variants would explode: build one master and generate variations by swapping molecules and atoms directly in context.

Atom Cell

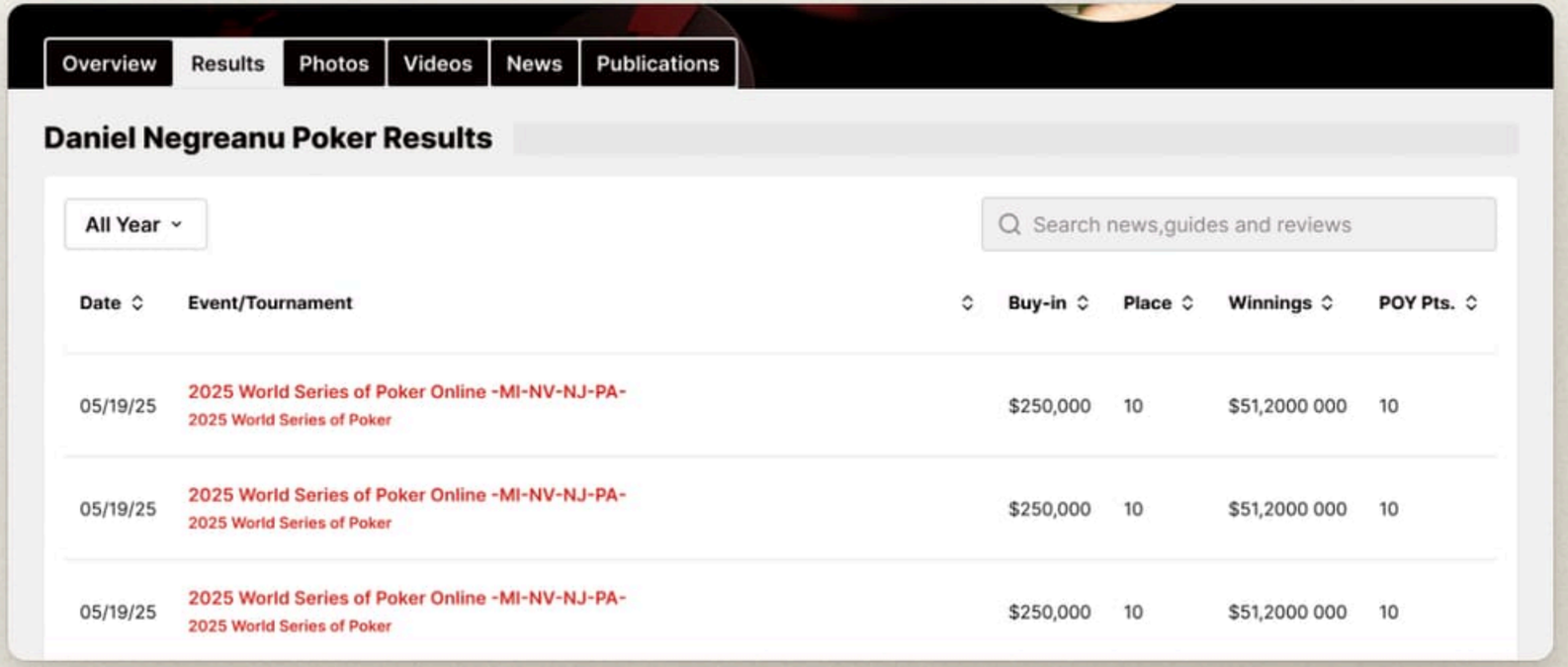


text 2 link

Molecules Tournament/HeaderCell · Tournament/Raw



Organism Master BigTable · on the Poker Tournament page



Key result · CardPlayer, ClickOut Media

9 → 1 archive families consolidated

Template-system migration for an SEO/affiliate business. Solo designer, 5 months.

- 143 templates tracked, 455 effort points (227 desktop + 228 mobile).
- 66.2% of desktop effort completed at handoff, 76.7% on high-priority templates.
- Magazine, Tournament, Region and CPTV reworked into single scalable models instead of parallel layouts.
- Effort-based delivery reporting: progress measured by weighted workload (XS-XXL) rather than raw page count.
- Growth surfaces added beyond brief: Google News CTA, newsletter hooks, missing SEO templates.



ClickOut Media



Poker Simulator Architectural Rebuild

A PM asked me to “adjust a few screens.” I found a poker simulator with broken game logic, no design system, and an architecture that couldn’t scale past 5 players. I reframed the brief into a full architectural rebuild, designed a responsive layout system, built the UI Kit from scratch, corrected all poker rule violations, and delivered executable dev specs.

My role	Senior Product Designer (Solo Designer)
Crew	Solo Designer, 1 PM, Dev Team
Problem	Unblock a revenue plugin by rebuilding architecture, correcting game logic, and delivering specs a dev team with no poker knowledge could implement.

Mission 01 — From “Adjust Screens” to Full Rebuild — 5 sub-missions

Mission 02 — Scalable Layout 3-9 Players — 4 sub-missions

Mission 03 — Dev Specs That Eliminate Guesswork — 4 sub-missions

Mission 04 — Poker Rule Compliance at System Level — 4 sub-missions

From “Adjust Screens” to Full Rebuild

The PM’s request was narrow: make room for more players. A quick audit showed the build was wrong at every level: layout, interaction logic, game mechanics, visual hierarchy. It had been built by a team with no poker domain knowledge, no competitive benchmarking, no design system. 5 failure categories documented before any redesign began.

Layout failures	Saturated at 5 players, controls below fold.
Game logic errors	Check available when a bet existed: illegal.
Visual design failures	Suits indistinguishable, no scan path.
Brief reframe	PM aligned on a full rebuild within 48h.
Competitive benchmark	Good and bad patterns from 4 poker leaders.

Original build not my work



Rebuilt River stage, 9 players



Poker Demo plugin on Odds Data · mobile, same height

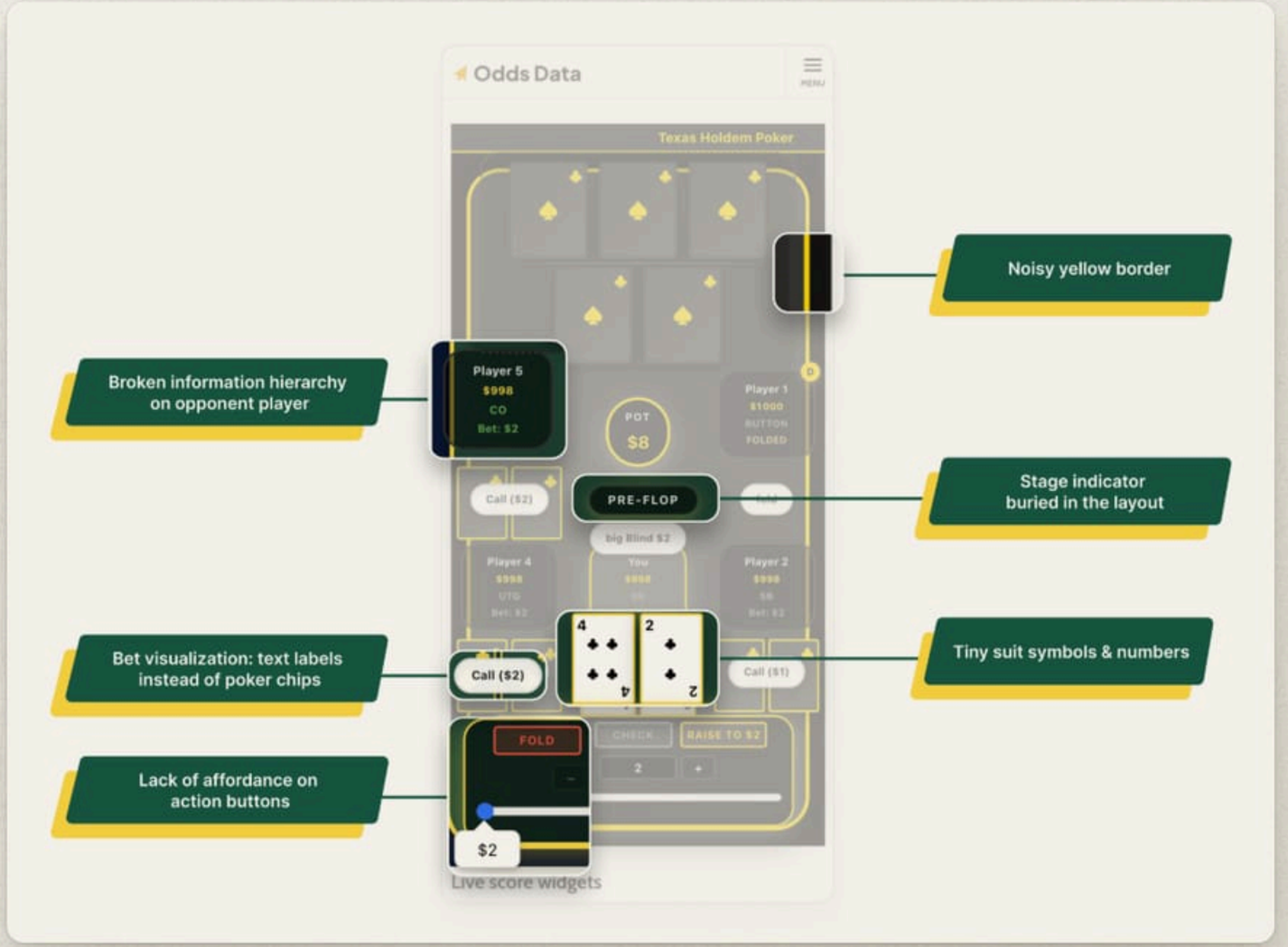
The build I audited

On screen

The original build, with 6 of the failures I documented: yellow borders, broken hierarchy on opponent players, a buried stage indicator, text labels instead of poker chips, tiny suit symbols, action buttons with no affordance.

What changed

This became the case for a full rebuild. Brief reframed from “adjust screens” to full architectural rebuild, PM aligned within 48h.



Original build, before my work · the annotations are mine

Scalable Layout
3-9 Players

The rebuild started from the constraints: 9 players, a full board, a pot, betting controls and a raise slider on a 375px wide screen, inside a scrollable web article. Real poker apps are native and own the full screen. This plugin did not. Every configuration, from iPhone 6 through desktop, needed to be validated at worst-case asset density.

Stress test first	Worst-case density validated first.
Responsive Screen playground	Master component, layouts per breakpoint.
Redesigned UI kit	208+ symbols: cards, tokens, modules.
Raise slider pivot	Scroll conflict caught, pivoted to horizontal.



Desktop table · 9 players, worst-case asset density

3, 6, 8 and 9 seats, at worst-case density

In the original build, seat positions were hardcoded, not adaptive. I built all seat configurations as explicit frames, not assumptions.

3 seats



6 seats



8 seats



9 seats



Mobile stress-test screens · the 4 seat configurations, same scale

Raise slider pivot

On screen

The raise slider in the same River screen: vertical in my initial design, then horizontal.

What changed

A vertical raise slider is the standard pattern across every competitor in the benchmark. But on a scrollable page, a vertical swipe triggers page scroll, not slider movement. I caught the conflict during prototyping, before it reached development, and pivoted to horizontal: a novel pattern, no competitor had solved this because none ship as embedded web plugins.

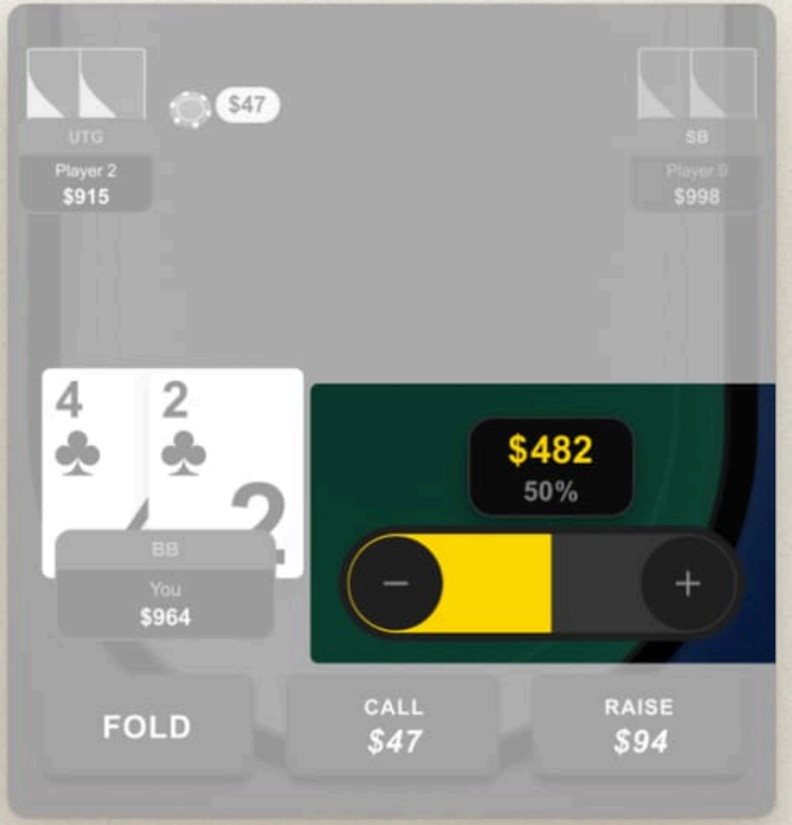
Initial design

Vertical slider



Pivot

Horizontal slider



Lower half of the River screen · slider zone in color

Dev Specs That Eliminate Guesswork

Static mockups document what should happen, not how it should feel, and the dev team had no poker domain knowledge. Every interaction needed to be specified explicitly to avoid implementation errors: targeted interactive prototypes, each isolating one interaction layer, built as executable specs the dev team could implement against.

Game simulation prototype	Pre-Flop to Flop: chips, glow, fold timing.
Designed intuitive gestures	For new patterns interactions.
Drafted animation specifications	Card reveal, raise slider.
Prototype to support the handoff	Zero interaction left to interpretation.



Scenarios board, Figma · OLD and NEW per street, Development annotations

Game simulation prototype

A curated sequence of key moments, Pre-Flop to Flop: bet chip logic, action glow, fold timing, buttons visible only on the hero's turn, pot merge at the end of the street.

Table initialized



Hero's turn



Fold timing



Pot merge



Flop



GameSimulation prototype · 5 of its 17 frames, in order

Poker Rule Compliance at System Level

A poker simulator shipping incorrect rules teaches wrong behavior and destroys brand credibility. The original build had no formal state architecture. Every game state, button configuration and pot display was specified and validated frame by frame against Texas Hold'em rules.

Pot display logic	Central pot = resolved bets, label = total.
State-driven buttons spec	4 action panels mapped to game conditions.
Complete hand lifecycle	Pre-Flop to Hand Result, State 0 and State 1.
OLD vs NEW comparison	Delivered for every street.



Development · State 1

All Bets Placed (Pre-Pot Merge)

Definition

All active players have matched the current bet. No pending action remains for this betting round.

Rules

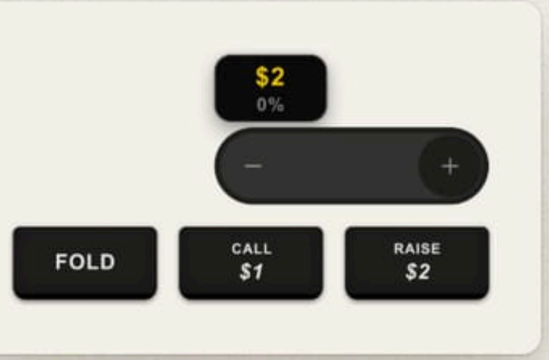
- Individual bet chips remain displayed in front of each player
- Each bet shows the exact amount committed by that player
- Pot value reflects the **sum of all bets**, but chips are not yet merged
- No action buttons available
- Table is visually in a "resolved but not progressed" state

Scenarios board, detail · 1 Pre-Flop frame, its Development annotation

State-driven buttons spec

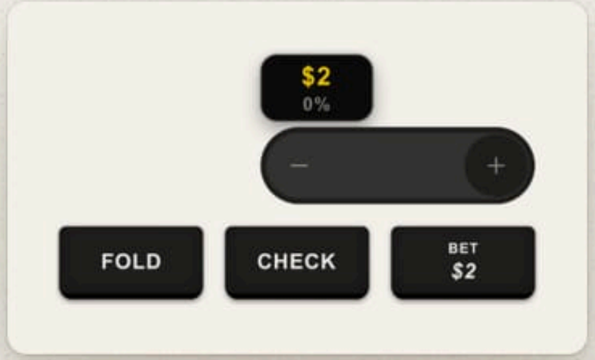
Which buttons appear on screen is not a UI decision, it is a poker rule. The dev team implements a state lookup, not a chain of assumptions.

Against Bet



At 0%, the slider equals the minimum legal raise above the current bet.

Against No Bet



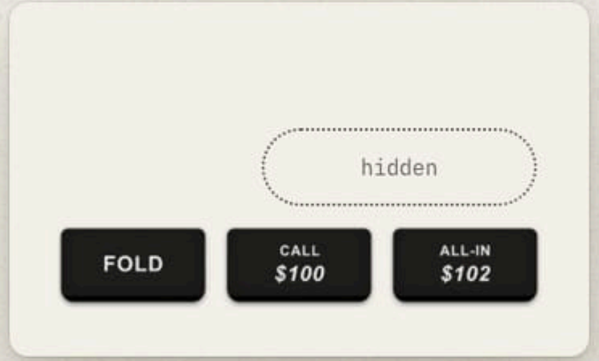
Against no Bet the slider equals the bet amount. The primary action switches from Bet to Raise once the slider exceeds the minimum bet.

No Raise



Player stack is insufficient to perform a legal raise. Only Call (all remaining chips if required) or Fold are allowed. Raise action and slider are hidden.

Partial Raise



All-In replaces Raise when stack < minimum legal raise. Betting is not reopened.

Action panel · 4 states, same frame · rules from my Development cards

OLD vs NEW, street by street

For each street, I set the original build next to my new spec. Here the Flop: bets become chips, buttons follow the game state.

OLD not my work



NEW



Scenarios board, Flop column in detail · after check

OLD not my work



NEW



Facing a bet · OLD = original build · same scale per pair

Key result · Poker Simulator, ClickOut Media

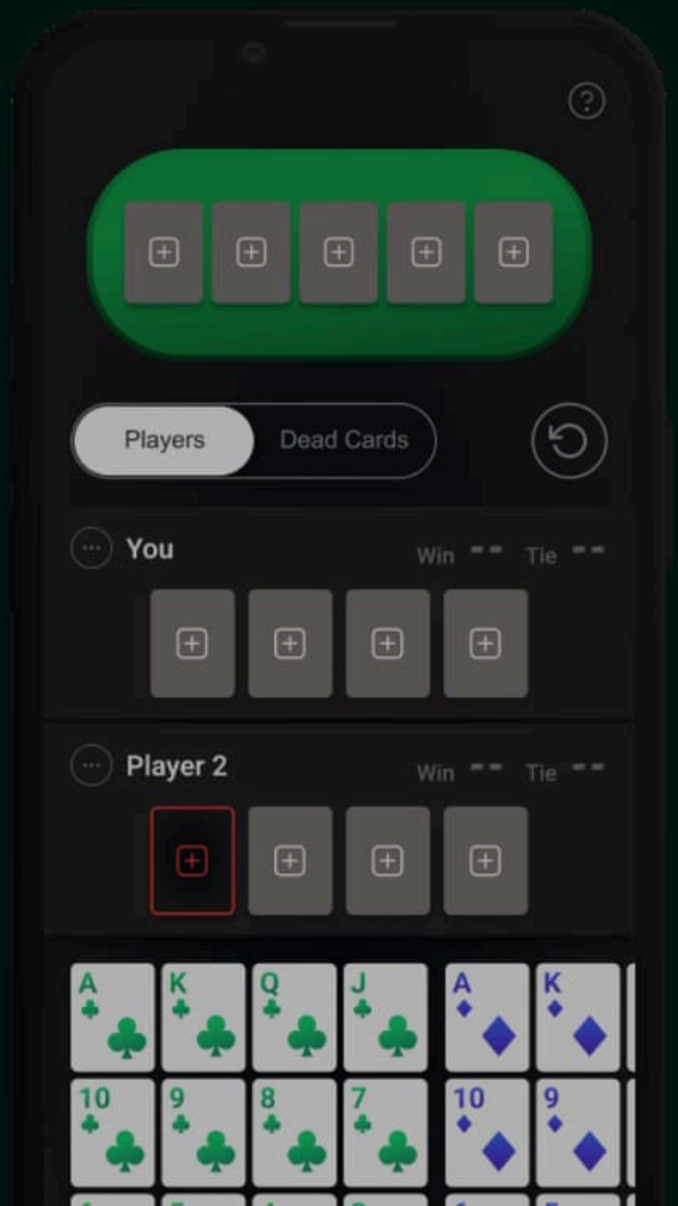
1.5 months to production

As solo designer. Domain expertise from BetRivers applied throughout.

- Rescued a broken prototype and shipped a production-grade poker simulator, on mobile and desktop.
- Reframed the brief into a full architectural rebuild, PM aligned within 48h.
- Covered 14 game states with zero poker rule violations.
- Benchmarked against 4 market leaders before launch.
- Scalable layout system: 3 to 9 players, validated across 3 mobile viewports and desktop.
- UI Kit built from scratch: 208+ symbols.



ClickOut Media



Poker Odds Calculator
SEO Growth Tool

PokerScout's revenue is 100% affiliate-driven, but editorial content was missing high-intent tool queries. The brief: ship a poker odds calculator covering all game modes, fast. I designed one component system serving 5 poker variants as an SEO entry point, an engagement layer, and an affiliate conversion surface.

My role	Senior Product Designer (Solo Designer)
Crew	Solo Designer, Product SEO Director, PM, 3 Engineers
Problem	Capture high-intent organic traffic across 5 poker variants that editorial pages cannot rank for, with one scalable component system and implementation-ready specs.

Mission 01

Interaction Design & Edge Cases

5 sub-missions

Mission 02

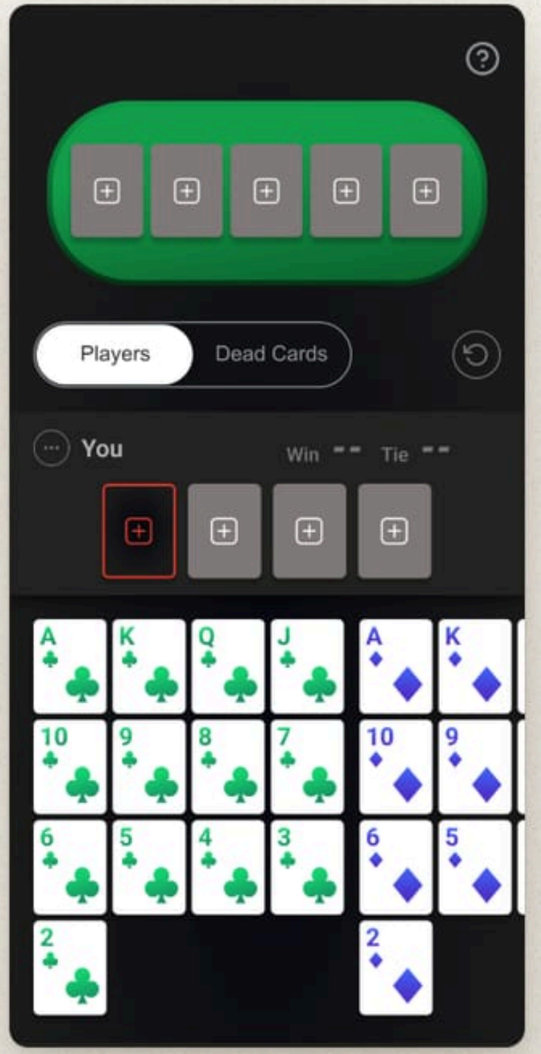
Cross-Platform Delivery & Dev Handoff

2 sub-missions

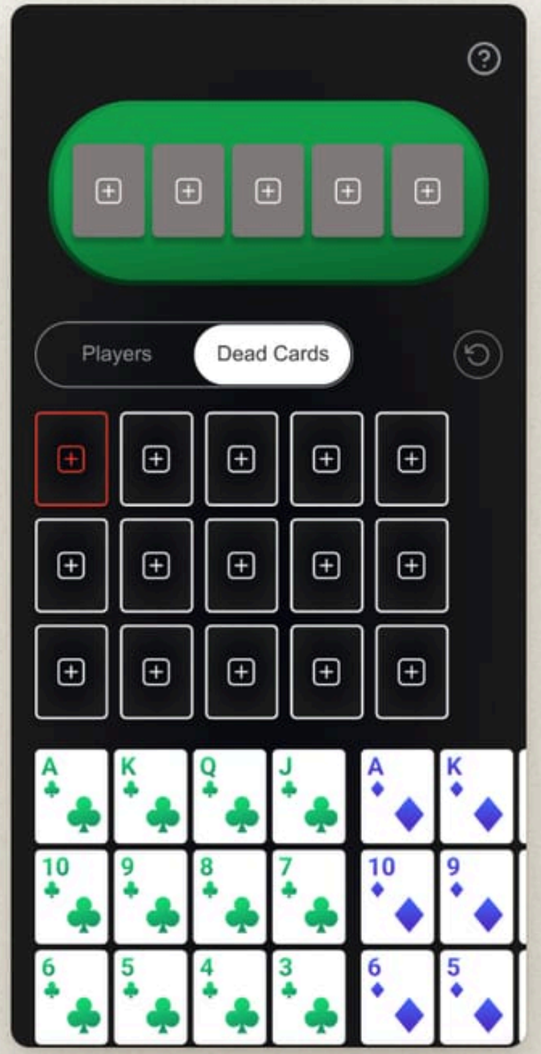
Interaction Design & Edge Cases

Poker players use tools one-handed on mobile, between hands, under time pressure. The UX needed fast card selection, a clear Win/Tie % output, and robust error handling across 5 variants with different rules. Every edge case (minimum players, maximum per variant, dead cards, reset) needed explicit handling with clear feedback.

Component architecture	1 system, 5 poker variants, not 5 tools.
Designing card selection	A full 52-card visual grid, no modal.
Focused output	Win % and Tie %, the only output metrics.
Dynamic player management	Toast feedback on every edge case.
Dead Cards management	Mobile tab, permanent row on desktop.



Players tab · deck inline



Dead Cards tab · 15 slots

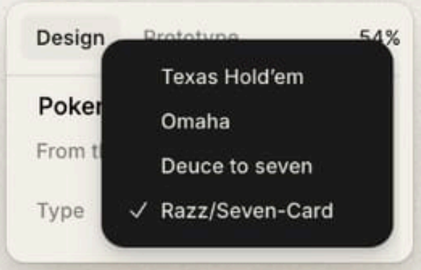
1 system, 5 poker variants

On screen

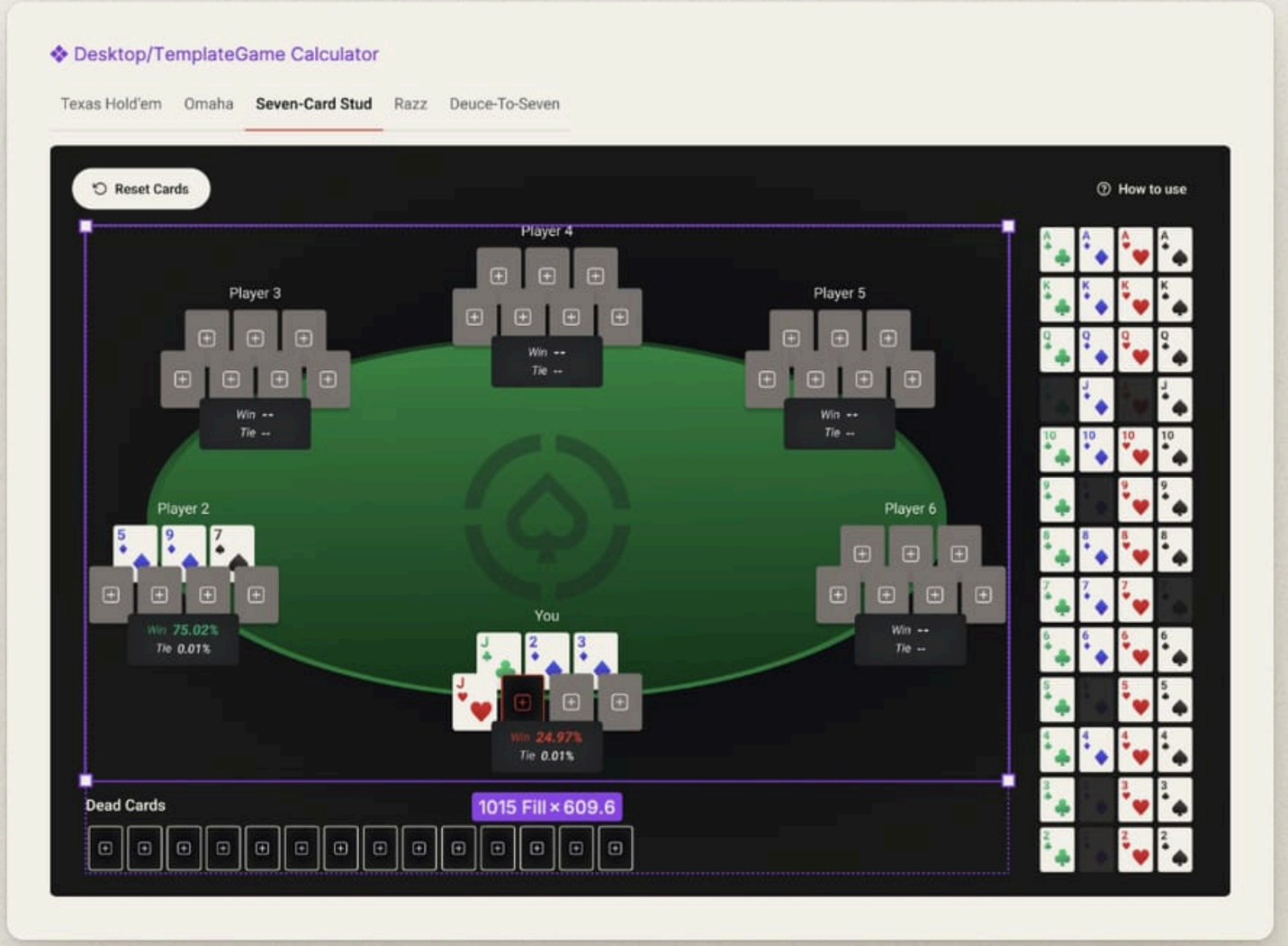
The calculator template in my Figma file, set to Seven-Card Stud. Its Type property, open below, switches the table. 5 tabs, 4 table types: Razz and Seven-Card Stud share one.

What changed

PokerScout covers 5 variants editorially. Building 5 separate tools would multiply dev cost and design debt. One component library in Figma serves all 5 variants through variant properties, not duplicated frames.



Same capture · Type property, 4 values

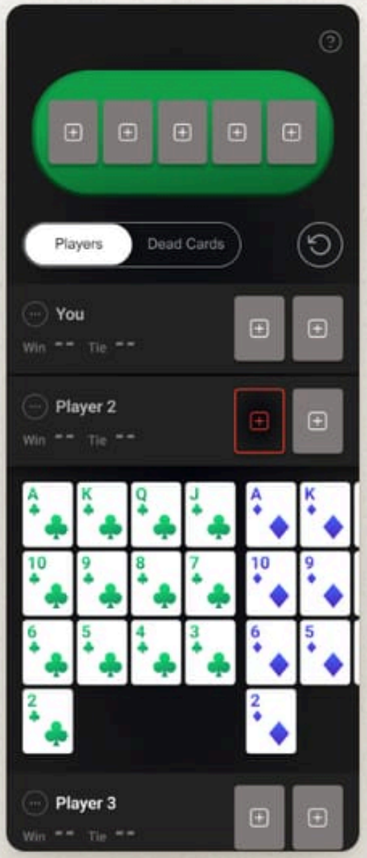


Figma · Desktop/TemplateGame Calculator, Seven-Card Stud tab

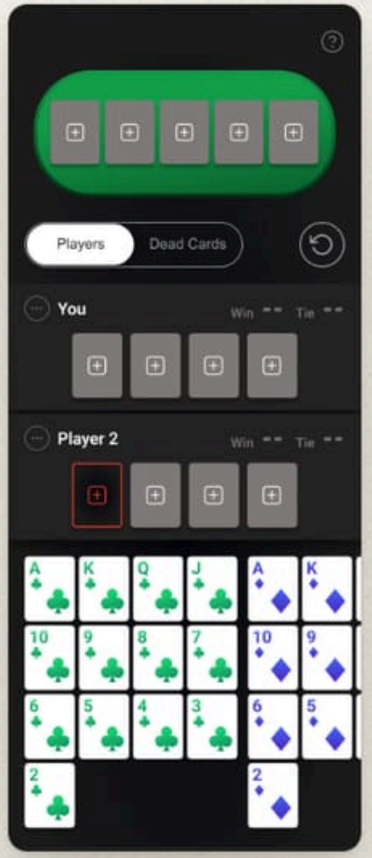
5 poker variants, 1 component system

Player hand slots, board card slots, and deck all adapt to variant rules.

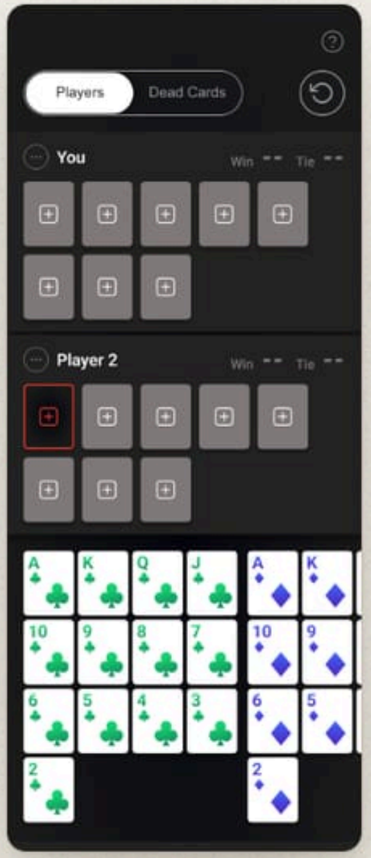
Texas Hold'em



Omaha



Seven-Card Stud



Razz



Deuce-To-Seven



Mobile calculator · the 5 variant tabs, same scale, same window

Designing card selection

On screen

The desktop calculator in Texas Hold'em: 10 seats, the board, and the full 52-card grid beside the table. Cards already dealt are greyed out in the deck.

What changed

The dominant competitor pattern is a two-step selection flow; on mobile, a full-screen modal that blocks visibility of the table entirely. Here the deck grid sits permanently beside the table: no overlay, no panel switch. The user sees odds update in real time as cards are selected.

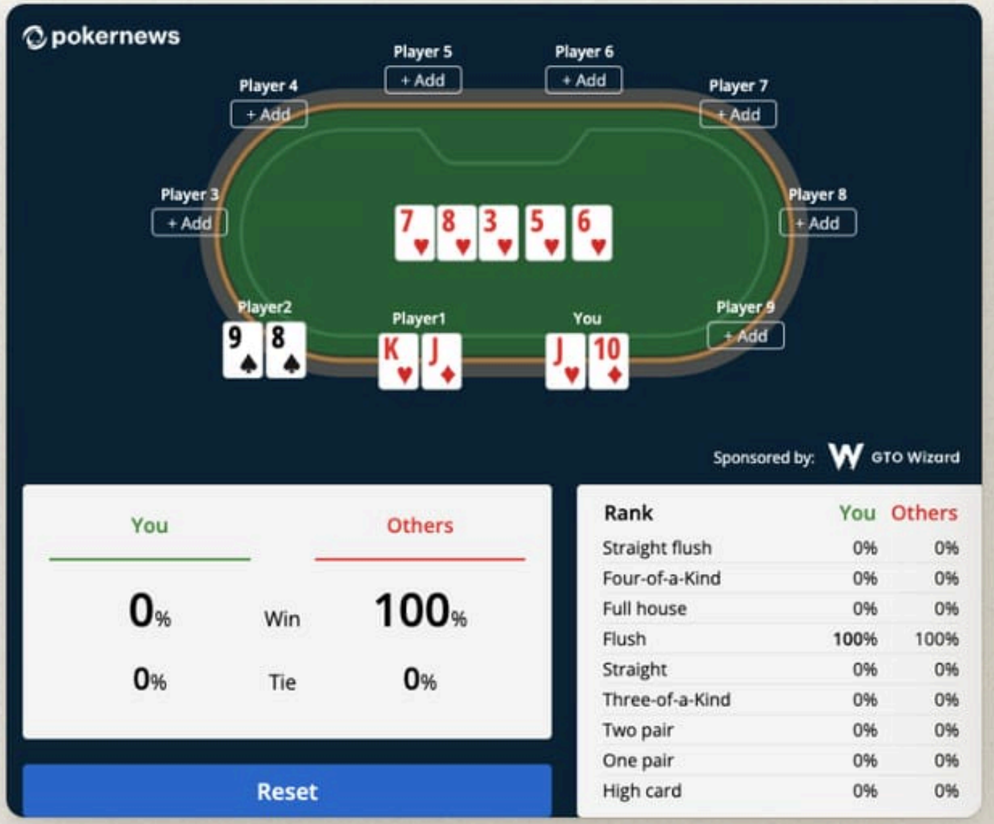


Desktop calculator · Texas Hold'em, 10 seats, Dead Cards row, 52-card grid

Focused output: Win % and Tie %

The output needs to answer one question fast: “what are my chances?” Win % and Tie % are the only output metrics, under each hand.

PokerNews competitor · not my work



Poker Odds Calculator Win % and Tie % under each hand



A table of hand ranks to cross-reference with the board

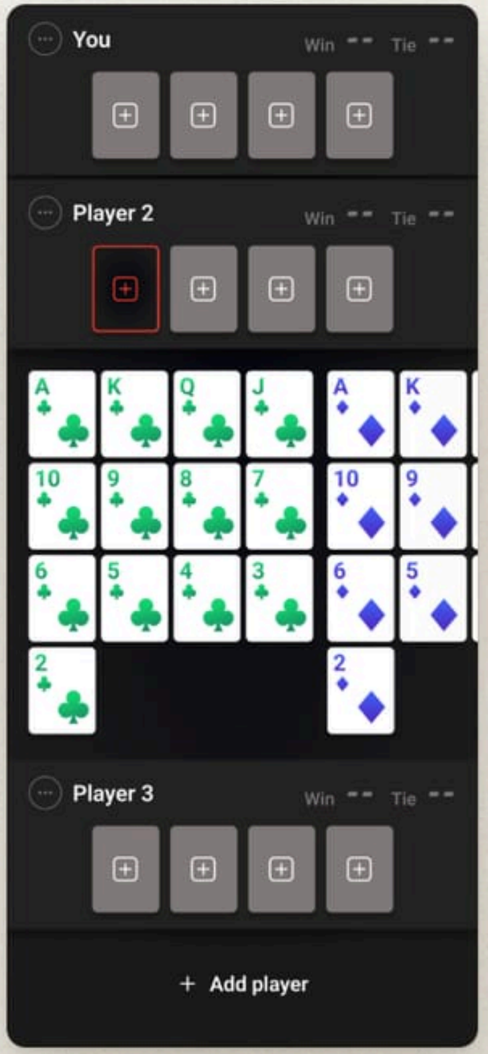
Dynamic player management and error handling

On screen

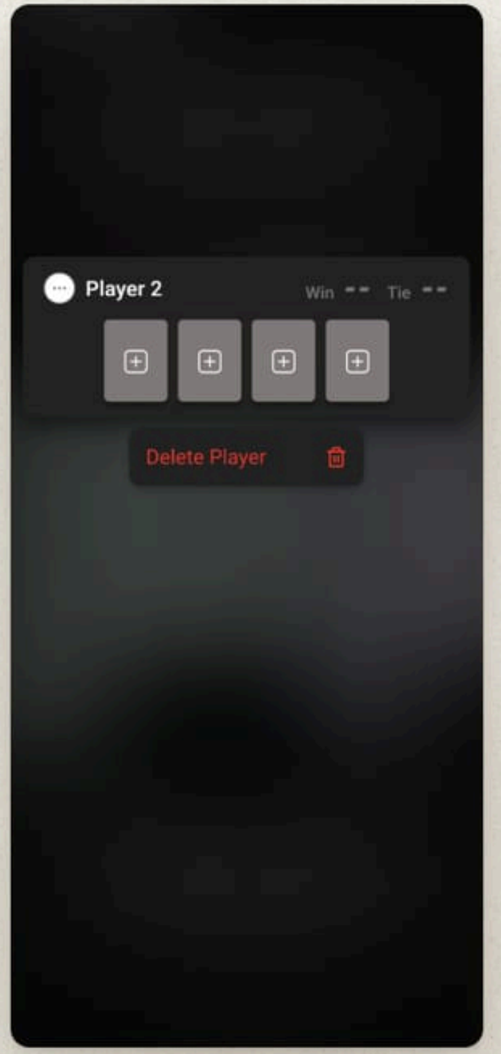
Removing a player on mobile: the three-dot menu, “Delete Player”, then the row collapses and “Add player” comes up. With only 2 players left, the removal is blocked and a toast says why.

What changed

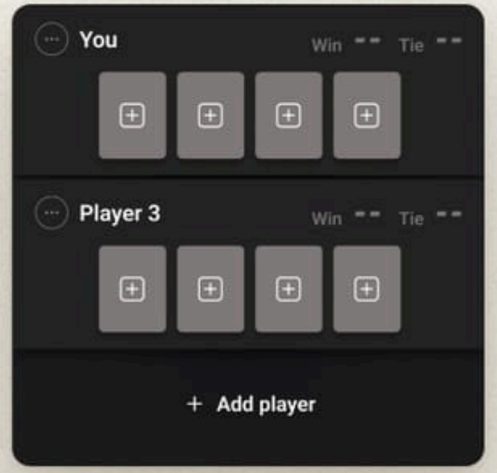
Three-dot contextual menu per player row replaces bare X icons, creating spatial separation from card slots and eliminating miss-touch risk. Minimum 2 players and maximum per variant enforced with toast notifications.



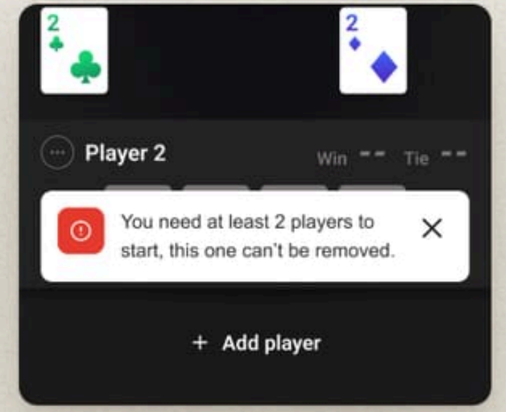
1 · Player list



2 · Menu open, list dimmed



3 · Row collapsed



4 · Below 2 players: toast

Cross-Platform Delivery & Dev Handoff

The tool needed to work across 3 breakpoints (desktop, tablet, mobile), with every interaction spec'd precisely enough for a dev team 6+ hours away to implement without synchronous overlap. The conversion layer (affiliate modules) needed to integrate seamlessly into the full PokerScout page, not bolted on as an afterthought.

Dedicated design per device	Desktop, tablet and mobile, each with component sizes and states adapted to the available space.
Animation specs for dev handoff	Every interaction documented with trigger, animation type, curve, and duration.

Specs: Select and Remove a Card

How to play prototype :

- Click the Jack of Hearts in the deck to add it to the player's hand
- Click the Jack of Hearts in the player's hand to remove it and return it to the deck
- Once a card is selected, the selection focus automatically shifts to the next card

Play prototype

Expected behavior :

- Click on a card (from the deck list on the right)
→ adds it to the player's hand
→ disables the card in the deck list
→ bouncy animation (Smart Animate, 800ms)
- Click again on the card (in the player's hand)
→ removes it from the hand
→ re-enables it in the deck list
→ same bouncy animation

Animations

- Trigger: On click
- Animation: Smart animate
- Curve: Bouncy
- Duration: 800ms

Spec card, desktop · 1 per scenario

Card selection/ deselection
Smart Animate
Bouncy · 800ms

Deck appearance
Smart Animate
Ease Out · 400ms

Dead Cards tab switch
Smart Animate
Ease Out · 400ms

Board card selection
Smart Animate
Bouncy · 800ms

Contextual menu
Fade + Slide Up
Dissolve · 300-400ms

Player removal
Collapse
Ease In · 400ms

Toast notifications
Fade + Slide Up
Ease Out · 300ms

My animation table

Dedicated design per device

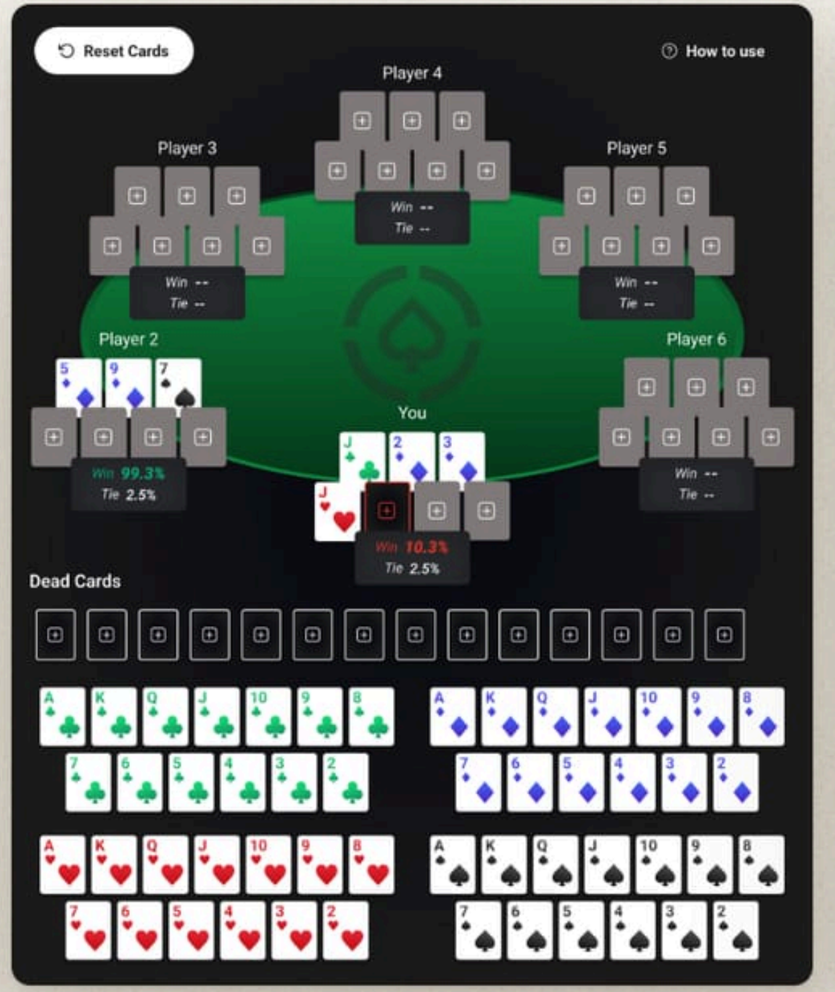
Each device received its own dedicated design, with component size variations and states adapted to the available space.

Desktop



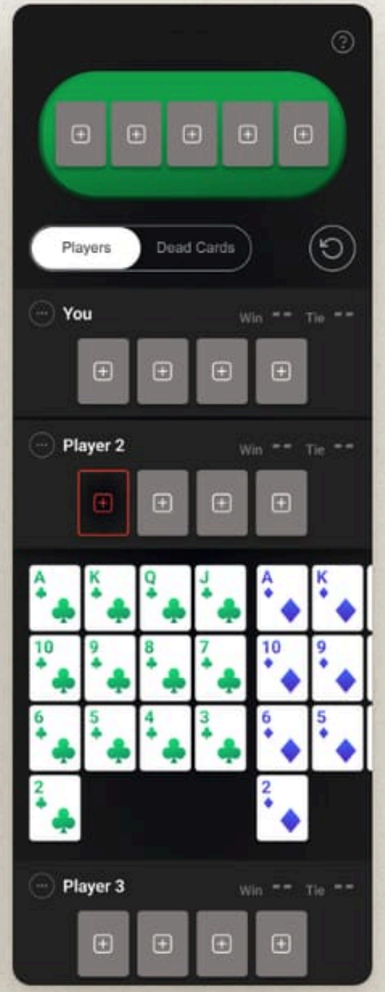
Deck beside the table

Tablet



Deck under the table, grouped by suit

Mobile



Deck inline

Key result · Poker Odds Calculator, ClickOut Media

~60% estimated dev cost reduction

1 system, not 5 separate tools. Live within 2 months, as solo designer.

- SEO growth tool delivered in 2 months. Shipped across 5 poker variants and 3 breakpoints: desktop, tablet, mobile portrait.
- Dead Cards management shipped as differentiator: none of 4 benchmarked competitors offer it.
- Async-first handoff across a 6h+ timezone gap: spec sheets with interactive prototype links per scenario.
- 1 component system serving 5 poker variants.
- Affiliate conversion surface at peak intent, targeting 2-4% CTR vs 0.5-1.5% industry standard.





Product visual · IDEMIA

Case study · IDEMIA, 1 month, 2 sprints

FCODE
Tablet App Design

F.CODE is the first biometric payment card: a fingerprint sensor on the card replaces the PIN. The cardholder has to enrol their fingerprints on a tablet in a bank branch. If it fails, the card is just a normal card.

My role	UX Designer. My mandate was the branch enrolment app.
Crew	1 Lead designer, 2 UX, 1 PM, 1 Dev fullstack
Problem	Build an application to enroll users' biometric fingerprints on a payment card. The first round of tests showed that not a single user could get through it unaided.

Mission 01	Find out what the product actually does	4 sub-missions
Mission 02	Co-construction workshop	4 sub-missions
Mission 03	Prototyping	2 sub-missions
Mission 04	Test	4 sub-missions
Mission 05	Analysis	3 sub-missions
Mission 06	Presentation of results	3 sub-missions
Finding 01	A third round, in October	3rd test round
Finding 02	The problem nobody fixes with a screen	3rd test round

Find out what the product actually does

An MVP had already been done by the developer, against a specification written by the project manager. The two of them did not describe the same journey, and neither description matched what the screens actually did.

Ergonomic audit

Of the existing MVP interfaces.

Existing user path

Checked against the project manager's specification.

Userflow

Every path of the solution mapped, each with its screens.

Experience map

The experience as lived by the users.

Problems detected

Several paths were missing. Many inconsistencies in their sequence.

Co-construction workshop

I ran a workshop on Mural, presenting the userflow and the experience map to the project manager and the developer at once.

Technical specifications
Understood with the developer.

Improved workflow
The current workflow, improved in the room.

Paths to test
Defined before any scenario was written.

Wireframes to produce
A list of the wireframes to produce.

Outcome

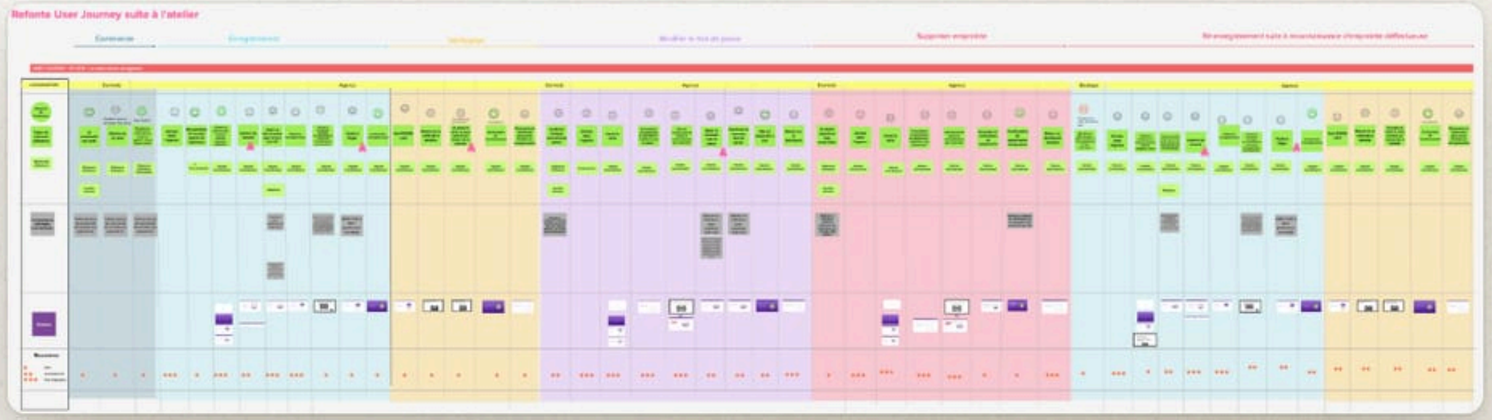
The first thing I produced was not a screen.

It was a single journey, from walking into the branch to holding an active card.

Why it mattered

Getting those two to agree on one path is what made the rest measurable.

The journey, redesigned after the workshop



My Mural experience map, in French · overview

On the board

6 phases, from ordering the card to re-enrolling a finger. For each step: place, feeling, resources, screens.

	Phase	Where	Steps	Point of vigilance for user tests
01	**Order**	Home	3	none
02	**Enrolment**	Branch	9	Reading the tutorial · Finger capture
03	**Verification**	Branch	5	Contactless tap on the tablet's reader
04	**Change the password**	Home, then branch	9	Entering the new password
05	**Delete fingerprint**	Home, then branch	8	none
06	**Re-enrolment** after faulty recognition	Shop, then branch	14	Reading the tutorial · Finger capture

The 4 paths of the solution



My Mural userflow, in French · overview

4 wireframes to produce

What do we do with your data?

Checking that your card works

Fingerprint recognition

Your card seems to have a fingerprint recognition problem

01

New user

Enrolment, then verification of the card.

02

User already enrolled

2 branches: deleting the fingerprints, changing the password.

03

Card with a recognition problem

Enrolment again, then verification.

04

Admin configuration

3 screens.

What the workshop put on the board

16 sticky notes above the first path, 17 more in a “Topics covered” frame.

01	**Run an A/B test: with the tutorial, and without.**	Enrolment path
02	**Check the limits of the sensor.**	Enrolment path
03	**Is there another check of the card, with insertion this time?**	Enrolment path
04	**The user must stay in the flow and test the card immediately.**	“Topics covered”
05	**Check that my fingerprints were really recorded, in case my card does not work.**	“Topics covered”

Prototyping

Once all the paths were validated, I built a design system on Idemia's graphic charter and an interactive prototype in Figma. I am not authorized to show the mock-ups themselves.

Design system

Built on Idemia's graphic charter.

Interactive prototype

In Figma, on the validated paths.

One flow, specified for 4 platforms

IDScreen	Idemia's own device
Android SDK	in the specification
Windows SDK	in the specification
Windows 10 build	in the specification

Test

The Figma prototype was tested in guerrilla sessions. The people who will fail on a biometric payment card are not a segment, they are everyone.

Test guide
Objectives, scope, scenario, missions, post-test questions.

Guerrilla test
Short sessions. It buys speed, and it costs representativity.

Panels
Small and deliberately varied.

Recorded with consent
Confidentiality form signed, every session recorded.

5–7

Protocol

people between 18 and 65, from outside Idemia

5

June round

testers

From a receptionist of 51 to a student of 20, an analyst and an engineer.

6

October round

more testers

All external to Idemia.

Analysis

I processed my observation grid in Excel, as atomic research, and defined the recommendations for each problem found in the user tests.

Excel observation grid
1 row per tester, 1 column per screen, 3 scenarios.

Atomic research
My observations processed as atomic research.

Recommendations
Defined against each problem found.

Hypothesis to verify, scenario 1	Tester 1–5	Green
Inserting the card	□□□□□	0 / 5
Physical discomfort or pain during enrolment	□□□□■	1 / 5
Position of the finger on the card	■■■□■	4 / 5
Checking that the card works	□□■■■	3 / 5

From my Excel grid, June round · filled: cell in green on the grid

7 usability problems found, from unclear wording to physical pain

Enough to show that

- The tutorial was read as decoration
- The guidance between steps was missing
- The sensor hurt to hold the way the screens implied

01	The majority of users were unable to complete the main scenarios independently
02	The tutorial was misinterpreted
03	Confused functionality
04	Lack of guidance in the solution
05	Physical pain encountered during handling
06	Lack of confidence in the security of their data
07	UX writing sometimes too technical or imprecise

Problems detected, first test round

Presentation of results

I implemented my recommendations on the Figma prototype and ran a second test round on the same scenarios, with new users.

Recommendations implemented
On the Figma prototype.

Second test round
Same scenarios, new users.

3 fixes
A rewritten tutorial, clearer guidance between steps, plain wording.

First round · users completing the enrolment unaided

0/5



Second round · same scenarios, new users

3/5



Same task, same 5 minutes

A third round, in October

Protocol

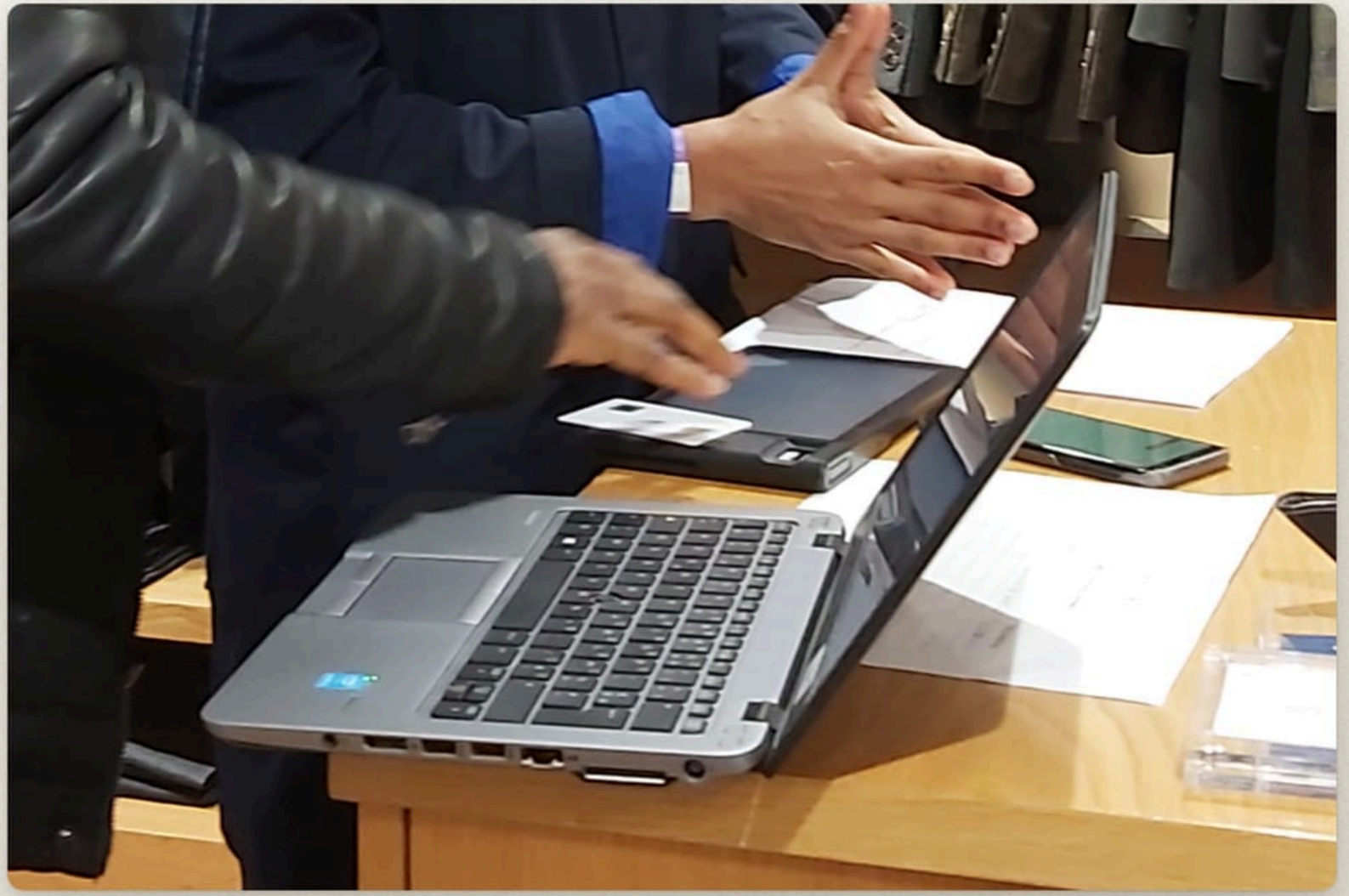
Four months later, 6 people from outside the company, in French rather than English, on all 4 scenarios.

Result

■■■□□□ **3 of the 6** could not find where to insert the card.

■■■□□□ **3** put their finger on the tablet instead of on the card.

The lowest score of the panel was 2 out of 6.



Test session at a shop counter, 27 October 2020 · the card in the enrolment device

The June result did not hold

My reading

I do not read that as a regression in the design. The June testers were a small group who had met the product in an office; the October ones were strangers meeting a bank terminal for the first time.

Lesson

**June tested my fixes.
October tested the product
on strangers.**

The problem nobody fixes with a screen

In October the dominant subject was not usability. It was trust.

One tester	Practical but scary, and I would not buy a card like that.
Another	Refused to put her finger on the card, afraid her fingerprints would be taken.
A third	Worried about what a state could later demand of the banks holding them.
Several	Doubted that deleting the fingerprints really deletes them.
One of them	For me the fingerprints are in the app.

Its core promise, fingerprints that never leave the card, was not reaching the people asked to trust it.

Key result · FCODE, IDEMIA

0→3/5 users enrolling unaided

Fingerprint enrolment, first test round to second. Same task, same 5 minutes.

- 7 usability problems found, from unclear wording to physical pain.
- A single journey, step by step, from walking into the branch to holding an active card.
- The mock-ups and the design system went to the project manager and the developer, and the work was carried over to other devices.
- F.CODE went on to be certified by Visa and Mastercard, then launched with banks in several countries: Rocker in Sweden, FAB and MAF in the UAE, manager.one and Sella.





Press photo · IDEMIA and Interpol

Case study · IDEMIA, for Interpol

Research without the software

MBIS is the biometric search engine IDEMIA built for criminal investigation, and the subject of a partnership with Interpol made public in 2021. The buyer and the user are not the same person, and only one of them has to live with the interface.

My role	UX Designer · led the second phase of a UX study on MBIS
Sample	2 framing interviews, 5 sessions, 1 examiner each
Problem	The brief was to improve the interfaces. The constraint was that I would never be allowed to use them: 0 minutes of hands-on access, by classification, for the whole engagement.

Mission 01 — Understand the work without touching the software — 3 sub-missions

Mission 02 — Let five experts rebuild the interface — 3 sub-missions

Mission 03 — Read what they ranked — 2 sub-missions

Mission 04 — Deliver, and push on the method — 3 sub-missions

Understand the work without touching it

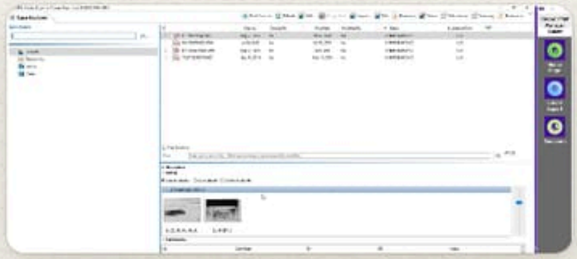
IDEMIA gave me a filmed session: an examiner working two cases, commenting as he went.

I turned a recording into a journey map, frame by frame
Read image by image: 6 steps, 20 pages.

I cut the interface down to one tool per card, so experts could rebuild it
5 examiners, 1 session each.

The brief was interface quality. The problem was trust in automation.
From a question about screens to a question about evidence.

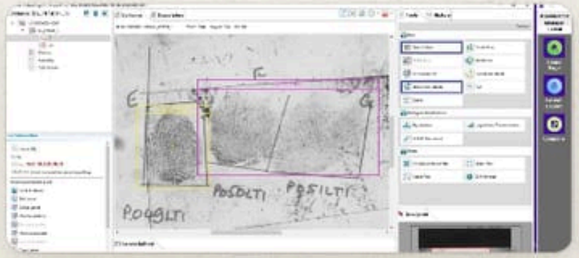
01 · Open the case



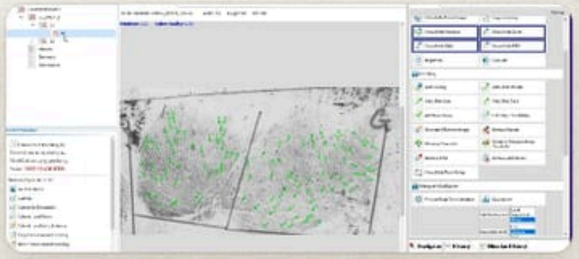
02 · Prepare the evidence



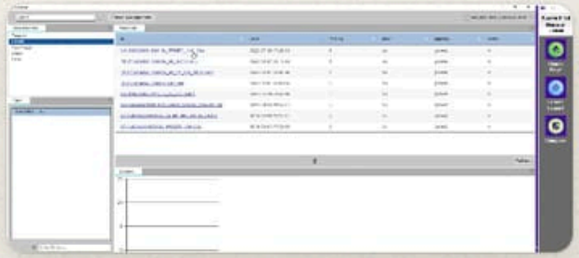
03 · Isolate the trace



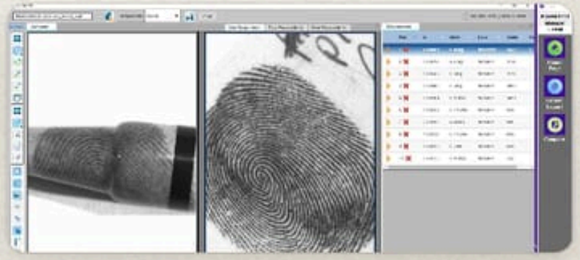
04 · Enhance and encode



05 · Submit and wait



06 · Compare and decide



The whole of my access to MBIS · original software, demo dataset

So I rebuilt it instead

On screen

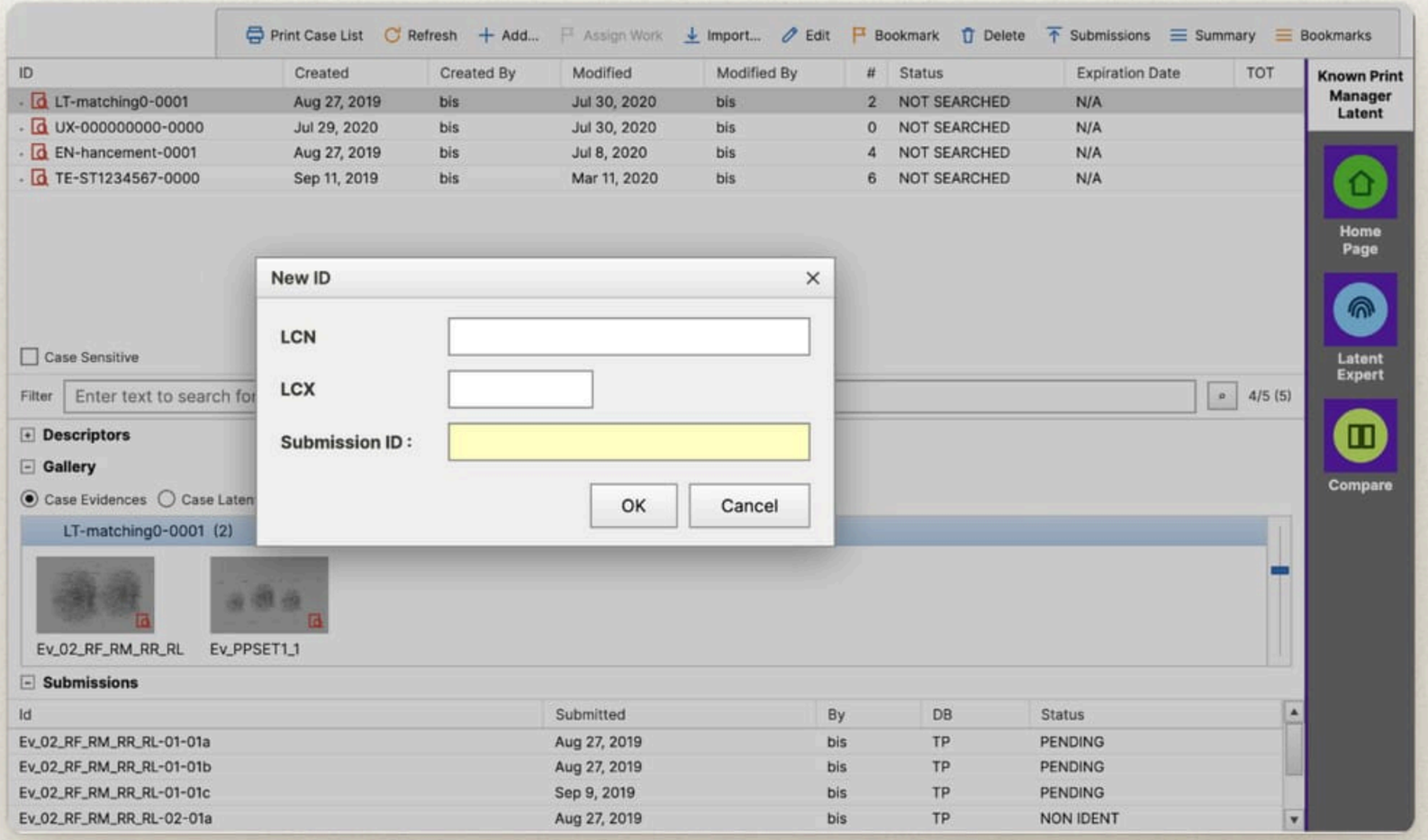
Step 01, Open the case. A dialog stands between him and the case, and it never says what it wants. He filled every field, was refused 3 times, and never learned which rule he was breaking.

What changed

20 pages of observation, written from a recording, became 6 screens. Each step frames the part of the screen that matters, at a size the software never gave the examiner.

What it cost

A second-hand understanding. I could never test a hypothesis myself, never try the alternative path, never check whether an irritant was a one-off or a pattern.



Reconstruction, built from a filmed session · not the original software

The brief was interface quality. The problem was trust in automation.

They struggled to know what the software had done to the evidence on their behalf, and whether they could still defend the result.

Asked for

Improve the interfaces

a question about screens

→ What they described

Trust in what the software did

a question about evidence

In their words

One of them called a powerful filter “magic”, and meant it as a warning, not a compliment. Another avoided a correction tool rather than risk altering the print beyond what would hold up.

What it cost

A recommendation the sponsor had not asked for, which took longer to land than a list of screen fixes would have.

Let five experts rebuild the interface

5 one-to-one video calls, 1 examiner at a time, each on their own Mural board: a puzzle of the interface, 1 piece per tool, re-sorted from most to least important and useful. Separate sessions give 5 rankings to compare, not 1 negotiated consensus.

5 independent sorts converged on the same 4 tools to bring forward
5 people who never met put the same 4 tools on top.

The sort is not research output. It is a prioritisation
3 bands of cards, 3 product decisions.

How each session ran, in 3 parts
Talk first, then open the board, draw at the end.

	Rank given by each examiner				
Tool	E1	E2	E3	E4	E5
kept by all 5					
Minutiae history	**1**	2	**1**	2	**1**
Navigator, zoom, image properties	2	**1**	3	4	3
Encoding	3	4	2	**1**	4
The original image	4	3	4	3	2
dropped by all 5					
The right-hand menu	○	○	○	○	○
Other traces in the case	○	○	○	○	○

One ranking, cut twice. Each cut is a decision.

stays in front · 4

Keep, and bring forward

- Minutiae history
- Navigator, zoom, image properties
- Encoding
- The original image, reachable at any point

stays, but not first · 2

Step back

- Obsolete functions sitting at the same level as daily ones
- Everything a screen shows at once, into progressive disclosure

the product stops carrying it · 3

Drop

- The right-hand menu
- The other traces in the case, which nobody asked to see
- Tools nobody had ever opened

The harder half

Deciding what to remove. Adding costs a sprint; removing costs an argument with whoever shipped it. 5 independent rankings are what made that argument winnable, because nothing in the removal list was my preference.

How each session ran, in 3 parts

I wrote an intervention guide first: context and stakes, objectives, the hypotheses under test, and the protocol that would answer them.

01

Talk first, screen later

Nothing of MBIS on screen yet, on purpose: open the board first and every answer calibrates itself on what is visible.

02

Then open the board

The interface, cut to the individual tool rather than the panel, re-sorted by value. The post-it annotations mattered as much as the sorting.

03

Draw at the end

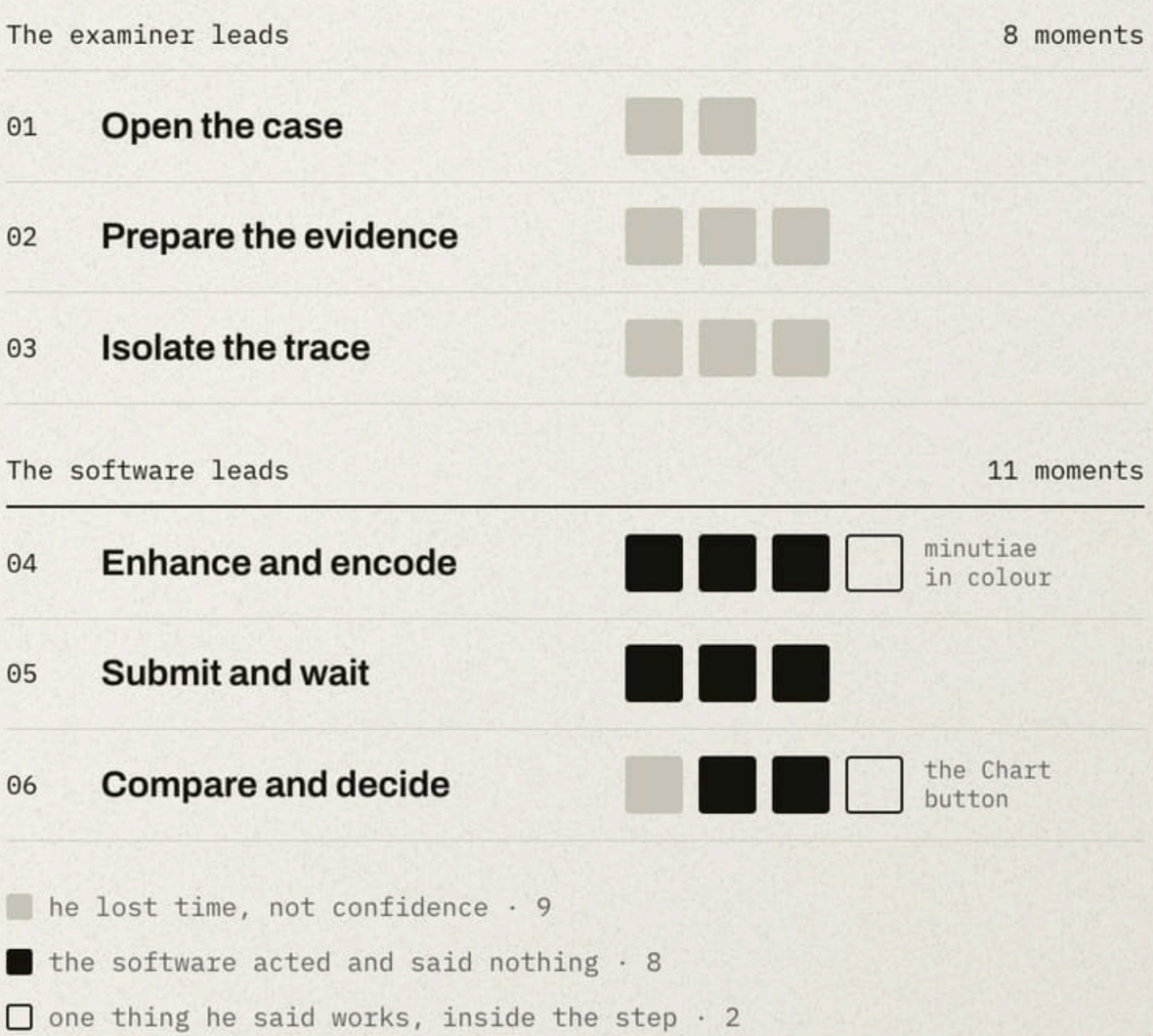
Asking people to design is usually a trap, because you get feature requests. Asking at the end of a session where they have described their own work and ranked their own tools is different.

Read what they ranked

A journey map of a product nobody on the design side had ever used, built from 1 filmed session and 20 pages of notes.

In the 1 filmed session, trust drops where the software takes over
Trust in the tool falls to step 04, then he takes it back.

Almost none of the friction was about how it looked
7 frictions, sorted by what they prevent.



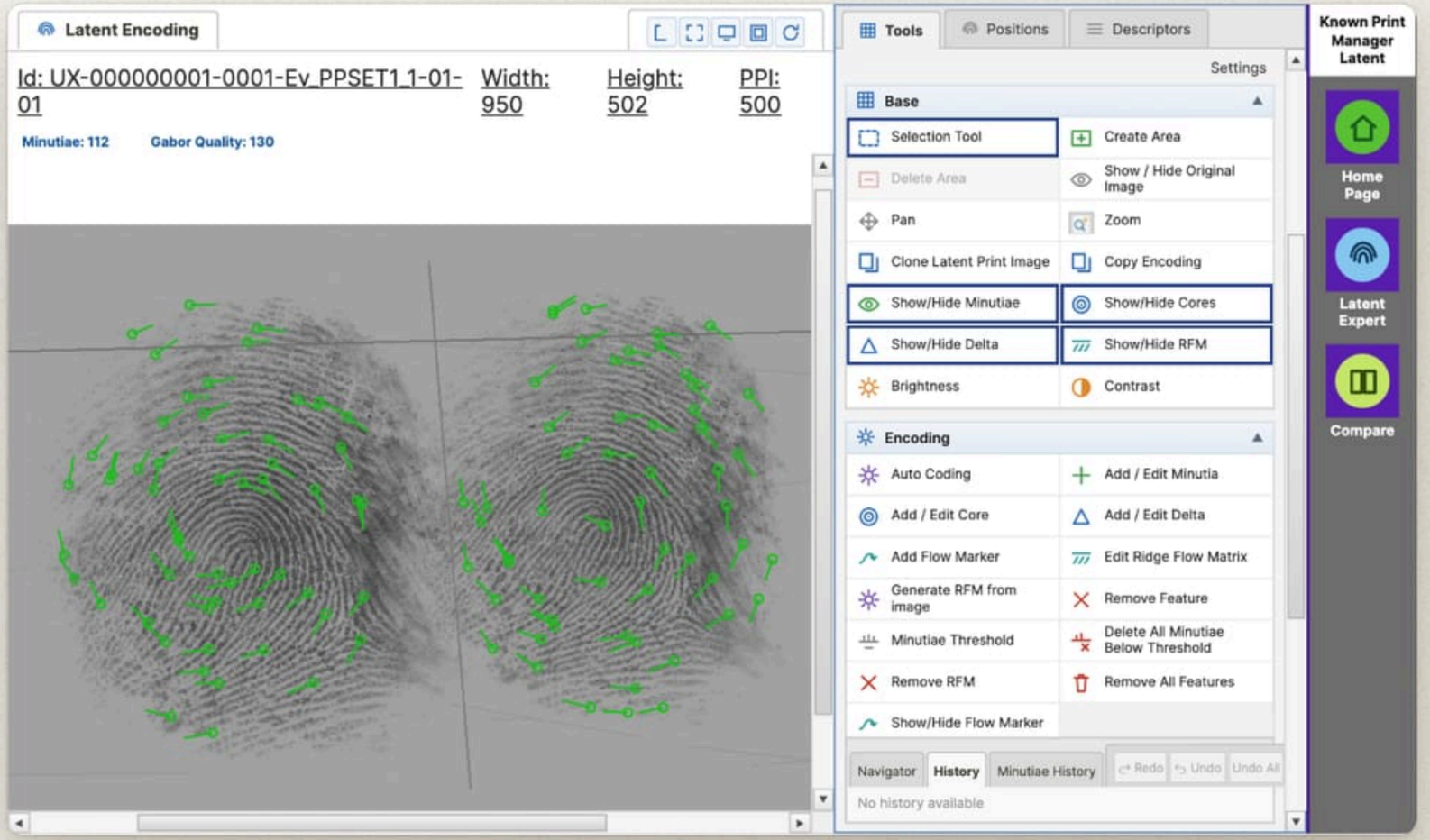
The friction changes nature at step 04

On screen

Step 04, Enhance and encode: the print and its 112 minutiae, and a History that reads “No history available”. This is the break: the software filtered and encoded the print on its own, and nothing on screen says when it acted.

What changed

Before step 04 the friction costs him time. From step 04 the software acts on the evidence without telling him, and he must defend every change in court. That is why the recommendation was not a tidier screen but a separation: fully automatic on one side, fully manual on the other.



Reconstruction · not the original software · step 04, Enhance and encode

7 frictions, sorted by what they prevent

0 about how it looks

Knowing where things are

A movement tool found only after great difficulty, opening a menu nobody knew existed

A history panel that changes position once encoding adds new tools

A settings panel so densely nested it made one examiner want to run away

Knowing what went wrong

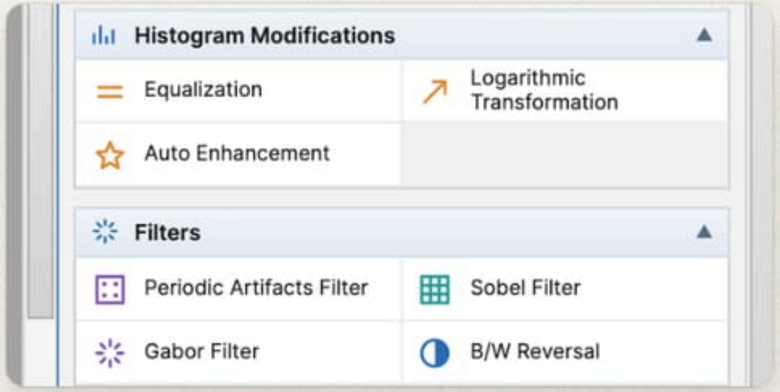
Errors that say a field is wrong without saying what would make it right: 3 attempts, 3 failures

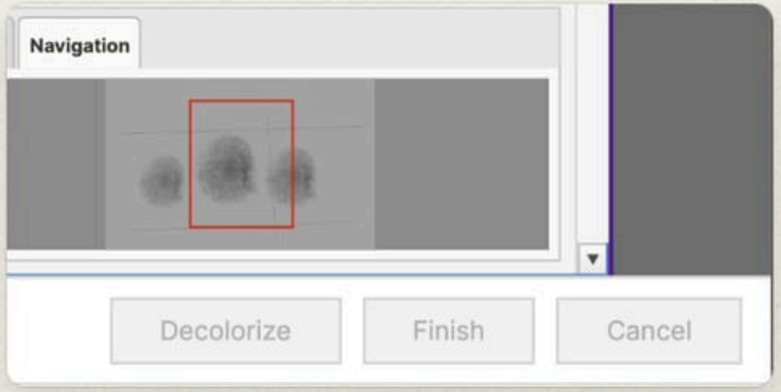
Mandatory fields that only reveal they are mandatory by blocking you

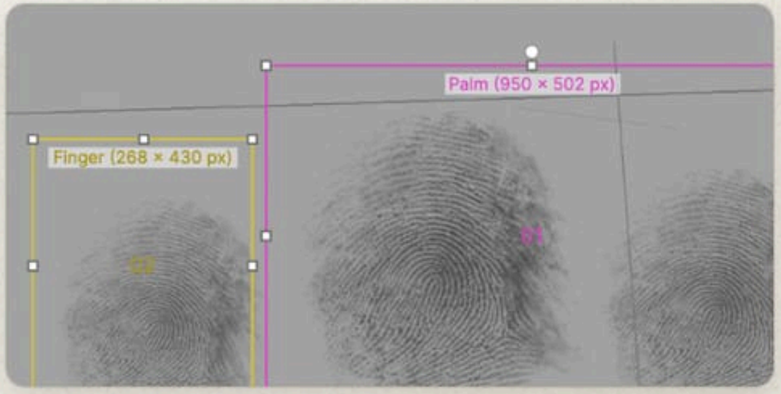
Being able to go back

No visible undo, although the keyboard shortcut works

The original image out of reach for comparison after heavy edits







Reconstruction · not the original software

Deliver, and push on the method

The UX team operated like an internal consultancy: each intervention was billed to the project team that requested it.

Billing the research to the project team bought weeks, never enough to verify
Enough to find problems, rarely to verify them.

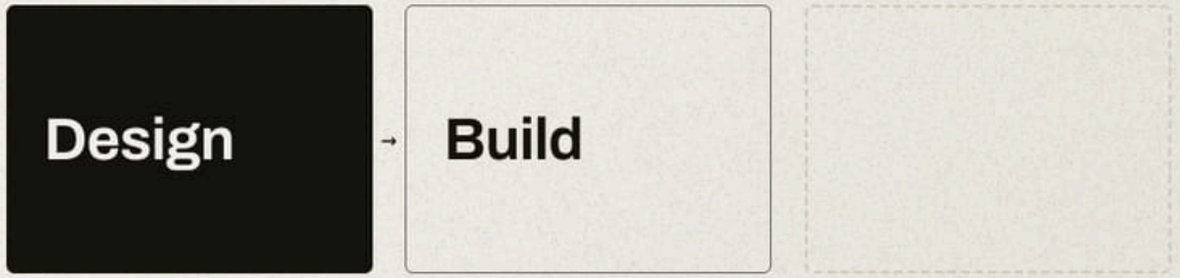
The recommendation went as far as the budget did
The sort gave the order; the recommendation turned it into interface direction.

I would fight for 30 minutes of the software, and I would not deliver in slides again
A deck freezes research the day it is written.

How it ran



What I argued for



5 calls handed to the product manager

Delivered as interface direction with first zonings, grounded in what 5 examiners ranked and said.

Borrowed, not invented

The first interview asked which other software they use, to map the conventions they already live in. Where a feature had an equivalent in a tool they open every day, image editors among them, the recommendation was to borrow that behaviour rather than design a new one. A convention an examiner already trusts costs nothing to teach.

01 **Separate fully automatic from fully manual.**

02 **Make the modification history permanent and ordered.**

03 **Keep the original image reachable at all times.**

04 **Bring the daily tools forward.**

05 **Push the obsolete ones back.**

Key result · MBIS, IDEMIA

5 product calls, each backed by 5 rankings

5 independent expert rankings · 0 minutes of hands-on access

■ Delivered

An ordered set of product calls, delivered to the product manager as a direction and first zonings.

A journey map of a product nobody on the design side could open, usable by whoever picks the work up next.

A prioritisation method for expert tools, reusable on any interface too large to sort by panel.

A reframing the sponsor had not asked for: from screen quality to trust in automation.

□ Afterwards

I argued for atomic research in the team, set up a session between my team and the founder of Glean.ly, then carried the purchase negotiation.





Product visual · IDEMIA

Verinium
Access Terminal

Verinium is a secure biometric authentication product: you enrol your face in a phone app, then authenticate at a terminal to get through your company's door, holding your phone to an NFC reader. The whole product rests on one sequence, done once, by somebody who has never seen it.

My role	UX Designer in R&D, from protocol to handover
Crew	UX team in R&D
Testers	16 across the two waves, recruited guerrilla style
Problem	Check that the instructions are understood, that the app reassures people about what happens to their data, and that somebody meeting the product for the first time can enrol and authenticate without help.

Mission 01	Map the sequence, and find where it can fail	3 sub-missions
Mission 02	Test on the real technology, not on a mock-up	5 sub-missions
Mission 03	Read what the testers did	5 sub-missions
Mission 04	Separate the interface failure from the technology one	3 sub-missions
Mission 05	Run an A/B test, and accept what it said	4 sub-missions
Mission 06	Hand over 26 recommendations, documented	6 sub-missions

Map the sequence, and find where it can fail

An access terminal is not an app you learn. You meet it in a corridor, with people behind you, on a morning when you are thinking about something else. The enrolment happens once. If it fails, you go and find somebody with a badge, and the product has lost its argument.

The sequence	7 steps on the phone, then 2 at the terminal.
9 moments where a person can stop	Somebody who has never seen any of it.
What the test had to check	Instructions understood, data reassurance, enrol and authenticate without help.

01	Tutorial	On the phone
02	Terms	
03	Permissions	
04	Selfie capture	
05	Confirm	
06	Identity valid	
07	NFC activation	
08	Face detection	At the terminal
09	NFC authentication	

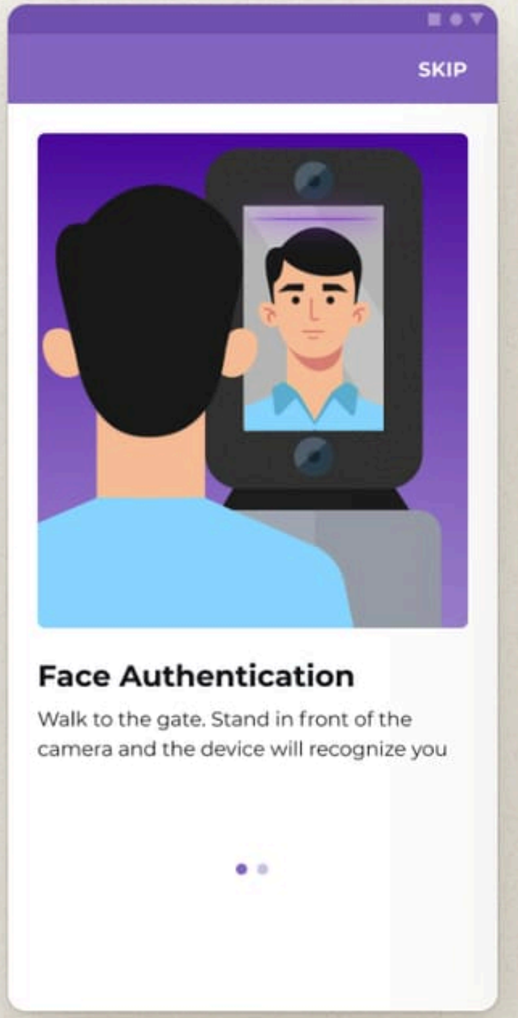
One sequence, done once, by somebody who has never seen it

On screen

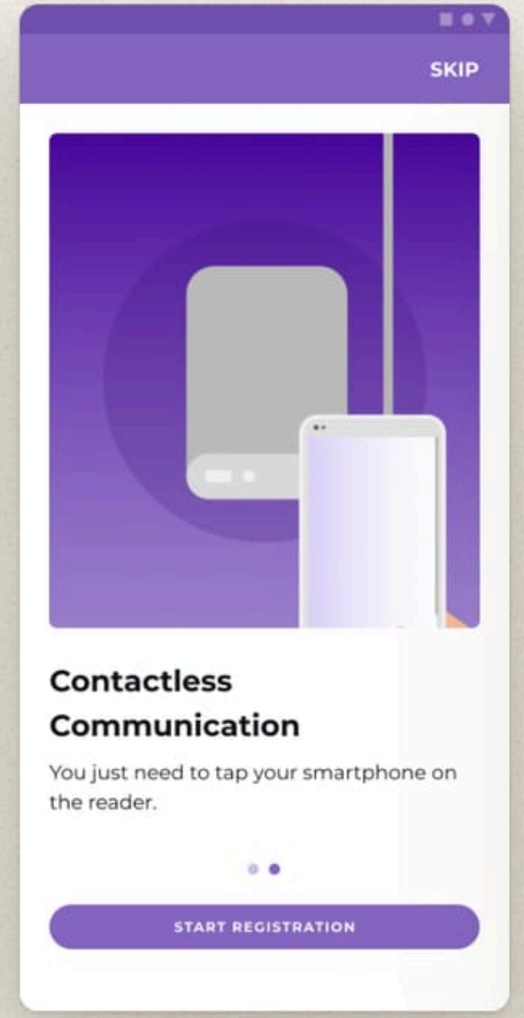
The enrolment on the phone as it was tested: the 2 tutorial screens, then the terms behind a link.

What it showed

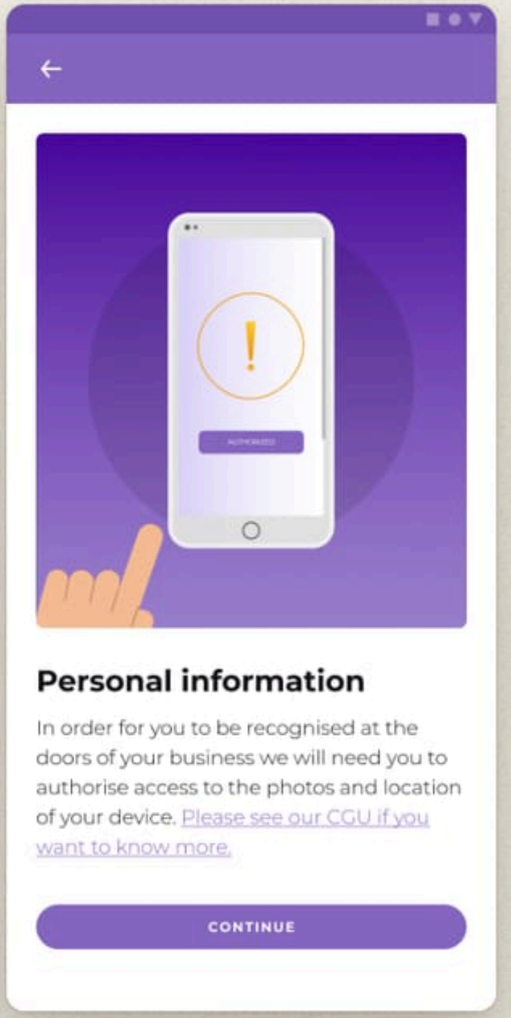
The enrolment happens once. If it fails, you go and find somebody with a badge.



Step 1 · Tutorial, 1 of 2



Step 1 · Tutorial, 2 of 2



Step 2 · Terms

Test on the real technology, not on a mock-up

I put the prototypes on the working technology rather than on a screen mock-up, because face capture fails in ways no prototype reproduces. A prototype always recognises you. The real camera does not.

2 separate set-ups	The full journey, and 1 prompt alone.
6 people on the full journey	From the phone to the terminal.
10 more on the A/B test	The NFC prompt alone, 5 per wording.
Protocol, sessions, grid, audio	I wrote the protocol and ran the sessions. 1 grid per tester, audio with consent.
One bias in the report	English prototype, tests in France.

Full journey

6 testers · 30 min sessions · phone, then terminal

No.	Gender	Age	Job, all at IDEMIA
1	M	35	Financial controller
2	F	27	Buyer
3	M	39	General services, logistics
4	F	63	Lawyer
5	M	34	Project manager
6	M	28	Access control officer

A/B on the NFC prompt

10 testers · 5 per wording

Students, a supervision technician from another firm, a fire safety officer.

Reading grid

4 statuses per observation

Validated	Most testers did what was expected.
To be redone	Most testers did something unexpected.
To be adjusted	About half did what was expected.
To be watched	Unusual behaviour, counts if it recurs.

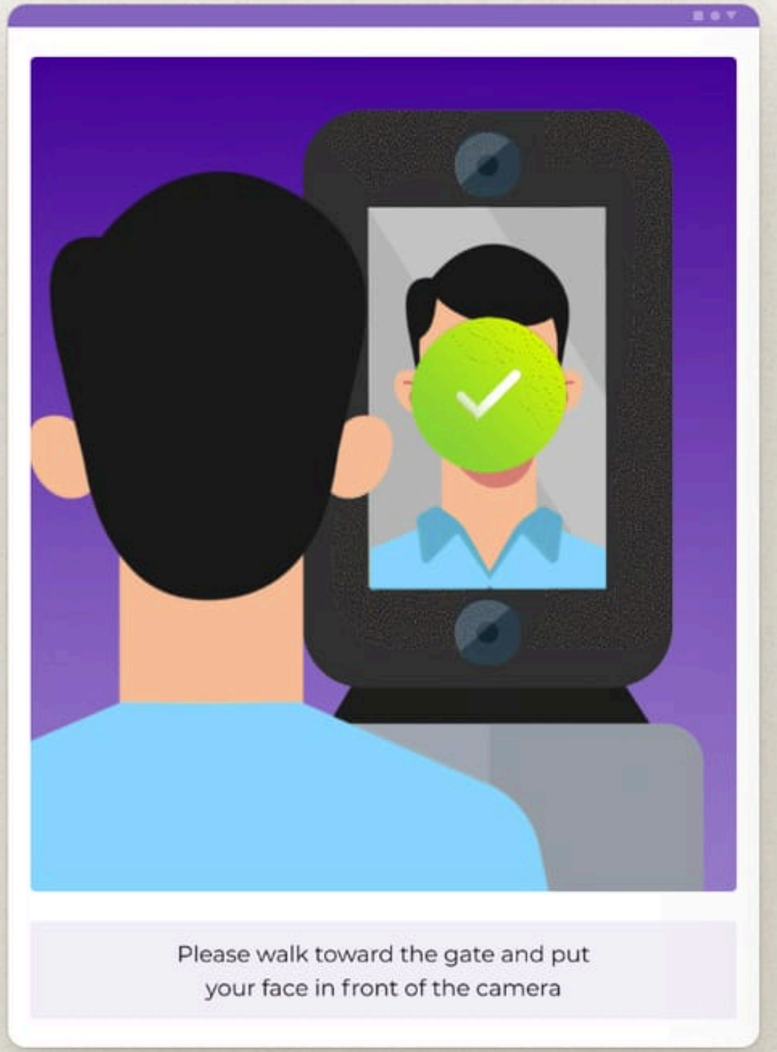
The last 2 steps, on the working terminal

On screen

The 2 terminal screens as they were tested: face detection, then the phone on the NFC reader.

What it showed

2 of 6 said the terminal took the picture before they had time to stand properly. 3 of 6 hunted for the NFC reader on the terminal.



Step 8 · Face detection

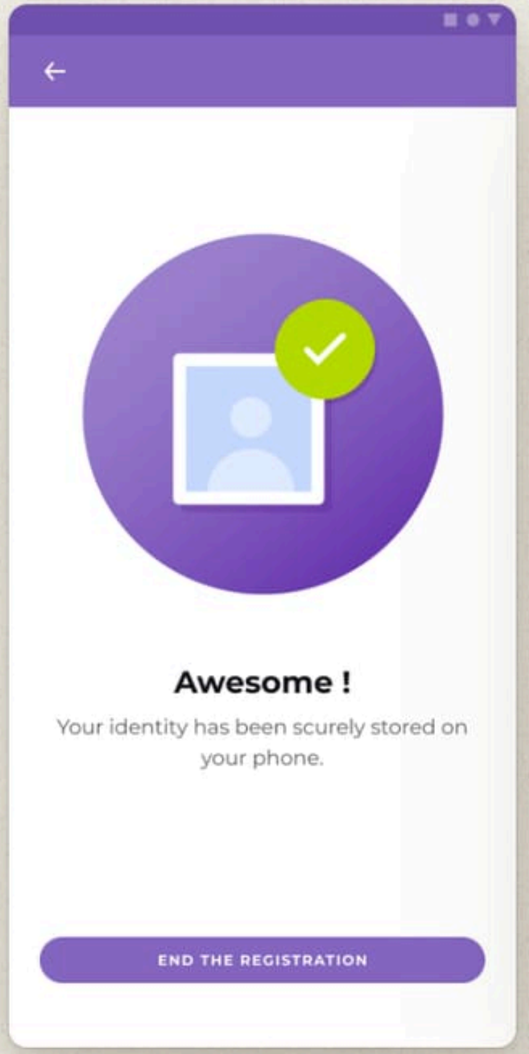


Step 9 · NFC authentication

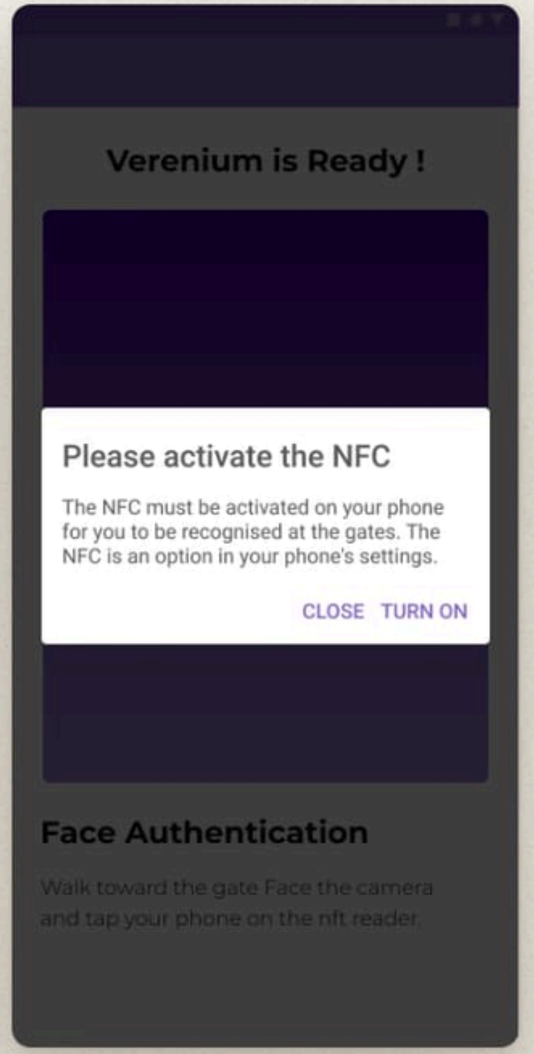
Read what the testers did

The count is highest on the very last step, the one the whole product exists for: 5 testers out of 6 could not finish it without help.

People understood the promise.	5 of 6 understood what they would have to do. 5 of 5 trusted the technology. Average score: 4.5 out of 6.
Nobody reads what you write.	6 of 6 accepted the terms without opening them.
The capture gives no feedback.	2 of 6 restarted the capture: nothing told them the photo had worked.
NFC means paying.	3 of 6 did not know what NFC was on a phone.
And the last step.	1 of 6 got through unaided.



Tested · Identity valid



Tested · NFC prompt

Where the 9 steps broke

On screen

The number of testers each step put in trouble on the full journey. 1 square is 1 tester. Counts, not a trend.

What it showed

The product was not rejected. It was just not usable yet.

5 of 6

on the last step, the one the whole product exists for

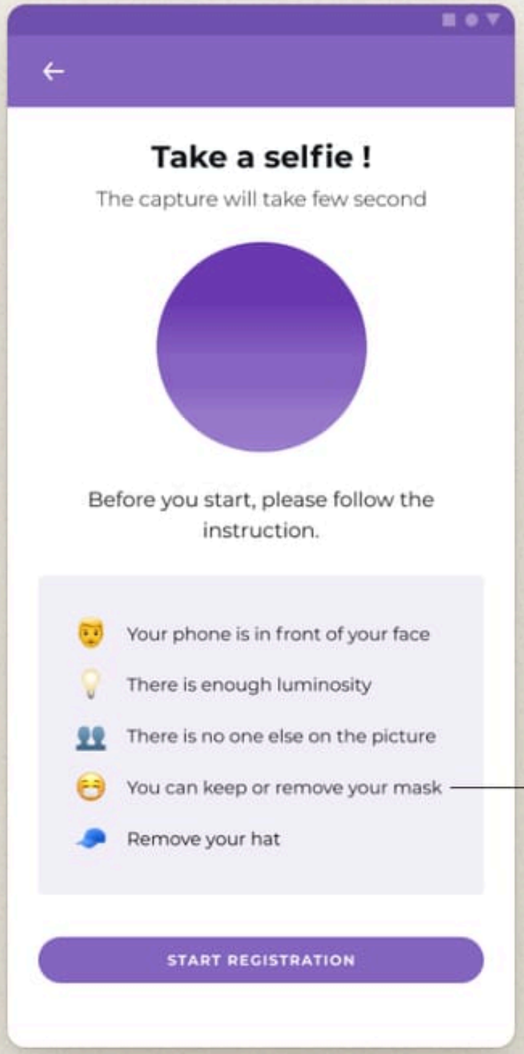
1	**Tutorial**	■□□□□□	**1** of 6	tried to capture their face during the tutorial
2	**Terms**	□□□□□□	**0** of 6	nobody failed here, and nobody opened the terms
3	**Permissions**	■□□□	**1** of 4	refused access to see if the app still worked
4	**Selfie capture**	■□□□□□	**1** of 6	the capture kept failing behind a mask, and they gave up
5	**Confirm**	■■□□□□	**2** of 6	restarted the capture, nothing said it had worked
6	**Identity valid**	■□□□□□	**1** of 6	read the word “awesome” as a bad sign
7	**NFC activation**	■■■□□□	**3** of 6	did not know what NFC on a phone is
8	**Face detection**	■■□□□□	**2** of 6	the terminal fired before they were in position
9	**NFC authentication**	■■■■■□	**5** of 6	could not reach the reader without help

6 testers on the full journey · permissions counted on 4

Separate the interface failure from the technology one

1 tester could not enrol at all. The capture failed, again and again. That tester was Black and wearing a mask. He gave up, because nothing on the screen told him why it was not working, and he ended up tapping the phone screen to try to take the photo himself.

Not a corner case	4 out of 6 kept their mask on for the capture.
A recommendation to the technology, not to the interface	Check the capture behind a mask across skin tones.
On a mock-up, every face works	Found because the tests ran on the real camera.



Selfie instructions, tested version · my finding in the report

Test report · selfie capture

The capture does not always work with a mask

Recommended

Check the capture behind a mask across skin tones, or ask every user to remove the mask on the capture screen so nobody is singled out by a failure they cannot explain.

On this screen

“You can keep or remove your mask”

One failure, two possible owners

The failure

1 tester could not enrol at all. The capture failed, again and again.

Owner 1

The interface

- Nothing on the screen told him why it was not working
- He ended up tapping the screen to take the photo himself

Owner 2

The technology

- The capture kept failing behind a mask
- 4 of 6 testers kept their mask on, so this was not somebody doing it wrong

What 1 case cannot settle

The mask, the skin tone, or both.

Where it went

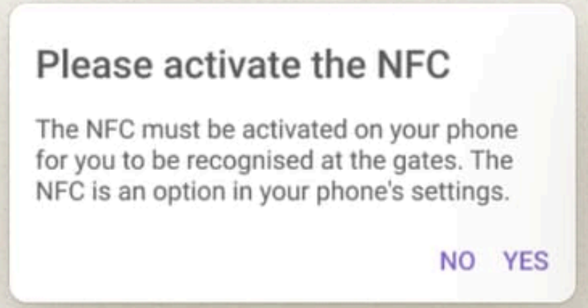
A recommendation to the technology, not to the interface.

Run an A/B test, and accept what it said

The NFC step was the worst one, so we tested 2 wordings for the prompt, 5 testers each.

4/5 against 2/5 on the first attempt	4 of 5 with “Yes / No”, 2 of 5 with “Close / Settings”.
Not a controlled comparison	2 separate panels: the wording moved the first attempt and not the outcome.
I kept it in the report as it was	The half that looked good would have sold a fix that does not fix anything.
What actually blocked people	No label solves that. An explanation and an error screen with a way out do.

Yes / No professionals

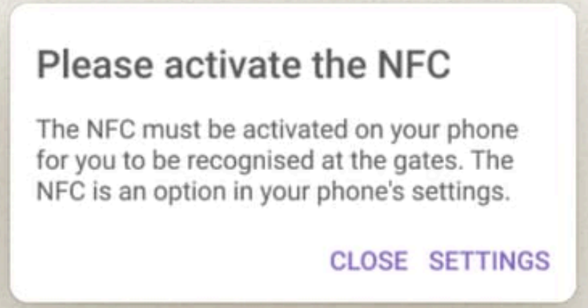




4 of 5 on the first try

4 of 5 through in the end

Close / Settings with 3 students





2 of 5 on the first try

5 of 5 through in the end

■ first try ▨ next try □ gave up

My own line in the file that day

No notable difference in behaviour between the 2 wordings.

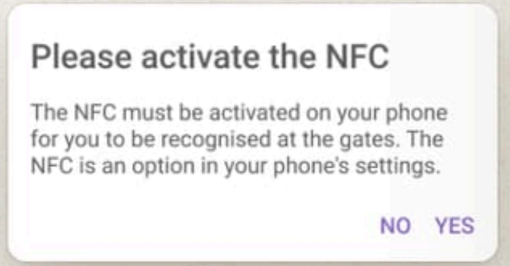
What actually blocked people

On screen

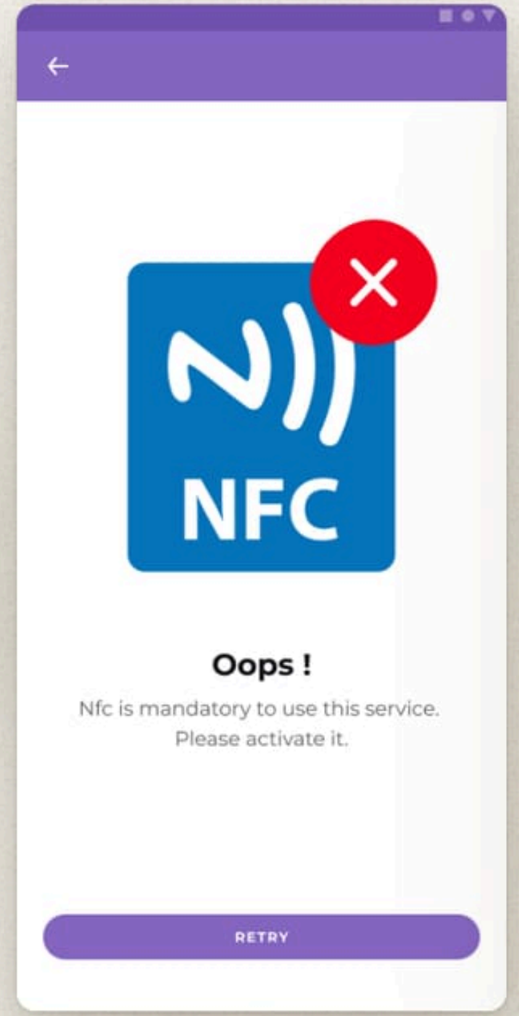
The “Yes / No” version: tap No and you land on my error screen, with a way back. Tap Yes and you land on the Android settings.

What it showed

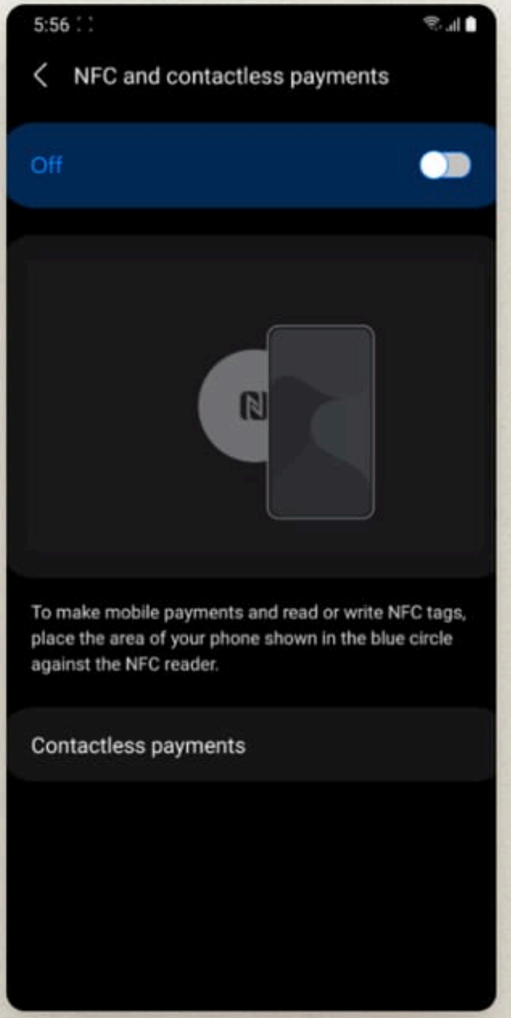
They did not know what NFC was, or why a product that had just recognised their face still needed it. No label solves that. An explanation and an error screen with a way out do.



1 · The prompt



2 · Tap No: my error screen



3 · Android · not my work

Hand over 26 recommendations, documented

The report came back with 26 changes, each tied to the observation that produced it. It would be easy to read the whole study as the NFC step not working. It did work: the chip authenticated every time it was reached. What failed was guidance, in three places that have nothing to do with each other.

The word.
3 of 6 read NFC as contactless payment.

The system screens.
3 of 6 stopped on the Android settings.

The reader itself.
The fix moves the hardware.

7 of the 26 are about NFC
3 on activation, 4 on authentication.

They are not ranked.
They follow the journey, screen by screen.

A documented handover
A handover says what to do with it when we are gone.

26 changes, in the order of the journey

On screen

The 26 changes, screen by screen, and 1 of them in my words for each screen.

What it showed

They are not ranked. The terms, where nobody failed, produced the most changes.

7 of 26

are about NFC, and one of them relocates a piece of hardware

Tutorial	3	■■■	Add a first step showing what enrolling a face actually involves
Terms	5	■■■■■	Make people tick a box before moving on
File and camera permissions	3	■■■	Remove the “don’t ask again” option
Location permission	1	■	Put the screen into production to test it
Selfie capture	2	■■	Add a countdown
Confirm	3	■■■	Add a green tick
NFC activation	3	■■■	Explain what NFC is
Face detection	1	■	Add a 3 second delay with a visible countdown
NFC authentication	4	■■■■	Move the physical reader under the terminal
Outside the flow	1	■	Offer another way in when the phone is flat

1 square is 1 change

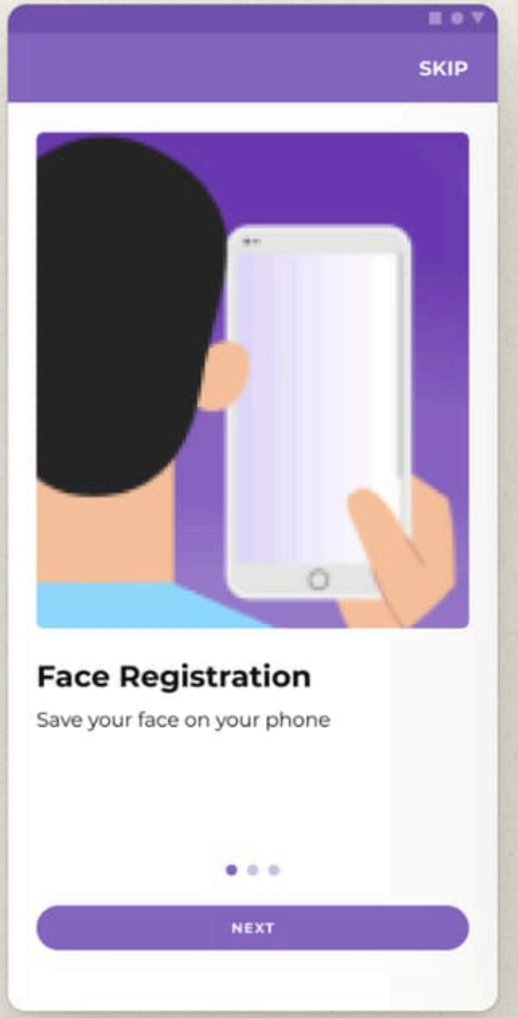
Add a first step showing what enrolling a face actually involves

On screen

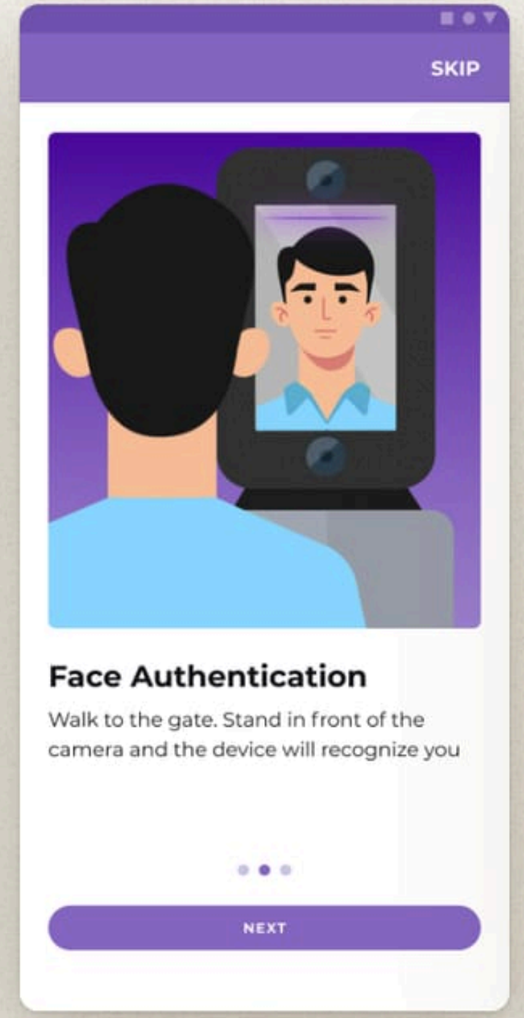
My recommended tutorial, after a first screen to choose the language: face registration first, then the 2 tested steps.

What changed

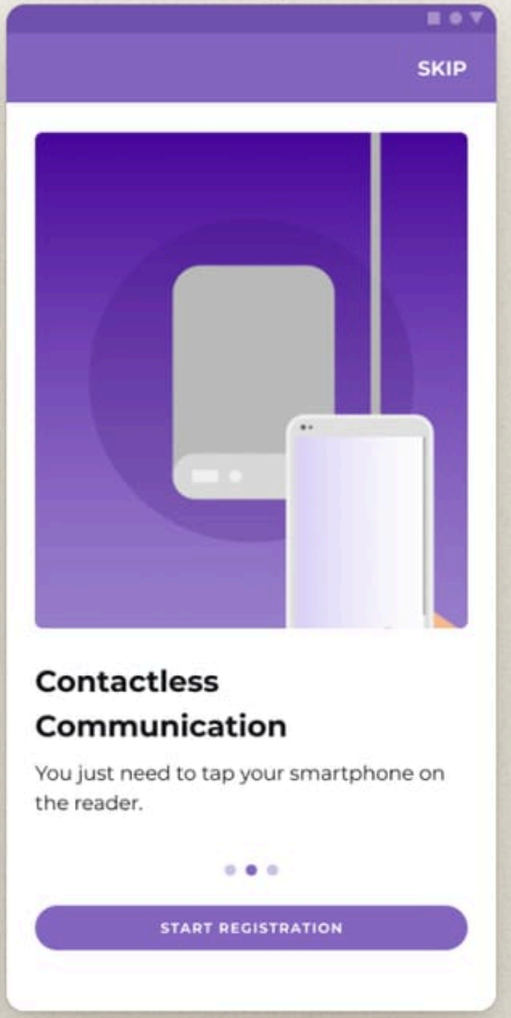
A first step showing what enrolling a face actually involves, and a Next button that makes it obvious the tutorial has several parts.



Recommended · step 1 of 3



Recommended · step 2 of 3



Recommended · step 3 of 3

The hardware moves to meet the gesture

On screen

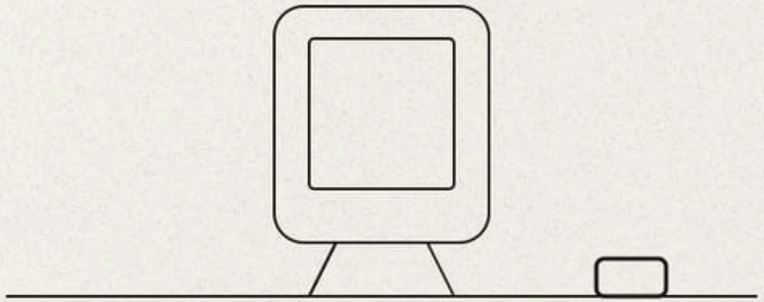
The NFC reader is not part of the terminal: it is an off-the-shelf HID OmniKey smart-card reader on the table, carrying no NFC marking at all.

Tester, translated

“There is no symbol saying this is for NFC, whereas a bank card carries the same symbol as the payment terminal.”

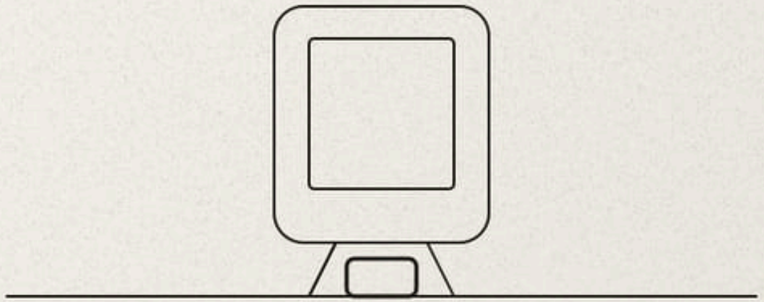
Tested

On the desk, to the right of the terminal



Recommended

Under the terminal, in the centre





NFC reader
HID OmniKey

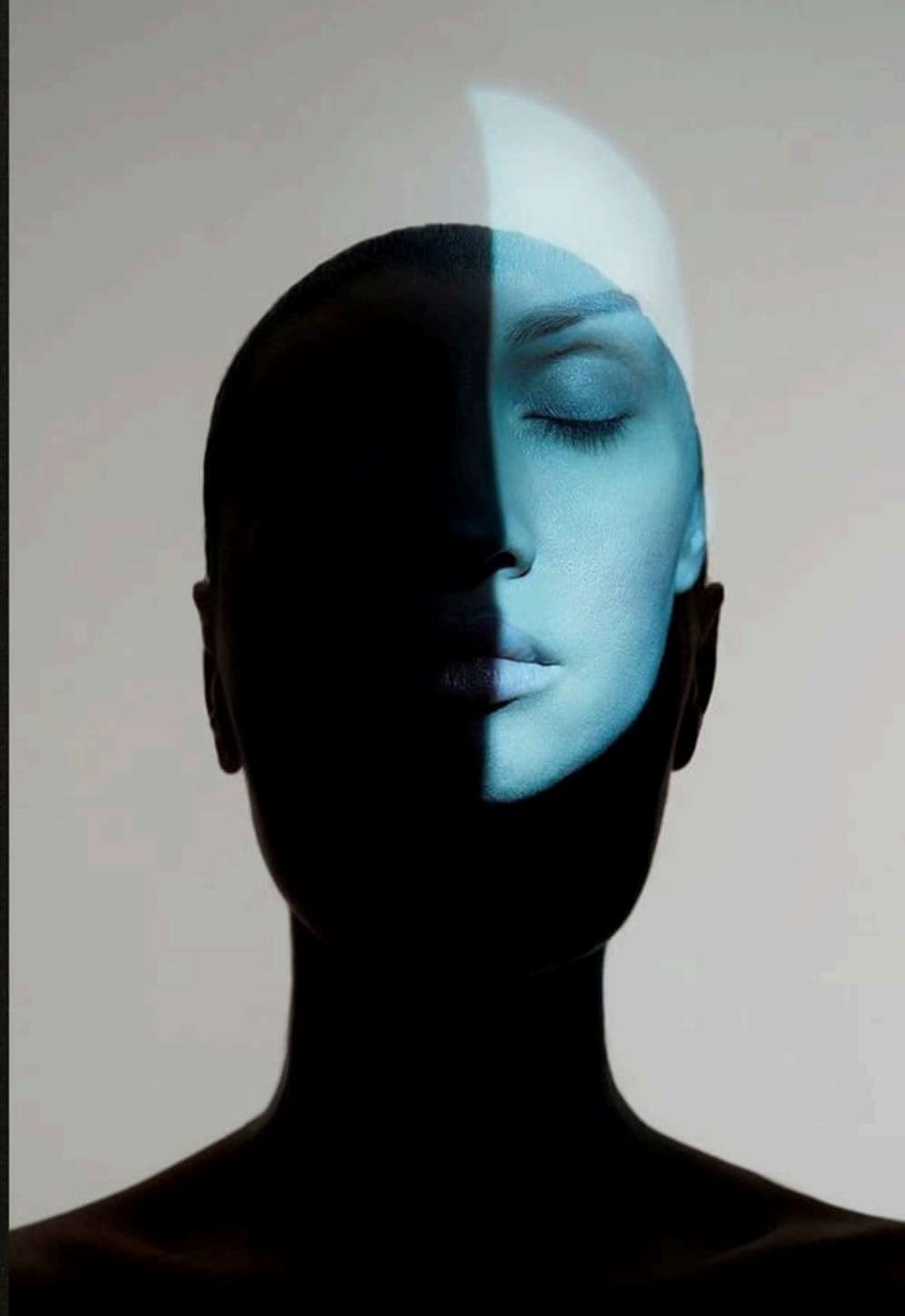
Photo of the test set-up · the reader is cut by the frame of my photo

Key result · Verinium Access Terminal, IDEMIA

26 recommendations handed over

25 across 9 screens, 1 outside the flow.
1 of 6 testers got through the final step on their own.

- 16 testers across 2 guerrilla waves, on the real technology.
- An A/B test reported with its negative result intact.
- For 5 of the 6 testers, the technology worked. The sequence did not. What failed was a phone, a reader, and a step that asked people to know a piece of vocabulary nobody outside the building uses.
- Every recommendation is attached to a number of testers and a moment in the session, so the engineers could argue with the finding rather than with me.
- A documented handover to the team that took the product on.





Phygital Field Research

IDEMIA builds the hardware and the software that states, police forces and airports use to prove who somebody is. In these 3 studies, the person using the product is outside, in a queue or on a roadside, and nobody can ask for help. The step that breaks is physical, not graphic.

My role	UX Designer inside R&D, next to the engineers and the hardware lab
Crew	UX team inside R&D, R&D engineers and the hardware lab, product managers
Problem	Find out where each product loses the person using it, before it is deployed to people who get one attempt and no help.

Mission 01 — Rebuild the thing at full scale — 3 sub-missions

Mission 02 — Land border: from around 20 to 10-13 minutes per vehicle — 9 sub-missions

Mission 03 — Guide a traveller at an airport gate, without instructions — 9 sub-missions

Mission 04 — Ride with the gendarmes and their mobile speed camera — 9 sub-missions

Rebuild the thing at full scale

Rather than show a prototype and ask people what they thought, we rebuilt the product at full scale and let people fail on it.

Land border	**A marked-out car park** The whole crossing, recruited panels, a stopwatch.
Airport gate	**A working turnstile in the lab** Hardware and screen tested together.
Speed camera	**Nothing to rebuild** The product lives in the boot of a car, so I got in the car.

My IDEMIA method	Land border	Airport gate	Speed camera
Observe Experience map User journeys		■	■
Co-create Discussion guide Scenarios User flows	■		■
Prototype Testable prototypes	■	■	
Test Test summaries Measurements	■	■	
Report back PowerPoint report, SharePoint Research repository Excel analysis, data evidence			■

Land border: from around 20 to 10-13 minutes per vehicle

Under the new European entry and exit rules, every traveller now has to register their face and their fingerprints, at a post that was already saturated.

The question

Not how to design a screen.

It was whether the crossing could absorb the new step at all.

Co-create

Workshops · Storyboards · Hardware + software scenarios

I had a blank page: hardware and software as one object.

Prototype

Wireframes, Figma · Portable terminal · Full-scale mock-up

A portable terminal taken to the vehicle guides each passenger in turn, without anyone stepping out of the car.

Test

Guerrilla tests · Timed field tests · Recruited panels

The number held whatever the number of occupants.

Timed on a marked-out car park

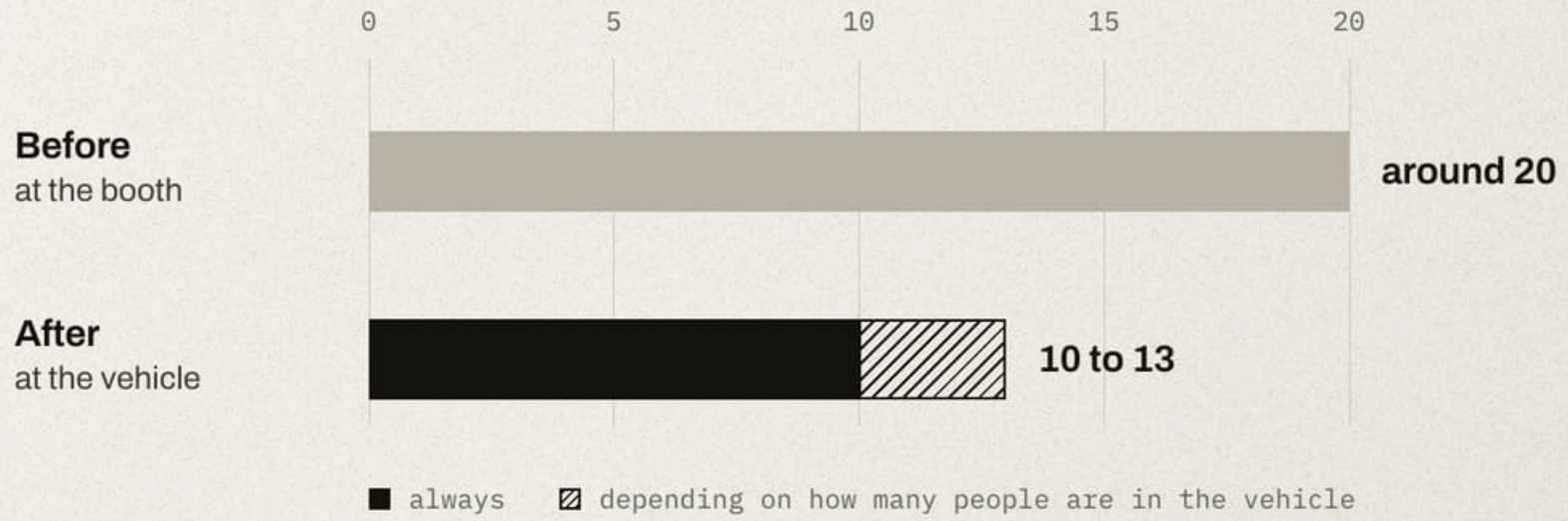
On screen

Minutes per vehicle, before at the booth and after at the vehicle, timed at full scale.

What changed

I started from the traveller's journey rather than from the technology, then measured it rather than argued it.

Minutes per vehicle



What every crossing now requires

Passport + Face + Fingerprints — at a post that was already saturated

already · new · new

Guide a traveller at an airport gate, without instructions

The traveller scans their passport, then a camera compares their face to the photo in the chip. Where somebody stands, where they look: a screen alone cannot carry that.

Observe	**Airport shadowing · Interviews** **Ergonomic audit of the gate** I audited the existing gate first.
Prototype	**Wireframes, Figma · Guidance screens** **Full-scale gate** A gate rebuilt at full scale in the engineers' lab.
Test	**User tests in workshop conditions** **Timed field tests · Recruited panels** So that a first-time traveller gets through alone.



Illustration · gates at Roissy



Illustration · IDEMIA face capture

Ride with the gendarmes and their mobile speed camera

A camera that lives in the boot of a car, with a case, cables to connect, and a tablet interface. Shadowing is the only way to see what the product actually costs.

Observe	**Field shadowing · Interviews** **Ergonomic audit** I rode along with gendarmes during real checks.
Co-create	**Field user journey · Workshops** **Storyboards** Usability problems ranked by what they cost in the moment.
Report back	**Prioritised recommendations** **Research repository · Atomic research** Their pain points, turned into improvements.







Illustration · the dashboard (top), the boot and the tablet (below)

A user who is on duty

In the field	The install before the shift, the cables, the moment the tablet is held in one hand while something else happens.
Came back with	A field user journey, usability problems ranked by what they cost in the moment, recommendations for the existing solution, innovation leads for the next one.
Lesson	**A gendarme will not abandon a bad tool.** They will work around it, every shift, and the workaround becomes invisible to everyone except the person paying for it.

What the 3 have in common

01 **A finding on a screen is not a finding.**

The step that breaks is a hand in the wrong place, a machine that looks idle while it is working, a cable to connect before the shift starts.

02 **Measure the journey, not the interface.**

Not a satisfaction score: the only unit the people who commissioned the study could act on.

03 **Go where the product is used, or rebuild that place.**

A car park marked out as a border post, a gate rebuilt in the lab, a gendarmerie car.

Kept since

Rebuild the thing at full scale, then watch somebody fail on it.

The failure is almost never on the screen.

Key result · Phygital Field Research, IDEMIA

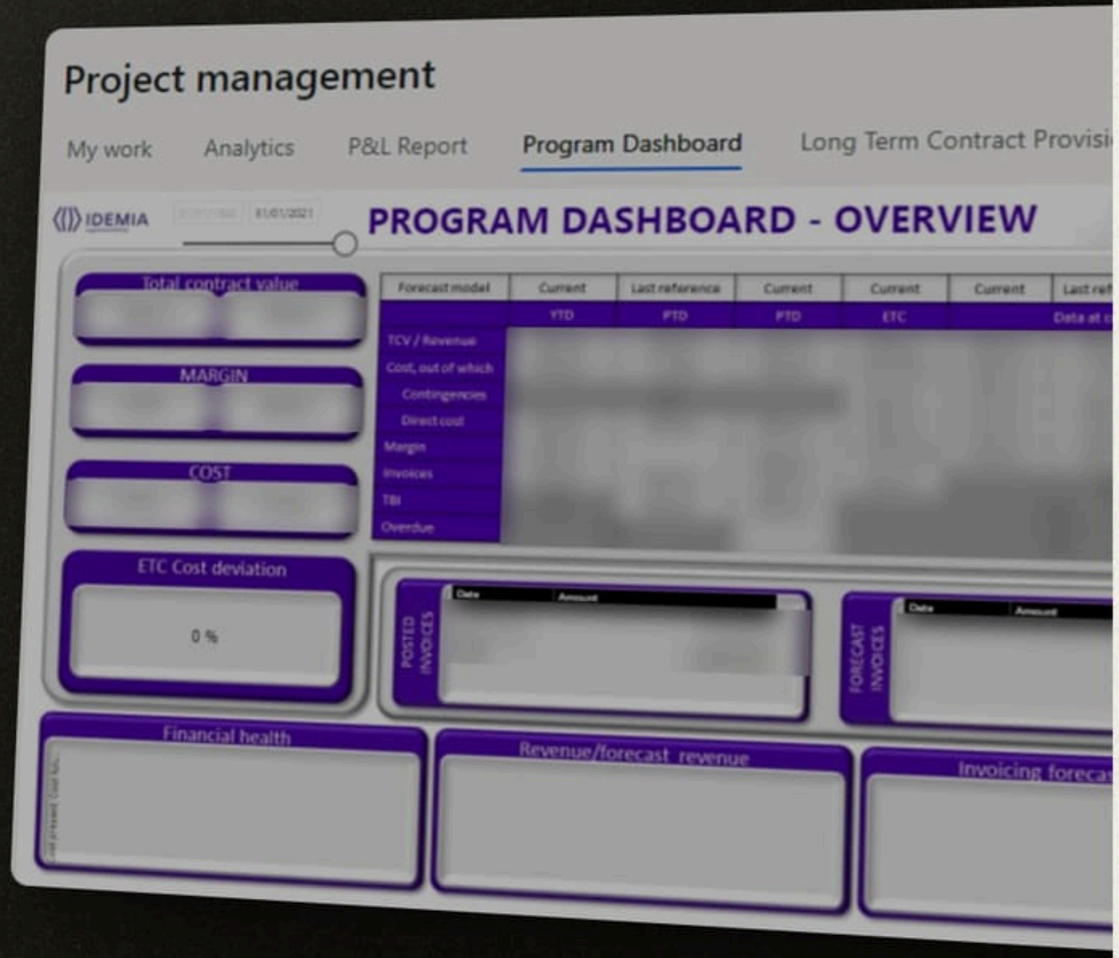
10 to 13 minutes per vehicle

Instead of around 20, whatever the number of occupants.

- A portable terminal taken to the vehicle, which guides each passenger in turn without anyone stepping out of the car, with enrolment completed alone on the first attempt.
- An airport gate audited, then rebuilt at full scale in the lab, so the guidance could be tested with the hardware and the screen together.
- A field journey of the mobile speed camera from real roadside checks, usability problems ranked by what they cost in the moment, innovation leads.



IDEMIA



Program Dashboard · existing tool, not my work

Case study · IDEMIA, UX research, 2021

Program Dashboard Research

A program, at IDEMIA, is the execution of a contract: an order, deliverables, a budget, and a review where a program manager reports to top management. The tool that carried all of it, across 4 continents, was a spreadsheet. The question was where the people filling it were losing their time, and which of the things everyone assumed about them were actually true.

My role	UX Designer: UX research, from protocol to recommendations
Crew	2 UX designers: another UX designer from the team and me
Problem	Find where to make the program review and the reporting faster: save program managers' time, cut the time it takes to reach a piece of information, and automate what can be automated.

Mission 01 — Write down what everyone already believed — 2 sub-missions

Mission 02 — Measure with 67, then listen to 5 — 3 sub-missions

Mission 03 — Read the numbers against the beliefs — 2 sub-missions

Mission 04 — Ask importance and uselessness as two questions — 4 sub-missions

Mission 05 — Follow the figures back to where they come from — 3 sub-missions

Mission 06 — Hand over 18 recommendations and the KPIs to judge them — 4 sub-missions

Write down what everyone already believed

The business unit already had a view of the problem, and so did I. So before collecting anything we wrote the assumptions down, 15 of them, in plain sentences. At the end of the research each one came back into 1 of 3 piles. That sorting is the deliverable I would keep if I could only keep one.

15 assumptions · where each one ended up



8
validated

2
invalidated

5
needs more insight

15 assumptions, in plain sentences

Written with the business unit before anything was collected.

Sorted into 3 piles

2 of those 5 marked as needing user tests before anybody could claim them.

15 assumptions, sorted into 3 piles

What it showed

The 2 the research broke are the 2 everybody was surest of, and both were about error.

Validated · 8 of 15

- ■ Filling the finance file automatically would save time
- ■ Scrolling in the spreadsheet slows the review down
- ■ The information in the file could be pre-filled
- ■ Managers want to know when a file is updated
- ■ Reaching a piece of information takes too long
- ■ Leaders do not use the deck the file generates
- ■ The document is updated at the last moment
- ■ Nobody wants a second tool for finance and invoicing

Invalidated · 2 of 15

- ▨ **The manager makes no mistake filling the Program tab**
- ▨ **Errors are left unfixed for lack of time**

Needs more insight · 5 of 15

- □ Splitting finance from the rest would read better
  needs user tests
- □ Top management wastes time hunting for information
- □ Reporting to top management could be automated
- □ The KPIs are used by the program managers
- □ A website would make the dashboard easier to use
  needs user tests

Measure with 67, then listen to 5

The survey targeted at least 60 answers out of around 90 users, spread across countries. It came back with 67. The survey says how many. The interviews say why.

The survey says how many	About three quarters of everyone who uses the tool.
The interviews say why	4 out of 5 wanted information that already exists in other tools to arrive on its own.
8 ergonomic criteria	Every finding tagged with 1 of them, and with a criticality level.

67

survey answers

5

interviews, run remotely

Africa, Europe, 2 in Asia-Pacific, Latin America, in quality and program management roles.

8

ergonomic criteria

Guidance, explicit control, cognitive load, compatibility, adaptability, error management, consistency and tool.

Read the numbers against the beliefs

Access was never the problem: 77% of users had no trouble reaching the file at all. The problem was what the file had become. The tool was not hard to reach. It was hard to trust.

First, what the tool does well	The baseline: 53% do not find it heavy to use.
What the numbers said that nobody wanted to hear	41% do not correct the errors they find.

The 2 beliefs the research broke

What the survey measured

The manager makes no mistake filling the Program tab

32% say they make mistakes from time to time

Errors are left unfixed for lack of time

56% do see errors while filling the dashboard, and 64% of them fix those before the review

First, what the tool does well

A study that only lists failures is a study nobody acts on. That is the baseline the 18 recommendations have to improve on. Not a blank page.

The survey agrees · 67 answers

53%

do not find the dashboard heavy to use

52%

say it does not waste too much of their time and is not too complex

The interviews name the strengths precisely

01 **It holds a whole program in one place**

02 **It gets you a program's financial position fast**

03 **It survives a handover**

In their words

01 "If you want a synthetic view of a program it's the dashboard, you have all the data inside."

03 "You can turn over a project you are working on to another person, and you can explain everything with the program dashboard."

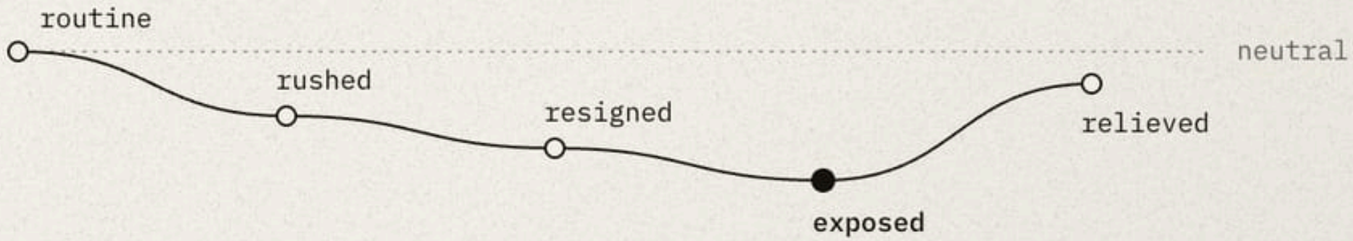
Not hard to reach. Hard to trust.

On screen

One cycle of the program review in 5 phases: what managers do, what the survey measured at each phase, and what I recommended.

What it showed

The low point is the review itself, the one moment the tool exists for.

	Open the file	Fill it in	Check the values	Run the review	After
How it feels, my reading					
What they do	opens the shared file	types the month's figures by hand	spots values that are wrong	scrolls the sheet in the meeting	closes it until next month
What the survey saw	**77%** reach it without trouble	**45%** fill it the day before the review	**41%** leave the errors they find	**75%** never use the deck it generates	**38%** do not update it right after
Where it hurts	The one phase that works	Filled in a hurry	36% say the file contains inconsistencies	A feature nobody takes to the room	Left until next month
What I recommended	nothing to fix	Make it simple to fill	A tool that guides as you fill it	Remove the generated deck	Alerts co-created with the managers

The curve is my reading of what they do, not a measured feeling. Every figure is measured.

The strongest evidence that a tool is failing is not a complaint about it, it is the system someone built to live without it.

In the interviews

2

of the 5 interviewees have stopped relying on the dashboard altogether

8 things managers built their own files to get

01 **A readable summary of the P&L**

02 **A month against month view, with a chart**

03 **Their budget across the year**

04 **A 12 month history instead of 2**

05 **The monthly cost trend**

06 **A standardized EAC (estimate at completion) file**

07 **A separate follow-up per sub-project**

08 **Costs planned against costs actually spent**

Ask importance and uselessness as two questions

The survey asked 2 separate questions about every section of the dashboard: how important is it, and is it useless to you. The effort was going into the parts nobody uses.

The sections they fill and few people defend	None called important by more than 20 of 67.
Internal Data, the one that takes the longest	Filled once, then forgotten for 12 months.
Streams, the worst of the five	62% do not use it regularly.
The sections nobody could explain	Banner, Customer Complaint, Opportunities, Actions.





The sections they fill and few people defend

On screen

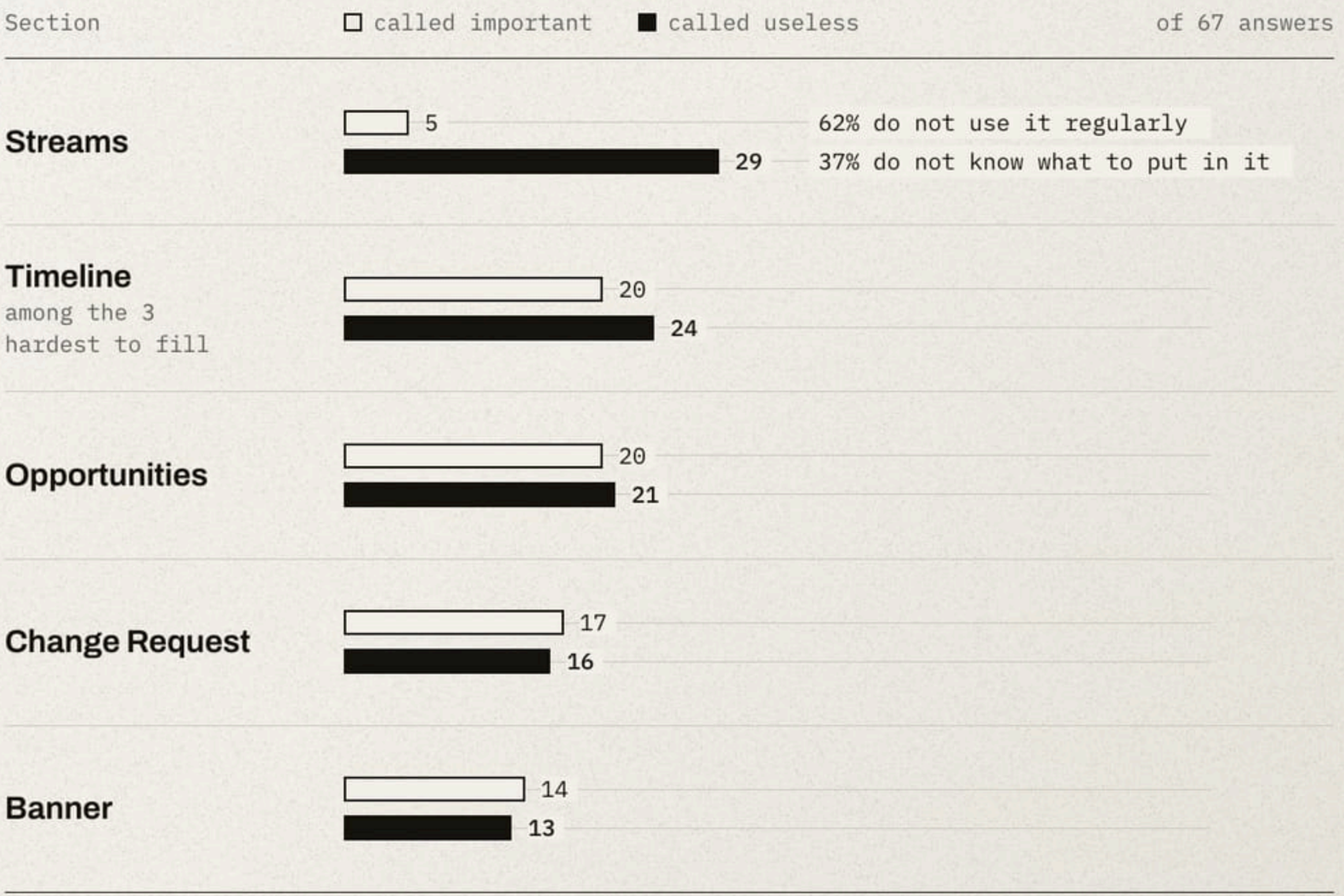
Five sections, asked about twice: how many of the 67 call each one important, and how many call it useless.

What it showed

Not one of the five is called important by more than 20 people out of 67. The recommendation was to remove or hide all five.

Streams · an interviewee

“If I could have, I would have deleted Streams.”



Asking only what matters gets you a list where everything matters.

Internal Data, the one that takes the longest

A program manager

“During the training, we understood that we only fill in the yellow columns, and if you ask a program manager today how to fill out the form, you’ll fail by 90%. You fill it once at the start of a contract, then you forget. In Latin America contracts run a year, so you do it again 12 months later having forgotten everything.”

Hardest to fill

Internal Data, 25 of 67

Wanted automated first

4th, 22 of 67

In Latin America

Done again 12 months later

The sections nobody could explain

Section	What the interviews found	In their words
Banner	4 interviewees out of 5 could not explain it.	“I don’t know what they use those for.”
Customer Complaint	2 interviewees out of 5 could not explain it.	“When I receive a mail from the quality team, I put 0 and I am at peace.”
Opportunities	The dashboard competes with a tool that already won.	“I won’t put an opportunity in if it isn’t in Salesforce.”
Actions	The field is too small for the job.	“You can have 30 action items on one project. I only put the top 5 in to send to management.”

4 sections, 4 different reasons, 1 shared cause

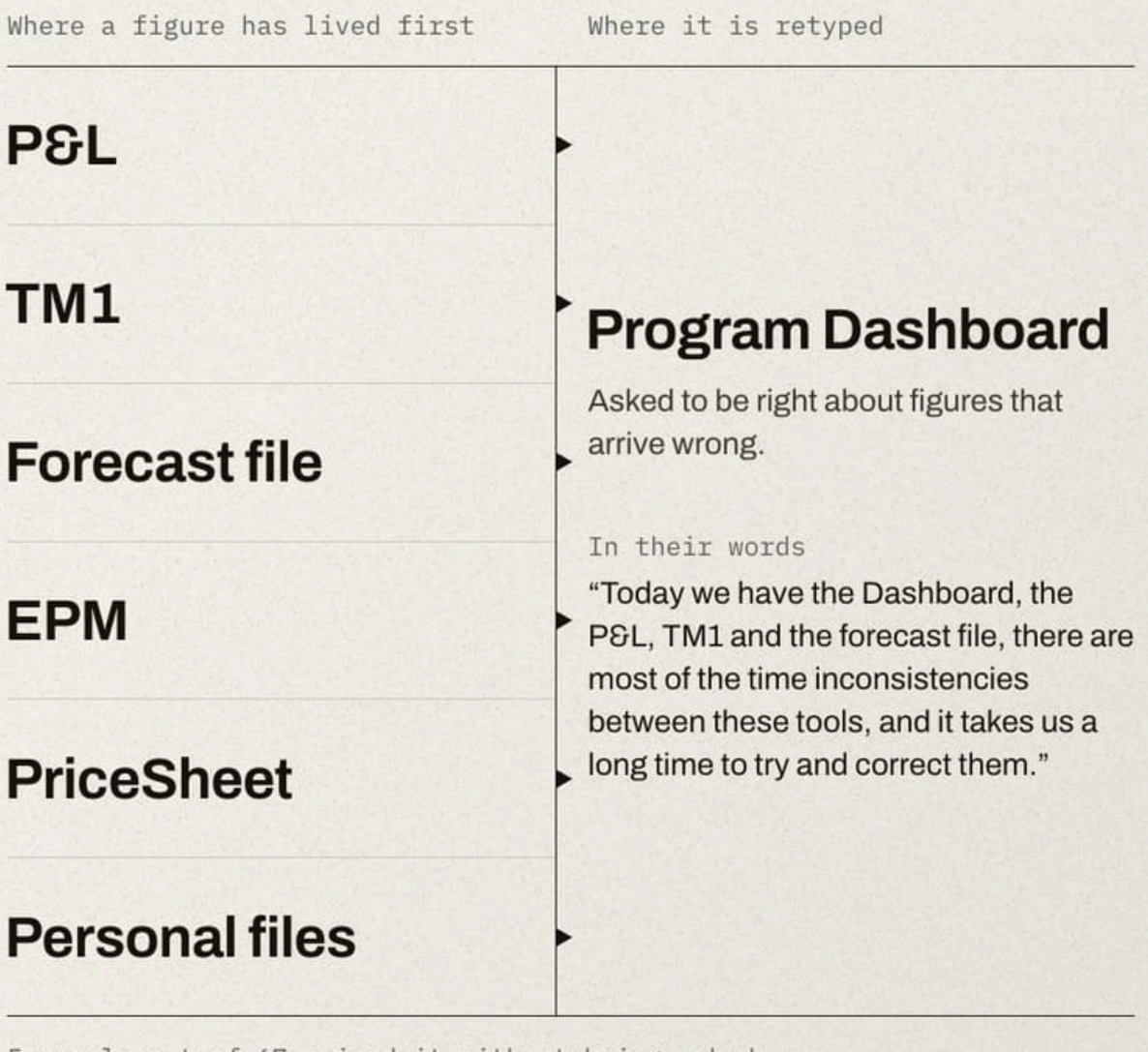
Nobody could name the reader.

A field whose audience is unknown gets filled defensively, or not at all, and either way the number that comes out of it means nothing.

Follow the figures back to where they come from

The sections are only half the story. The other half is what it takes to put a correct number into one of them, and the dashboard is rarely the first place that number has lived.

5 files, one truth, and no agreement between them	37 out of 67 ask to pull the data in rather than retype it.
The file fights back in small, concrete ways	Merged cells, 2 currency conversions for 1 figure, training that arrives 2 months late.
And it looks its age	30% say scrolling during the review stops it running properly.



The file fights back in small, concrete ways

What users told us

Merged cells

Do not merge cells, it makes 50% of the copy paste in excel fail.

Currencies

One manager budgets in New Zealand dollars, converts to Australian dollars, then converts again to euros before the figure lands in the dashboard.

Training that arrives after the fact

I had a training, but it was after I had already been using the dashboard for 2 months.

Written by a user in the survey

“Every attention shall be made so that it is easy to fill in the dashboard. Right now, it is focused on the display only, resulting in a poor data entry interface.”

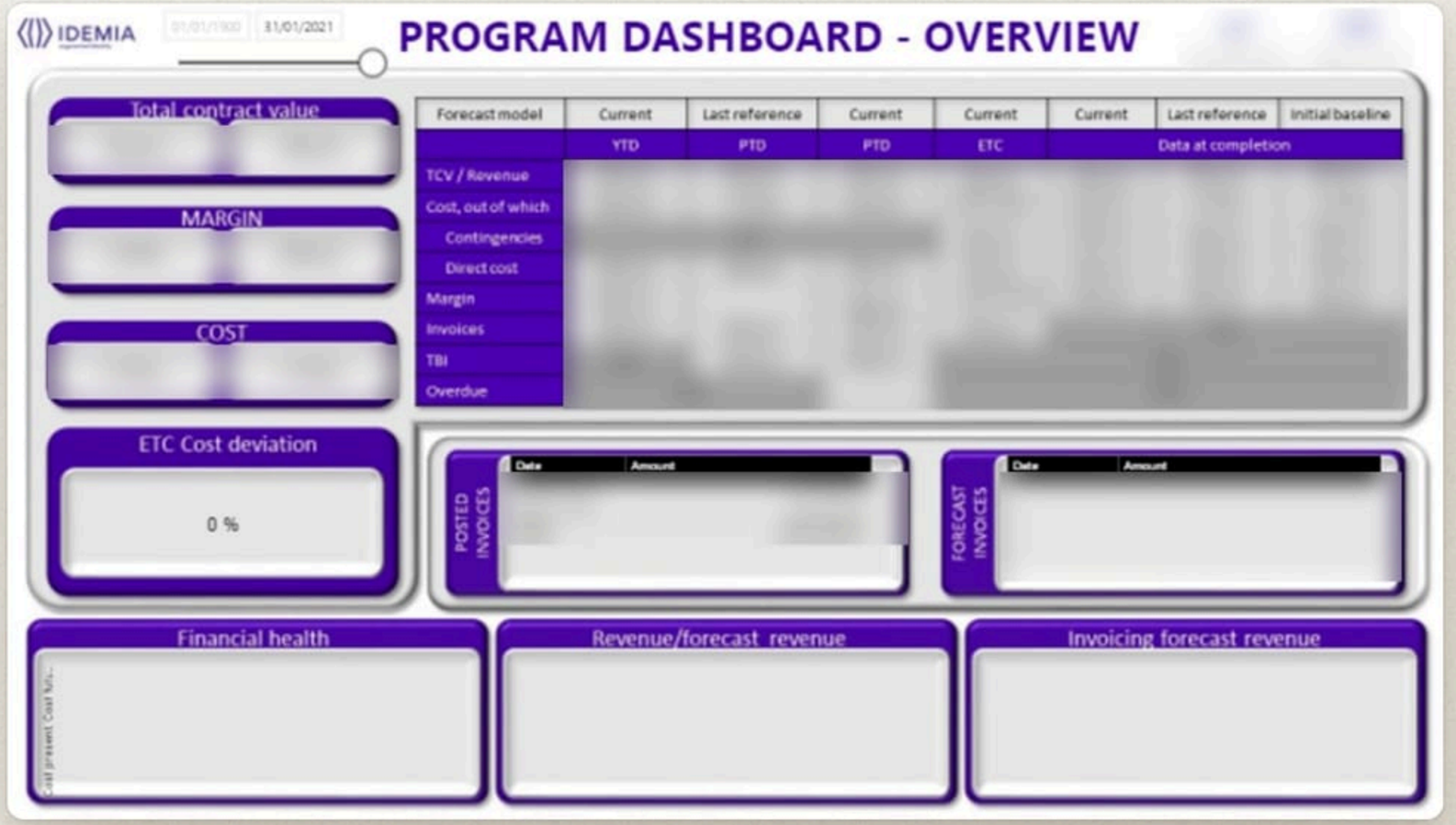
And it looks its age

What users said

10 people out of 67 asked for the thing to be more visual. “Modernisation of interface, it looks very 80's.” “I find the page very badly designed, it's a mess, you have to go up and down, right and left.”

What it meant

30% say that scrolling during the review stops it running properly. The complaint is not about beauty, it is about reading a wide sheet in front of an audience while someone waits for an answer.



Program Dashboard, overview screen · existing tool, not my work

Hand over 18 recommendations and the KPIs to judge them

The recommendations came back in 5 families, grouped under 2 ideas: put the tool at the service of the person, and put it at the service of the job. Each one carries the criticality the deck gave it.

At the service of the person	9 recommendations, in 2 families.
At the service of the job	9 recommendations, in 3 families.
What it would have taken to settle the rest	9 UX KPIs, measured before and after.
The website nobody asked for	The research rated it low.

The commission, planned in 4 steps

■ March
Brief and launch — this study

■ End of May
Research results — this study

☐ June
UX workshops

☐ Summer
Prototype and user tests

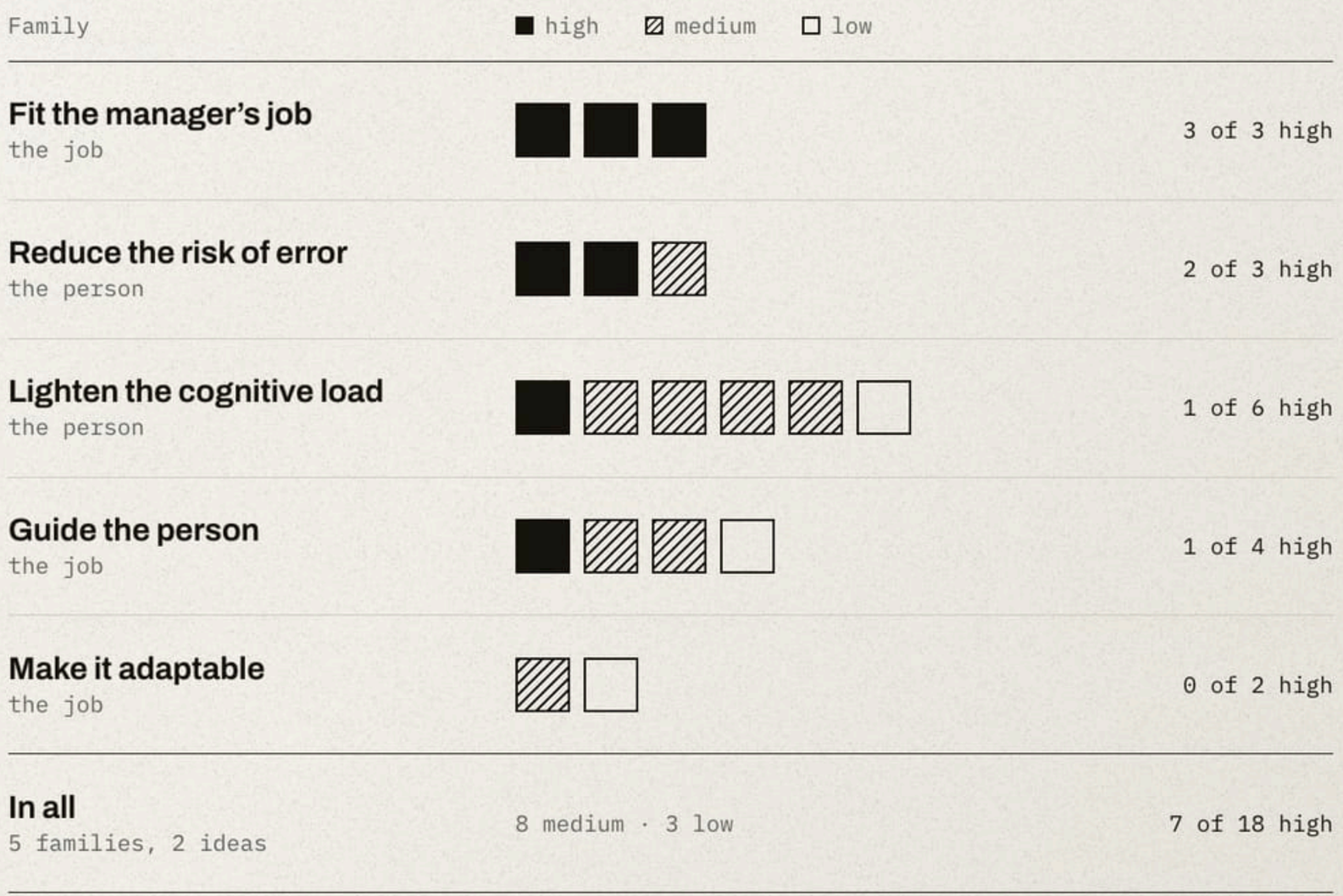
7 of 18 recommendations rated high

On screen

The 18 recommendations in their 5 families, 1 square per recommendation, with the criticality the deck carries next to each one.

What it showed

The only family rated high from end to end is the one about the manager's own job, and the biggest family, cognitive load, is almost entirely medium. The recommendation to move off the spreadsheet is rated low.



At the service of the person

9 recommendations in 2 families, each with the criticality the deck carries next to it.

Family	Recommendation	Level
Lighten the cognitive load 6	Co-create the alert system with the program managers, rather than deciding for them which alerts they want.	■ High
	Make the dashboard as simple as possible to fill, because it is updated in a hurry.	▨ Medium
	Delete the work that can be automated away, to kill the copy and paste errors.	▨ Medium
	Centralize and link the files and tools that belong to the same program.	▨ Medium
	Merge the tools a program manager opens every day.	▨ Medium
	Remove or hide the sections the survey had just shown to be dead weight: Timeline, Opportunities, Streams, Banner, Change Request, the automatically generated deck.	□ Low
Reduce the risk of error 3	Let a program be tracked in its own local currency, so the finances stop drifting.	■ High
	Choose an underlying tool less prone to error than a spreadsheet, and guide people as they fill the fields.	■ High
	Automate everything that can be automated.	▨ Medium

At the service of the job

9 recommendations in 3 families. The only family rated high from end to end is the one about the manager's own job.

Family	Recommendation	Level
Fit the manager's job 3	Allow foreign currencies, with a conversion that does not touch the main figures.	■ High
	Add a monthly view of finances to make the follow-up possible.	■ High
	Open a way for similar programs around the world to stop solving the same costly problem separately, to be confirmed by a later survey.	■ High
Guide the person 4	Give hints on how to fill each field.	■ High
	A simplified dashboard for newcomers, with training or mentoring to go with it.	▨ Medium
	Reorder the sections to cut the scrolling in both directions, the order itself to be settled in a workshop.	▨ Medium
	Bring the most consulted sections to the front: highlights and lowlights, financials, invoicing, contractual milestones.	□ Low
Make it adaptable 2	A separate view when a sub-project appears inside a program, next to the global one.	▨ Medium
	Move off the spreadsheet to something more visual, modular and personal.	□ Low

9 UX KPIs to judge it on afterwards

The 9 KPIs exist for the step after: measured before and after, so the fix can be judged on something other than the enthusiasm of the people who built it.

4 on efficiency

01 The time to fill the dashboard

02 The share of users keeping a personal file to track finances on the side

03 The share making mistakes

04 The share needing a colleague's help to fill it

5 on satisfaction

05 The share who find it takes a long time to get used to

06 The share who find the tool complex

07 The share who believe it holds values that are wrong

08 The share who think it should be organized differently

09 The share who feel they can navigate it efficiently during the review itself

The website nobody asked for

The commission

The brief asked the UX team to work on a website capable of replacing the spreadsheet.

What users said

4 users out of 67 proposed a web tool spontaneously in the survey. In the interviews 4 out of 5 said yes to the idea, but did not raise it themselves.

How the study rated it

Moving off the spreadsheet is rated low, the lowest level the study uses. Filling, converting currencies and guiding the user are rated high.

The finding I would put in front of the business unit first

The tool is not the first problem.

3 things for the business unit to do first

01 Confirm or adjust the criticality of each finding and each recommendation.

02 Decide what the dashboard would run on: a spreadsheet, a website or an existing business tool.

03 Ship the quick wins on the spreadsheet in the meantime.

Key result · Program Dashboard Research, IDEMIA

7 of 18 recommendations rated high

None of them is the website the business unit had commissioned.

- 67 answers out of around 90 users, and 5 interviews across 4 continents.
- 15 assumptions sorted into 8 validated, 2 invalidated and 5 needing more insight.
- 18 prioritized recommendations, with 9 UX KPIs to judge them on.
- The finding that reframed the project: the tool was not hard to reach, it was hard to trust.
- Here I produced the research and handed the recommendations to the build. At Pernod Ricard, a year later, I was the one building from somebody else's research.



Loxam



Create a B2B app

Loxam is a French construction equipment rental company. Its target businesses are building, heavy construction via general building contractors and road works. In addition to the online customer area available since 2017, Loxam wanted to offer a mobile app to allow the “artisan pro” target to manage their rentals online.

My role	Lead designer, with a second designer in support
Crew	2 product designers, 2 squad masters, 1 strategy and method lead, 1 business expert
Problem	Validate the mobile application concept and the interest of the functionalities. Identify and collect additional needs and user pain points. Recommend key paths and model them.

Mission 01	Understand	4 sub-missions
Mission 02	Prioritize key features	4 sub-missions
Mission 03	Prototyping	2 sub-missions
Mission 04	Test	5 sub-missions
Mission 05	Analysis	6 sub-missions
Mission 06	Presentation of results	5 sub-missions

Understand

MyLoxam App aimed to simplify the rental experience: operating features (booking), a dashboard of contracts and rental machines, and a conversational platform with the agency.

The client had a specification and wireframes.

Project planning	2 sprints: recruit 18 to 24 users and prototype, then test, analyse and recommend.
Ergonomic audit	An audit of the interfaces and an analysis of the existing user path.
Userflow	The different paths of the solution, mapped to their corresponding screens.
Problems detected	3 problems found in the wireframes, and a workshop proposed to the client.



Page liste matériels



Page produit



Page panier



Page choix du mode de livraison / retrait agence

Loxam's wireframes, the starting point · not my work

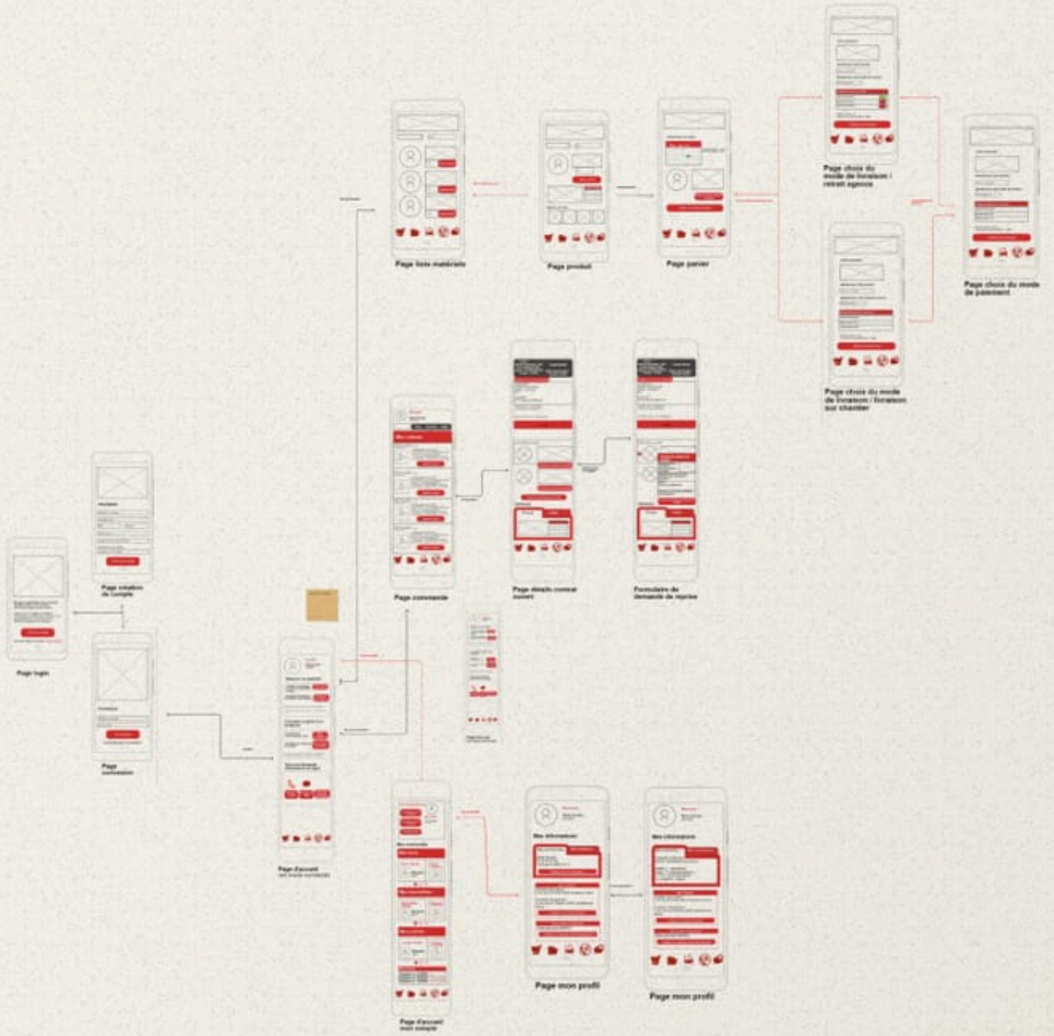
Problems detected

On screen

The client’s wireframes, linked path by path: login, account creation, equipment list, product, basket, delivery mode, payment, orders, contract, return request, profile.

What it showed

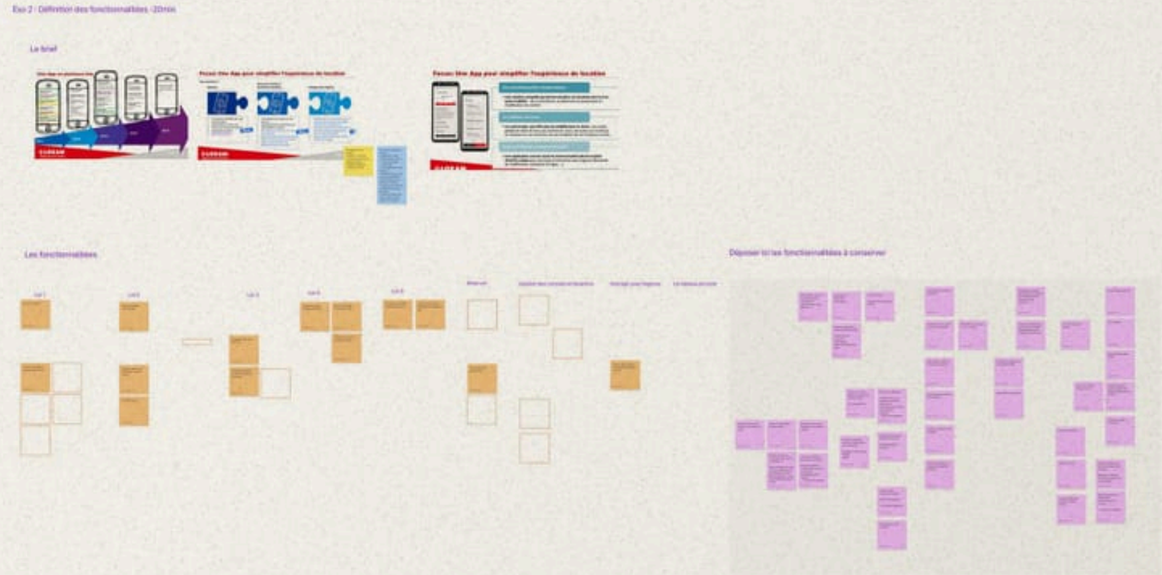
I mapped every path of the specification to its screens. Several paths did not appear in the wireframes, several features were confusing, the sequences were inconsistent. I proposed a workshop to the client.



Userflow of the MyLoxam App · the screens are the client’s wireframes

Prioritize key features

We prioritized together the features that should appear in v1 of the application during a workshop with the client. At the end of this workshop, the following objectives had been achieved.



Workshop board · “Exercise 2: defining the features”

The list of key features to keep or add

An understanding of the various technical specifications of the product

The definition of the different user journeys to be tested

A list of wireframes to create or modify on the client side

22	7	10	5
pages listed	ready	to rework	with no mock-up

The 5 with no mock-up: payment, fleet, help and contact, notifications, inspection report.

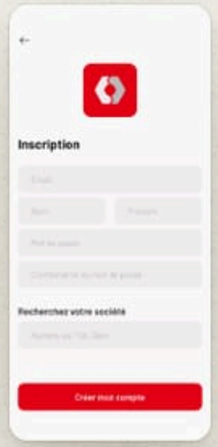
Prototyping

The client wanted me to translate the wireframes into interactive models in order to assess the relevance of the journeys with end users. Figma mock-ups were first due from the client; at kick-off the agency took them on, and I built them.

Once all the paths were validated with the project manager, I built a UI kit on Loxam's graphic charter.

UI kit	Built on Loxam's graphic charter: icons, buttons and states, cards, calendars, fields, tab bars.
Interactive mock-ups in Figma	V1 of the MyLoxam app, in 6 sections: registration, catalogue, fleet, orders, assistant, account.

Registration



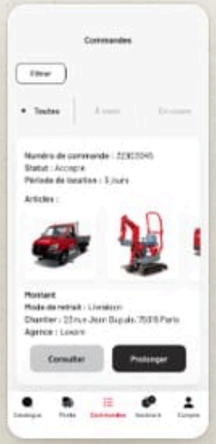
Catalogue



Fleet



Orders



Assistant



Account



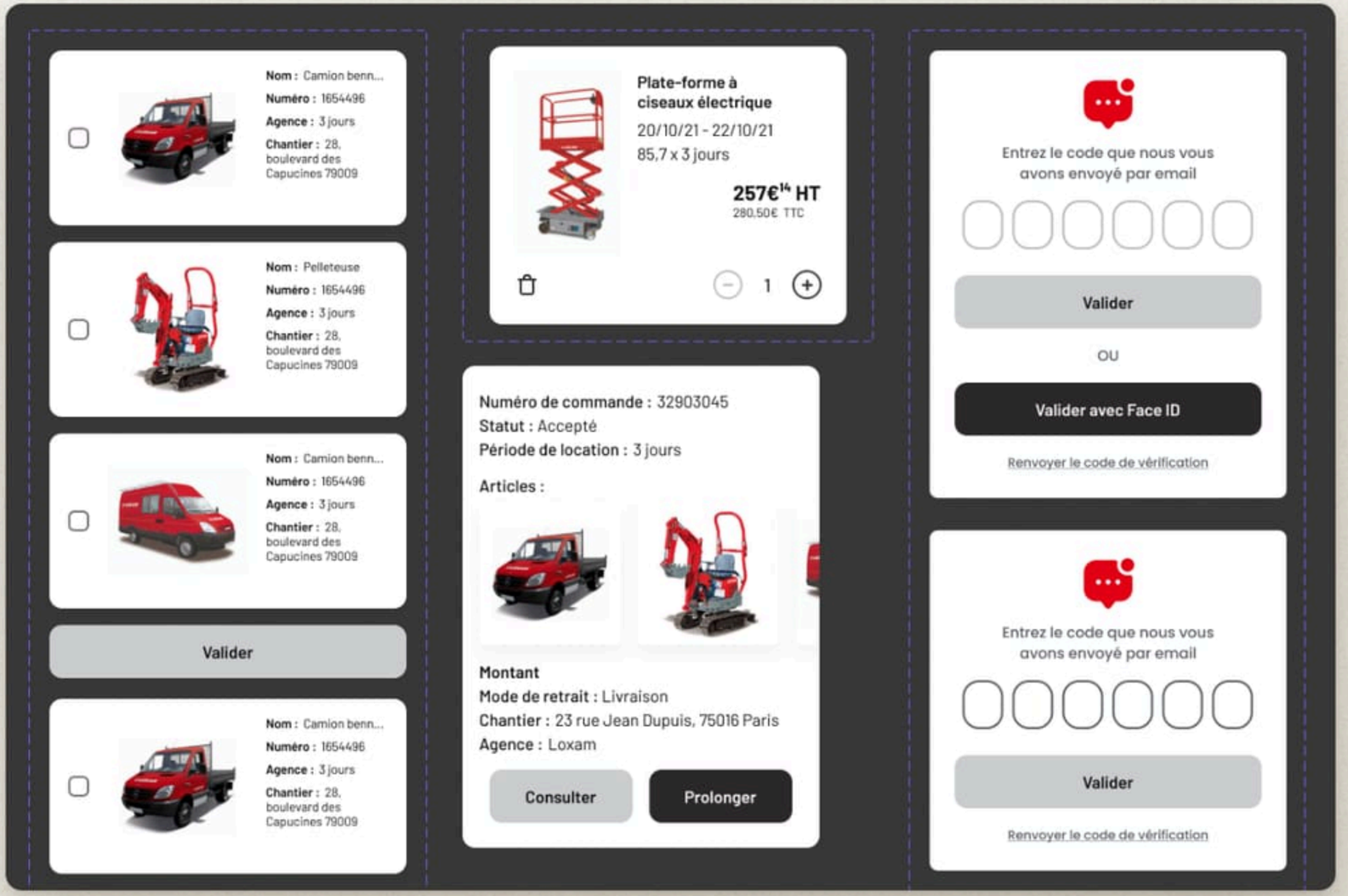
A UI kit on Loxam's graphic charter

On screen

Order and basket cards, the verification code and its states: 3 component columns of the kit, in my Figma file.

What it gave

The V1 and the 3 iterations were built with this kit.



UI kit · components, in Figma

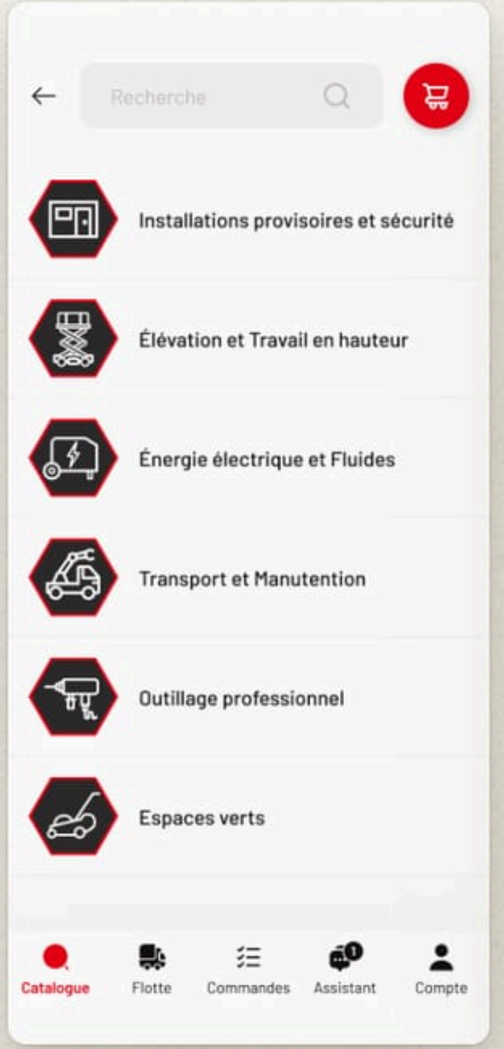
V1 of the MyLoxam app

On screen

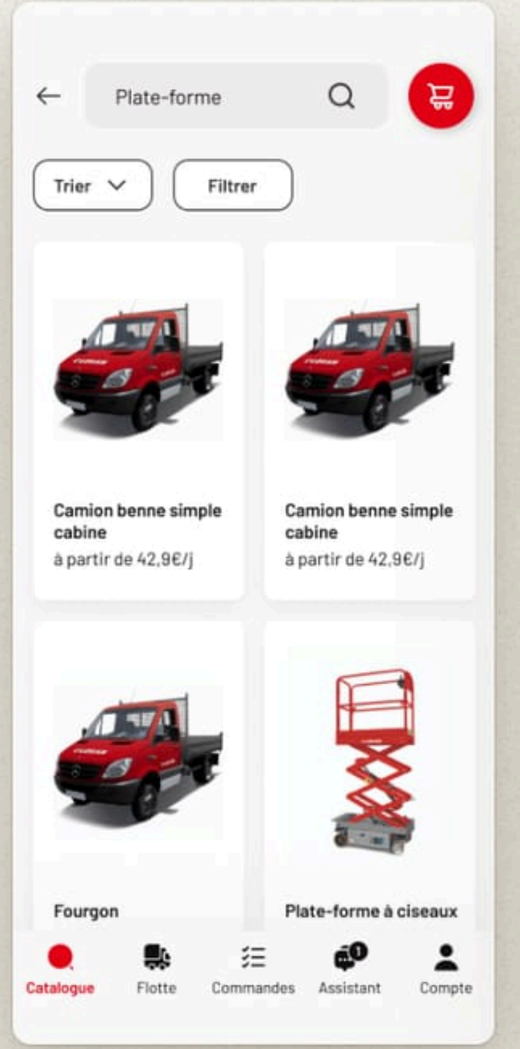
The first prototype, on the booking path: the catalogue by category, the product list, an item added to the basket.

What changed

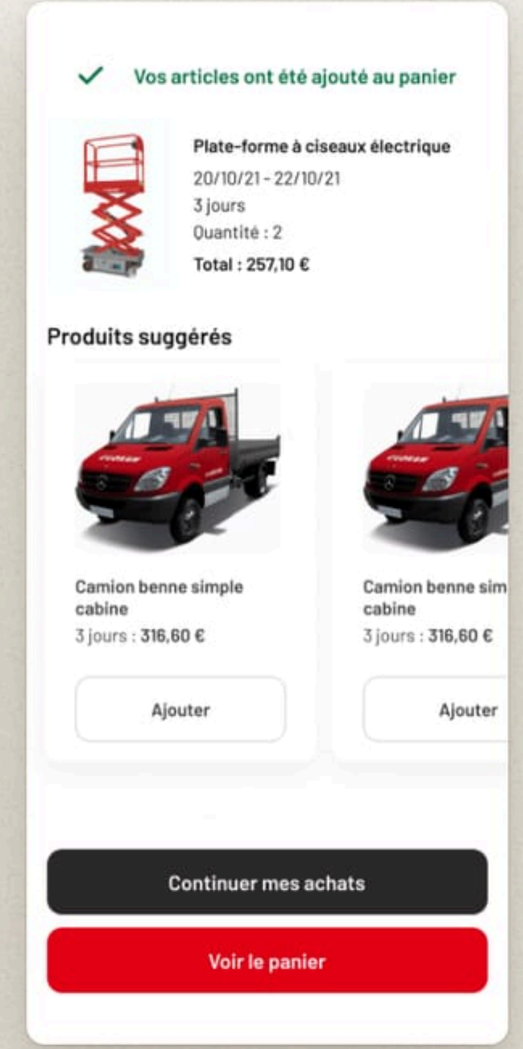
From the client's wireframes to interactive mock-ups on Loxam's graphic charter, so that the paths could be tested with end users.



1 · Catalogue



2 · Product list



3 · Added to the basket

Test

I led RITE tests over 1 week (29 November, 1 and 3 December 2021) with 18 to 24 users: field employees of companies with 10 to 100 employees.

User profiles	Ambassador user, occasional user, non-customer.
Testing guide	18 hypotheses, 25 test missions, success and failure criteria for each screen.
Test RITE	3 sessions in 1 week, 45 min per test, 3 design days to revise the prototype.
Face-to-face and remote test session	In the Loxam branches of Dreux, Arpajon and Goussainville, and remotely.
From a Word document to Miro	Each comment of the testers annotated below the screen it concerns.



Test session in the Loxam branch of Dreux · 29 November 2021

Test RITE: 3 sessions in 1 week

RITE: Rapid Iterative Testing and Evaluation. 45 min per test; on the 3 design days, I revised the prototype with a second designer.

Session 1

29 November · Dreux

The V1 prototype, 25 test missions. Then my recommendations on registration, order funnel, basket and labels.

Session 2

1 December · Arpajon

The revised prototype, 22 test missions.

Session 3

3 December · Goussainville

The third iteration, then the analysis.

Between sessions 1 and 2

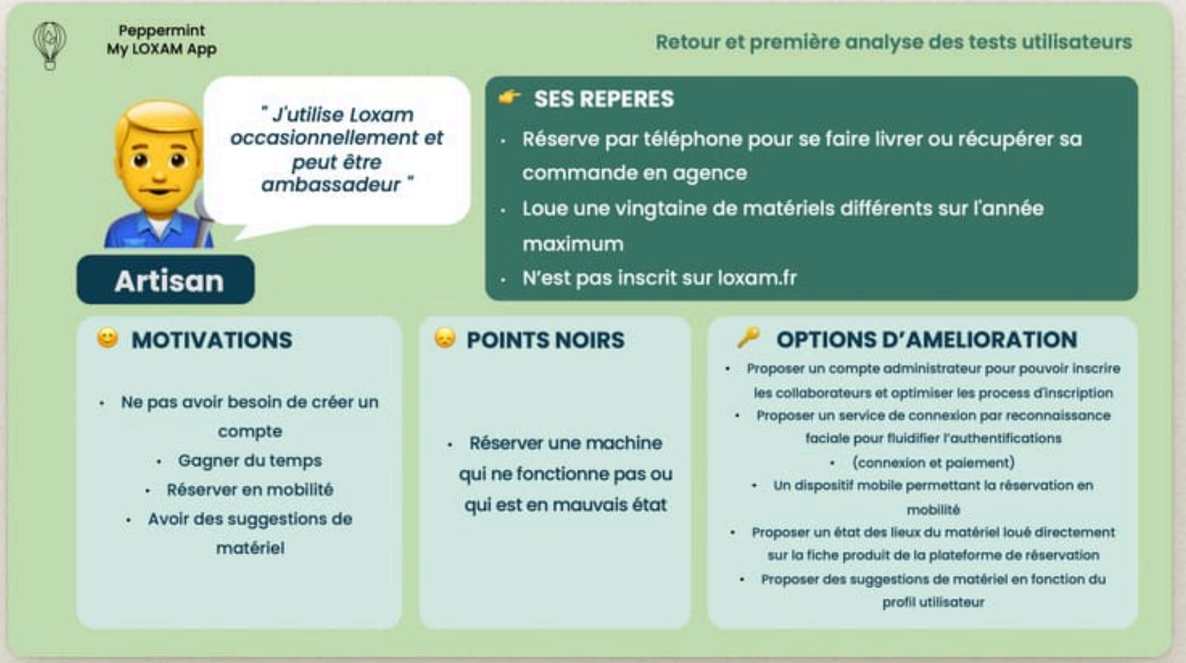
25 test missions, then 22

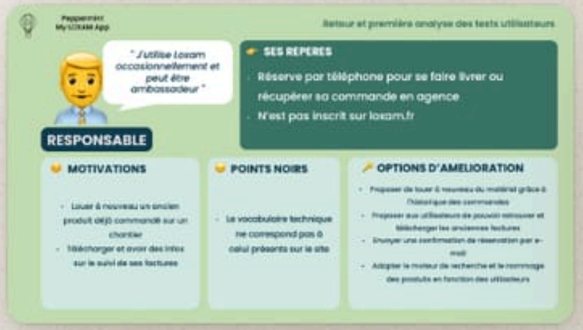
A login path and user rights added, a period filter, the scan in the search bar, card payment replaced by the company to bill, the fleet moved into the account.

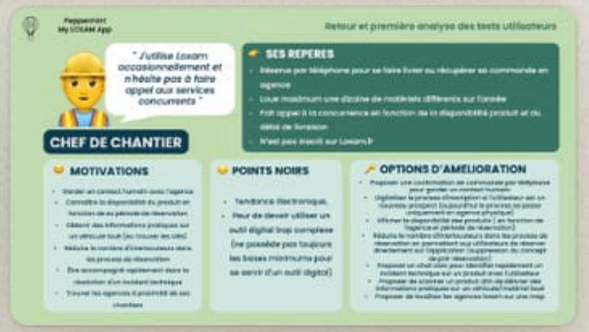
Analysis

To carry out the analysis of the tests on Miro, I applied a tagging plan to each target. I presented the results obtained in the form of personae and at the same time proceeded to modify the models in order to stage the new paths.

Tagging plan	Benchmarks, motivations, pain points, improvement options.
Users personae	From the data collected during the tests.
Test hypotheses	10 validated, 8 invalidated.
New features	Re-rent from the order history, an administrator account, facial recognition.
Prioritization of key paths	Registration and booking first.
Removal of paths and features that do not meet any need	9 features set aside, card payment included.







Personae · craftsman, manager, site manager

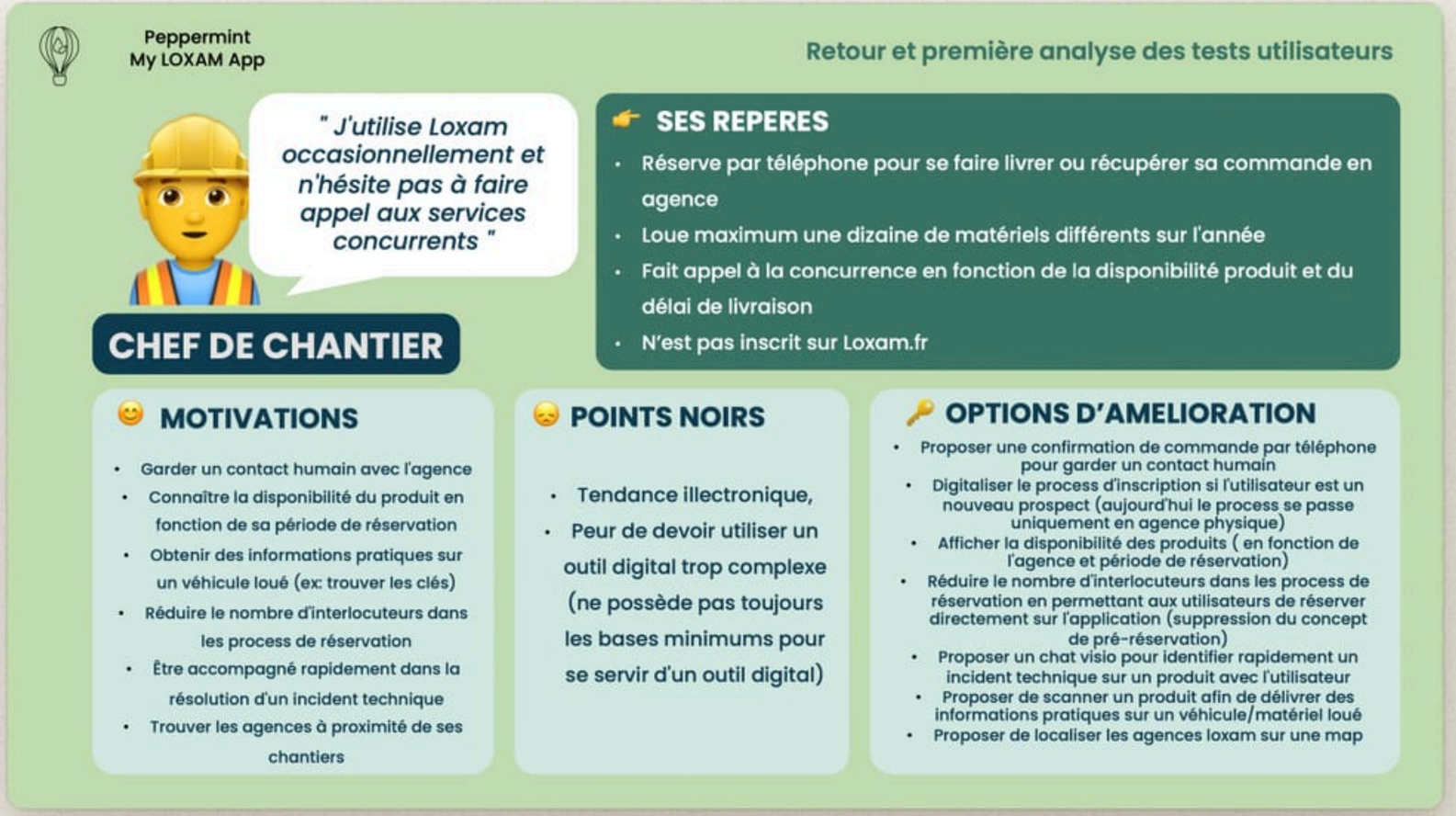
Users personae

On screen

The site manager: books by phone, rents up to about ten different machines a year, is not registered on Loxam.fr. Pain point: the fear of a digital tool that is too complex.

What it showed

Most users preferred to order by phone, and had to go to a branch in person to register a payment account.



Persona 1 of 3 · site manager, in French as delivered

Validation and invalidation of test hypotheses

Written before the sessions, settled by the tests.

Validated 10 hypotheses

01	Interest in a mobile app
02	Same machine types rented often
03	Real-time availability wanted
04	Most bookings made by phone
05	Time saved with the app
06	Booking status is reassuring
07	Personalised catalogue wanted
08	Vehicle scan to get details
09	Online rental documents needed
10	Change or cancel a booking alone

Invalidated 8 hypotheses

01	All comfortable with digital tools
02	All users can use a smartphone
03	Location of rented machines needed
04	Equipment reviews needed
05	Inspection report after booking
06	Incident reports from the app
07	24/7 assistance by message
08	Online card payment

What it led to

Set aside: card payment, chat, fleet, inspection report.

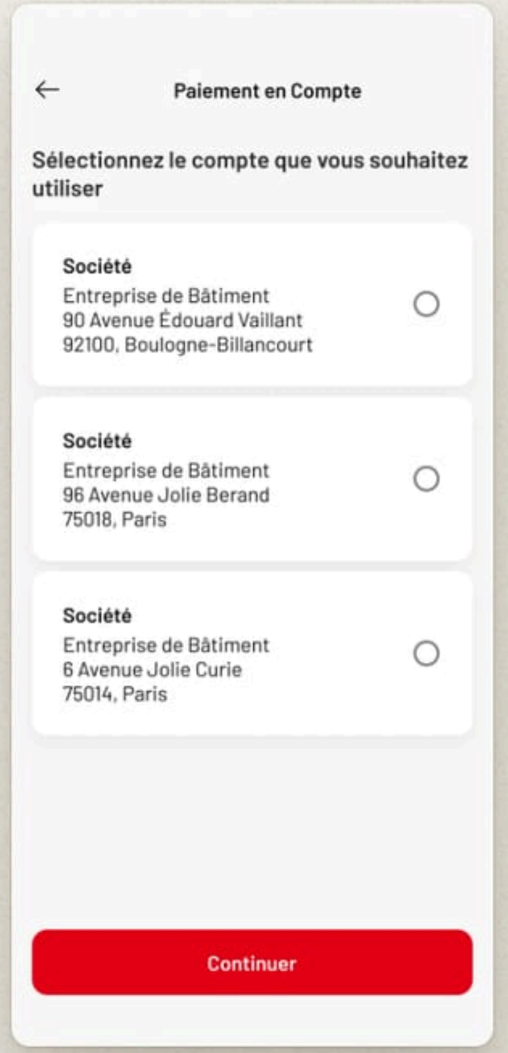
Presentation of results

Potential users of MyLoxam App are heterogeneous in terms of their appetite for using a mobile application. Even the more reluctant users pointed to gaps in their current journey, such as booking or checking availability on the move or outside branch opening hours.

The concept is relevant	It should focus on 2 key processes: the registration and the product reservation.
Prioritized issues	By times quoted, value, ease.
“Registration in 2 minutes”	Key path 1, for existing and new customers.
“Booking in 45 seconds”	Key path 2, from an item already ordered.
Expected benefits	6 for customers, 2 for Loxam staff.



Booking · sign-in with Face ID



Booking · payment on account

2 key paths first

MMP

First release

“Registration in 2 minutes”

“Booking in 45 seconds”

Prioritized by

Times quoted

Value to users

Ease of delivery

MMP + 8 paths

V1

Download invoices

Return request

Scan a vehicle

Branches nearby

Re-rent from past sites

Inspection report

Extend or view orders

Assistant

4 families

Later versions

Orders
invoices by e-mail, incidents, extend or shorten a rental

Home
scan from home, map of branches, voice search

Account
machines by past site, invoice filters

Catalogue
inspection report on the product page

9 features

Set aside

Assistant
chat, notifications

Fleet
map and list views, machine details, inspection report, damage report

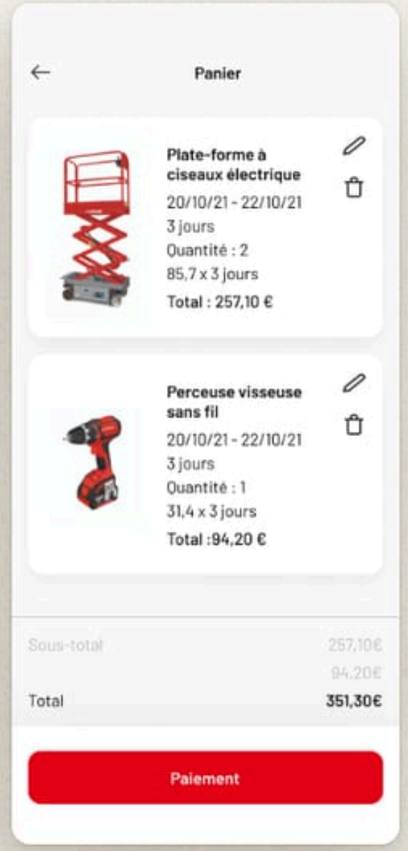
Catalogue
basket success page, card or Apple Pay

From V1 to the final prototype

V1 first session, 29 November



Product sheet



Basket

Session 1

The V1 funnel started in the catalogue and ended with a card payment.

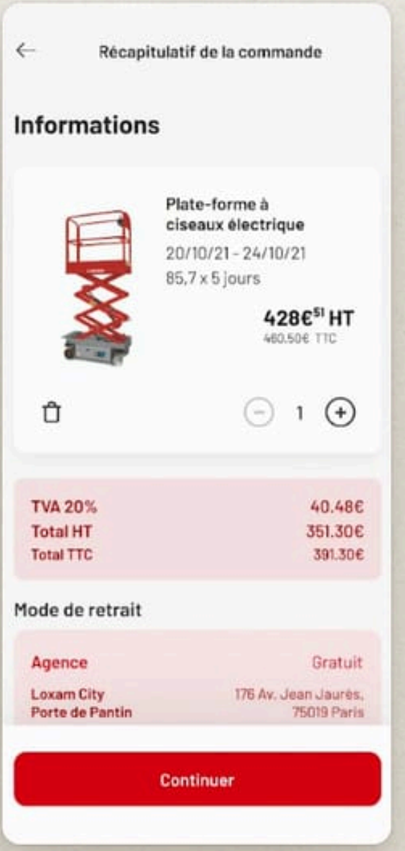
What changed

The pick-up in a branch chosen before payment, card payment replaced by the company to bill.

Final prototype booking path



Product and pick-up mode



Order summary

“Registration in 2 minutes”

On screen

A company that is not yet a customer starts its registration, imports its registration extract and bank details (KBIS, RIB), and is registered.

Expected benefit

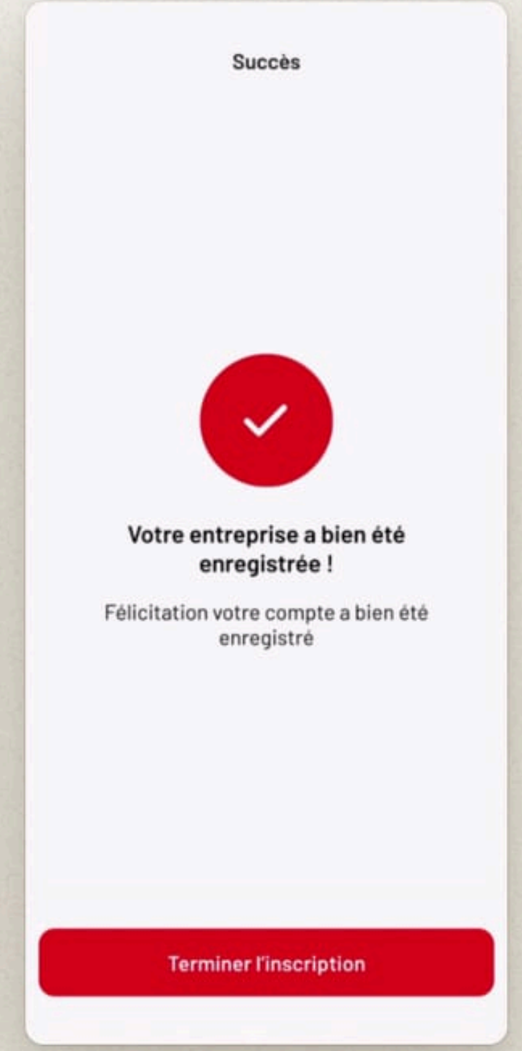
Registering a new prospect had to be done in a branch: time saved in the 24/7 registration process.



1 · New company



2 · Company documents



3 · Company registered

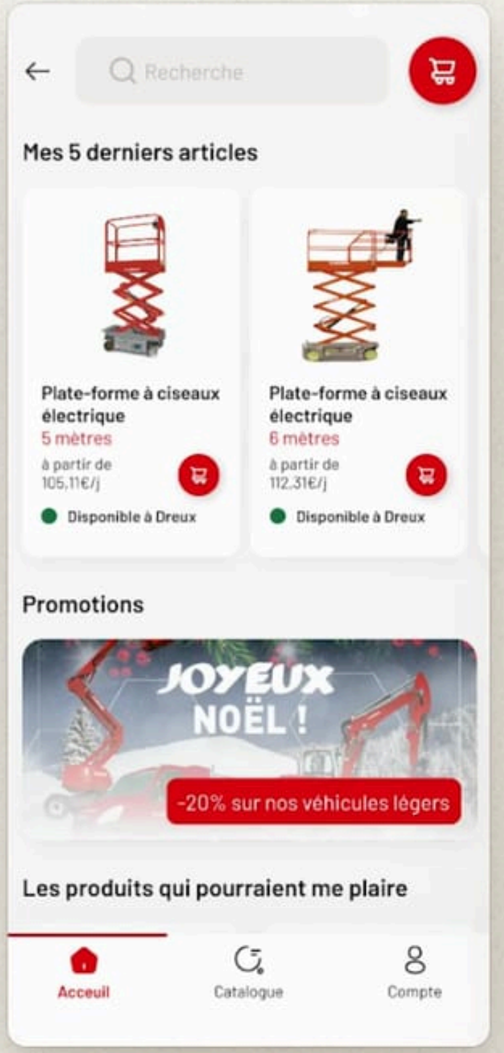
“Booking in 45 seconds”

On screen

Home opens on “My last 5 items”, each with its availability in the branch. Then the rental dates, and the confirmation by code or facial recognition.

Expected benefits

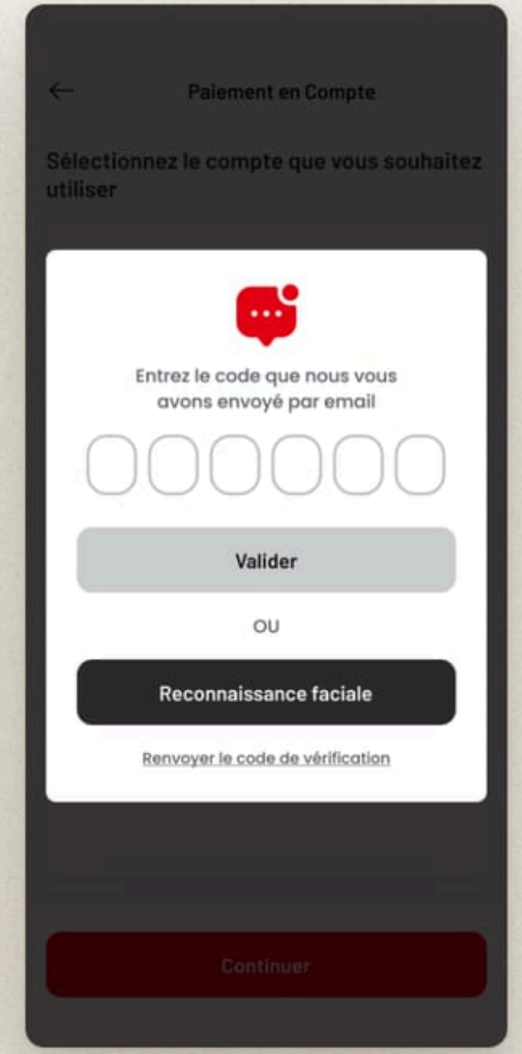
Save time on your reservation thanks to the possibility of re-renting an item already ordered. Limit travel with 24/7 real time machine availability.



1 · My last 5 items



2 · Rental dates



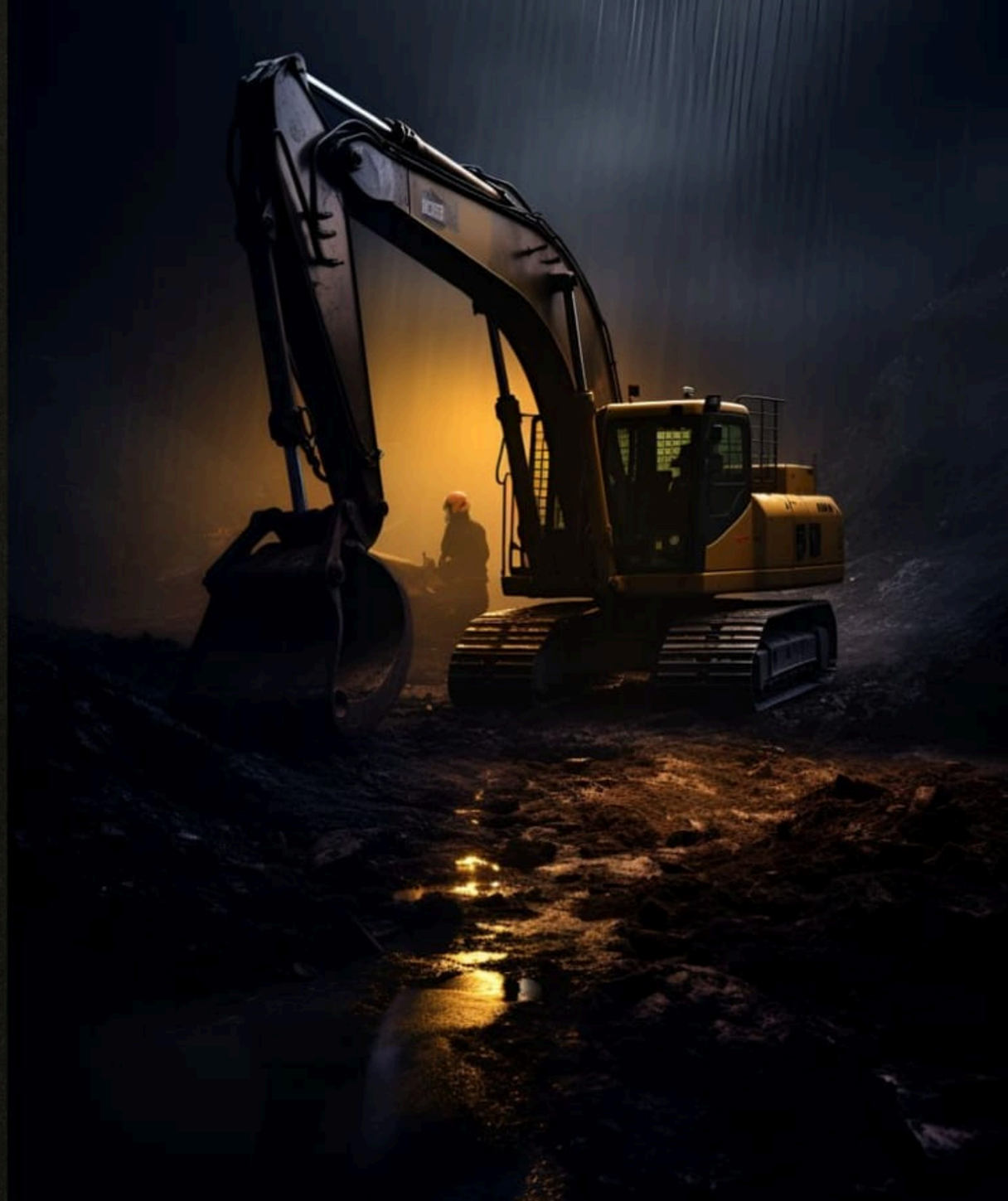
3 · Code or facial recognition

Key result · Loxam, mobile app

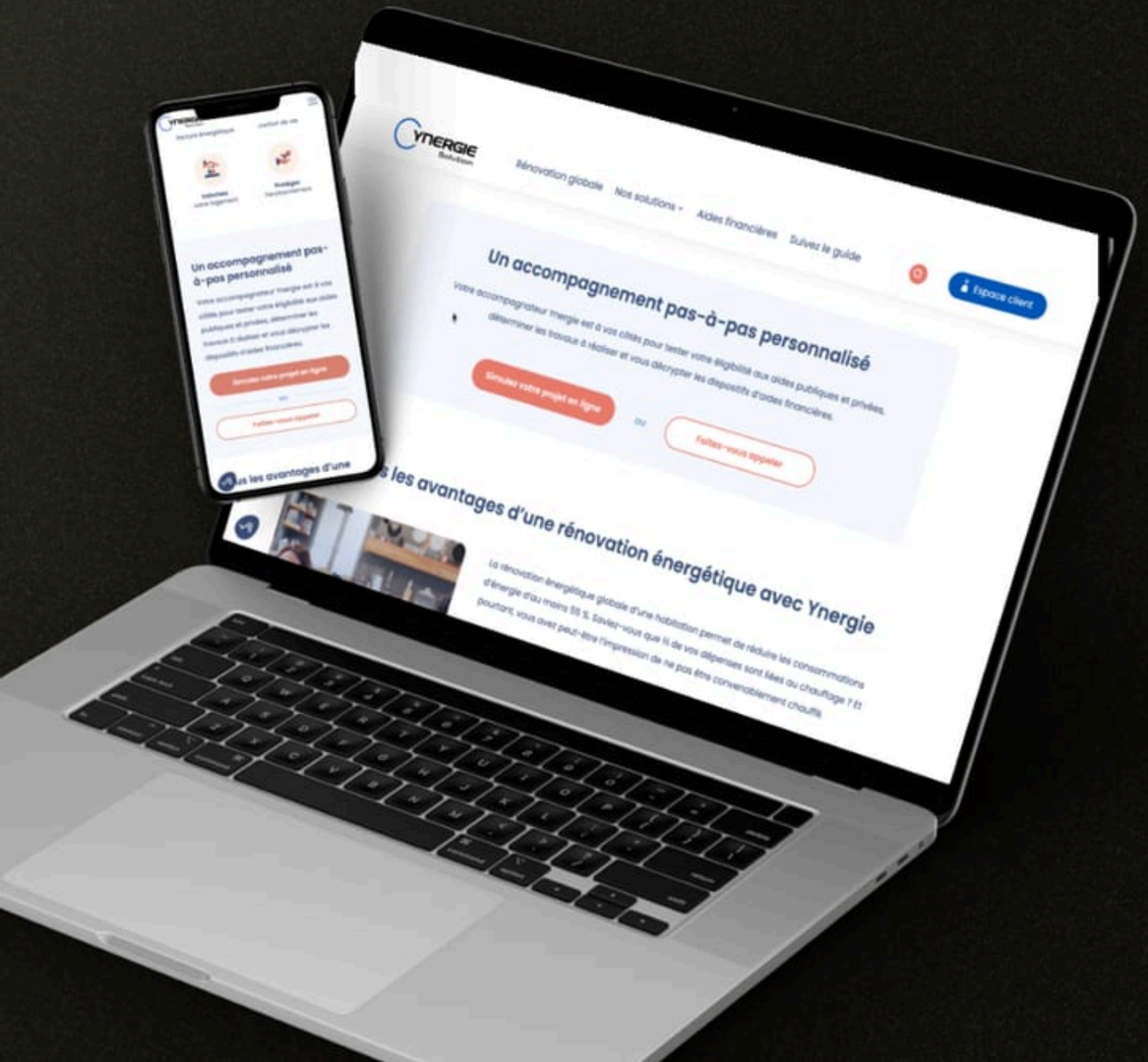
9 features removed before development

8 of 18 hypotheses invalidated · card payment, chat, fleet

- Mobile App Concept Validation: the concept is relevant, focused on the registration and the product reservation.
- Identification of new opportunities, from new features requested by users.
- UI Operational Kit, built on Loxam's graphic charter.
- Creation of an MVP in line with user needs.
- A Lean UX approach to obtain rapid results based on the needs of end users: 3 RITE test iterations in 1 week, with 18 to 24 users.



Ynergie



Create a B2C Web App

Ynergie is a company specialized in energy renovation for individuals. Ynergie had contacted my company to evolve its vision and business strategy and to offer new services. A total of 9 sprints were sold to the client to build a B2C web app and redesign their website.

My role	Product Designer
Crew	1 PO, 1 CRM Manager, 3 Growth Hackers, 3 Product Designers, 6 Devs
Problem	The company needed reinforcements during sprints 3 and 4 to support its design team in the production of desktop models of the web application and the performance of user tests on the mobile version.

Mission 01	Understand	5 sub-missions
Mission 02	Design of the desktop version of the web app	4 sub-missions
Mission 03	PO reviews	2 sub-missions
Mission 04	User Testing	5 sub-missions
Mission 05	UX Evangelization	3 sub-missions

Understand

I was asked to help the design team, which had difficulty meeting its production deadlines.

Analysis of the customer's service offer	The services offered by the customer, studied first.
The actors involved in the services	9 actors, 3 families.
Organization of the internal product team	Who does what in a 15-person product team.
Backlog and roadmap	To set my scope of intervention.
Working with the design team	What was done, what was expected, the design system.

Main actors	
B2C customer	Individual house owner
Accompanying project leader	Dedicated, from A to Z

Secondary actors, internal	
Ynergie Solution	Mandates and quotes
Ynergie Opération	Estimates and purchasing
Ynergie Valorisation	Financial support
Ynergie Groupe	Pays the artisans

Secondary actors, external	
Craftsman	Carries out the work
Vehicle partner	Business partner
Convenience/auditor partner	Uploads key files

Design of the desktop version of the web app

The Ynergie web app has been developed in mobile first. I was asked to decline the screens in desktop version on 2 key paths: the pre-diagnosis stage of the renovation project and the diagnosis stage.

Pre-diagnosis step	The first login: password creation after the email.
Diagnosis step	1 page to follow housing, work and financing.
Building a KIT UI on Figma	The desktop components, in the team's design system.
Prototyping of interactive models	Interactive prototypes on Figma.



Diagnosis page, desktop · as staged in my original deck

Diagnosis: housing, work and financing on 1 page

On screen

The same section, mobile then desktop. My desktop page: housing, work and financing, each with its own tabs from estimate to invoice or payment.

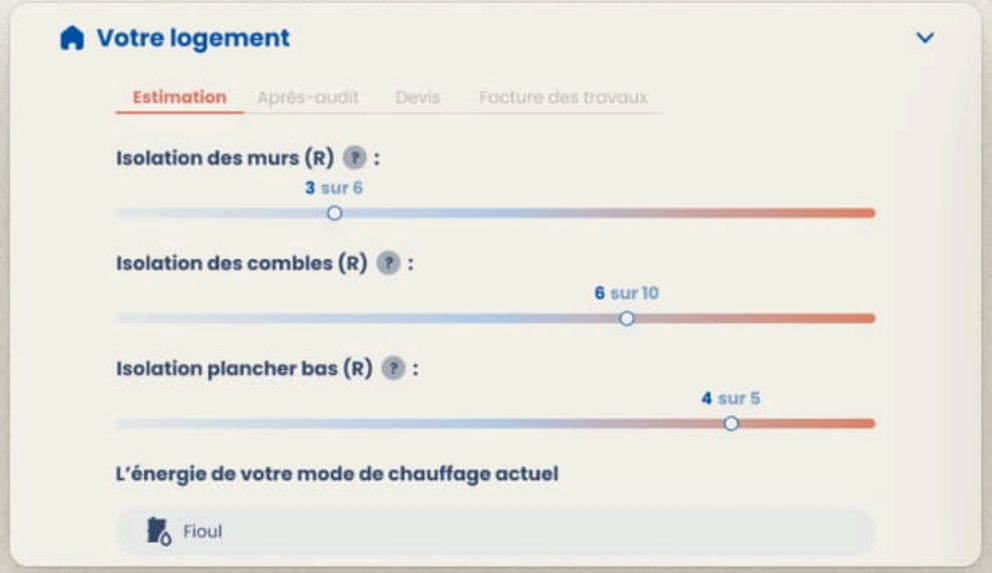
What changed

The web app had been developed in mobile first. At the end of the sprint, the desktop mock-ups for the pre-diagnosis and diagnosis steps were delivered.

Mobile



Your housing Votre logement · desktop



Your work Vos travaux · desktop



Mobile by the design team, previous sprints · desktop, my screens

PO reviews

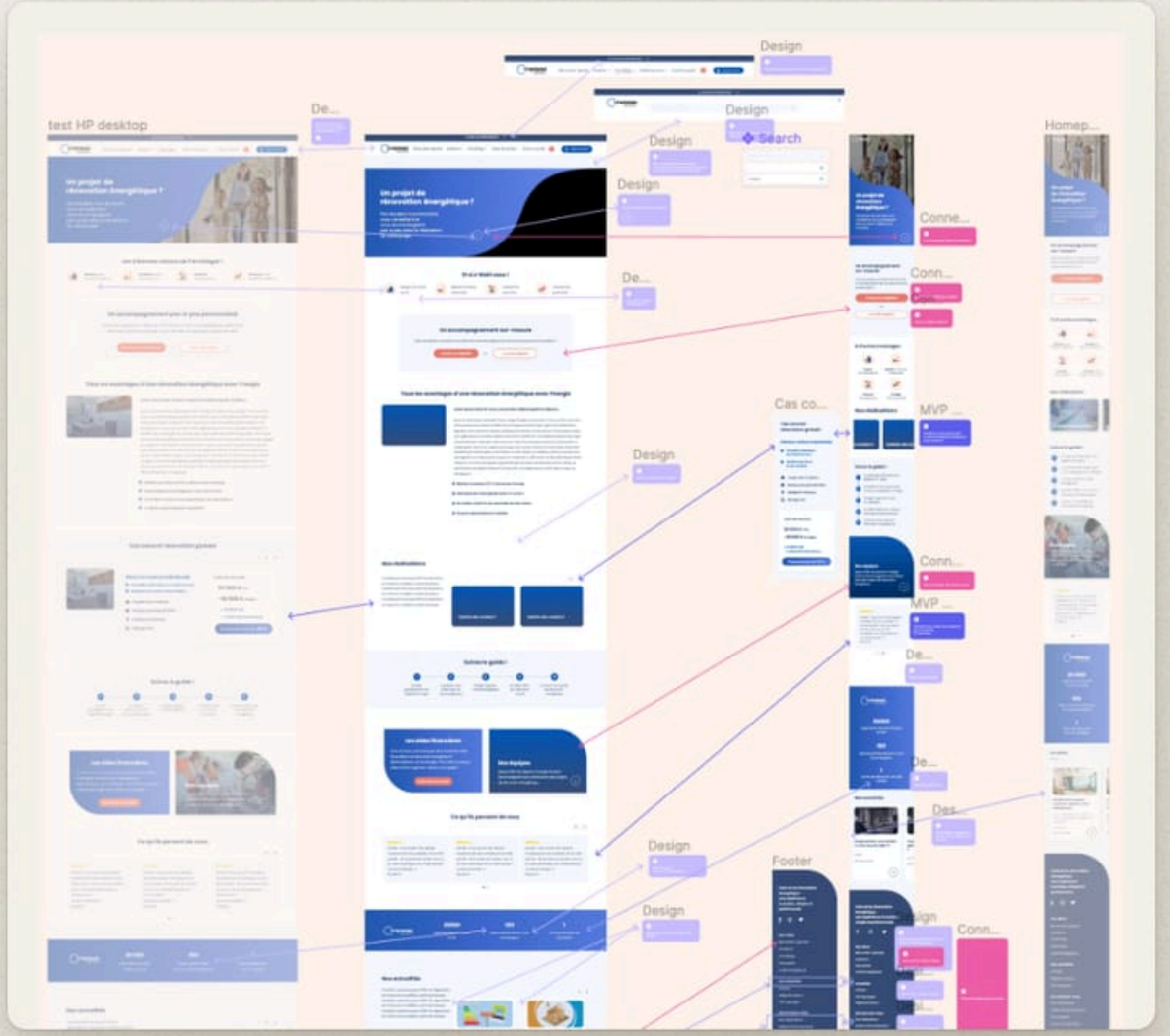
The PO review consisted in helping the PO to give feedback on the integration of the models developed on Figma and to keep the developers informed of updates in Jira tickets.

Feedback on the integration

The integrated homepage, “test HP desktop” on the left, checked against the Figma mock-ups, desktop and mobile.

Jira tickets

A comment on each gap (Design, MVP), every update written in Jira tickets for the developers.

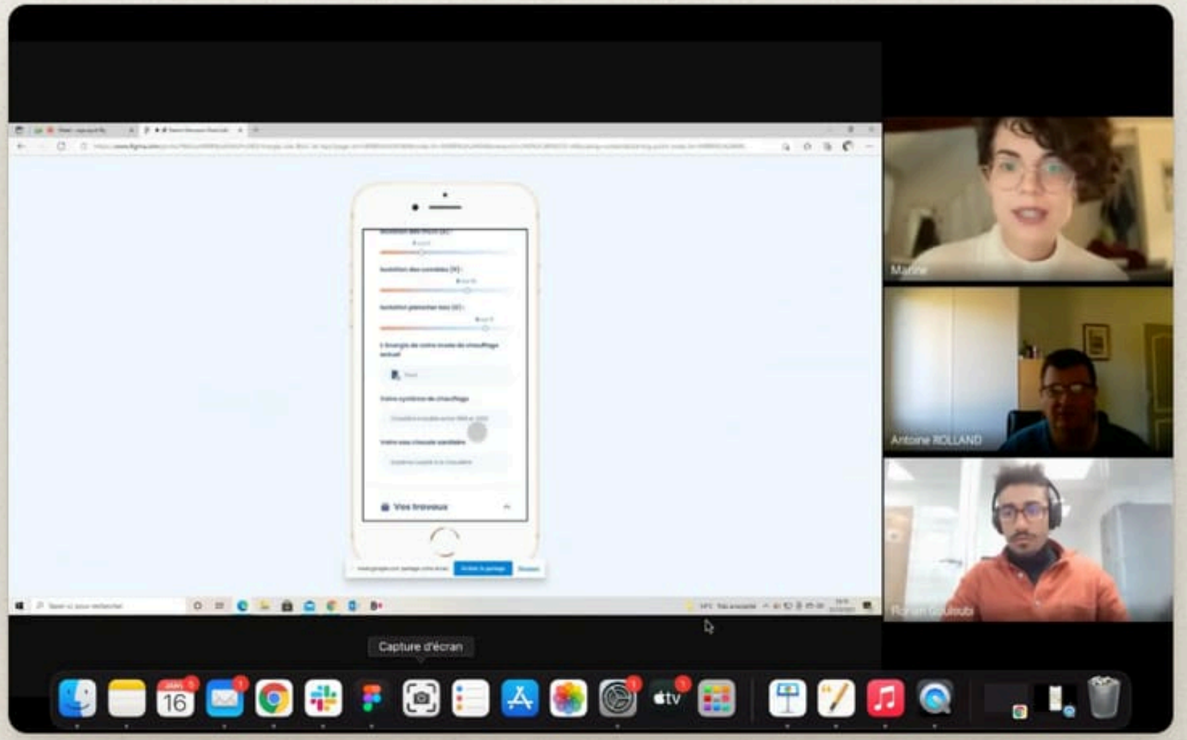


PO review on the B2C site mock-ups · my review board, top

User Testing

I participated in remote user testing sessions with end users to assess the usability of mockups that had been made in previous sprints.

Testing guide	The guide for the sessions.
Face-to-face and remote test session	With end users, on the mobile version.
Analysis with tagging plan	Hypotheses validated or invalidated, new features requested by users.
Recommendation	Models corrected, key paths prioritized, unnecessary flows removed, new paths proposed.
Presentation of the results to the client	Synthesis and recommendations.

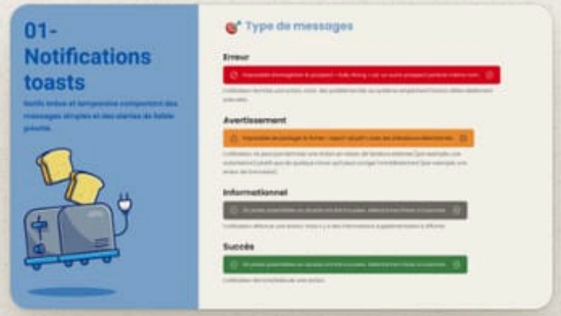


Remote session on Google Meet · 22/10/2021
Tester's screen: Figma prototype "Demo Parcours Post-Call", mobile
Step: Votre logement, then Vos travaux · me, bottom right

UX Evangelization

My lead designer informed me that the developers had started to integrate toasts throughout the application to replace error messages. She asked me to track best practices so she knows when toasts should be used to guide teams on do’s and don’ts.

Monitoring of good practices	Best practices on toasts, from 2 references.
Recommendations	Remove the error toasts, put the error messages back in place as on the mock-ups, align with the design system.
Evangelism	Presented to the client and the product team, as guidelines on what to do and what not to do.



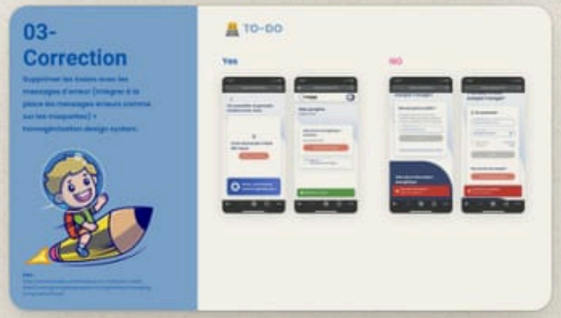
Toast notifications

01 · 4 message types



Mistakes to avoid

02 · 6 don’ts



Correction

03 · Yes and No, on the build

Toast notifications: 4 message types

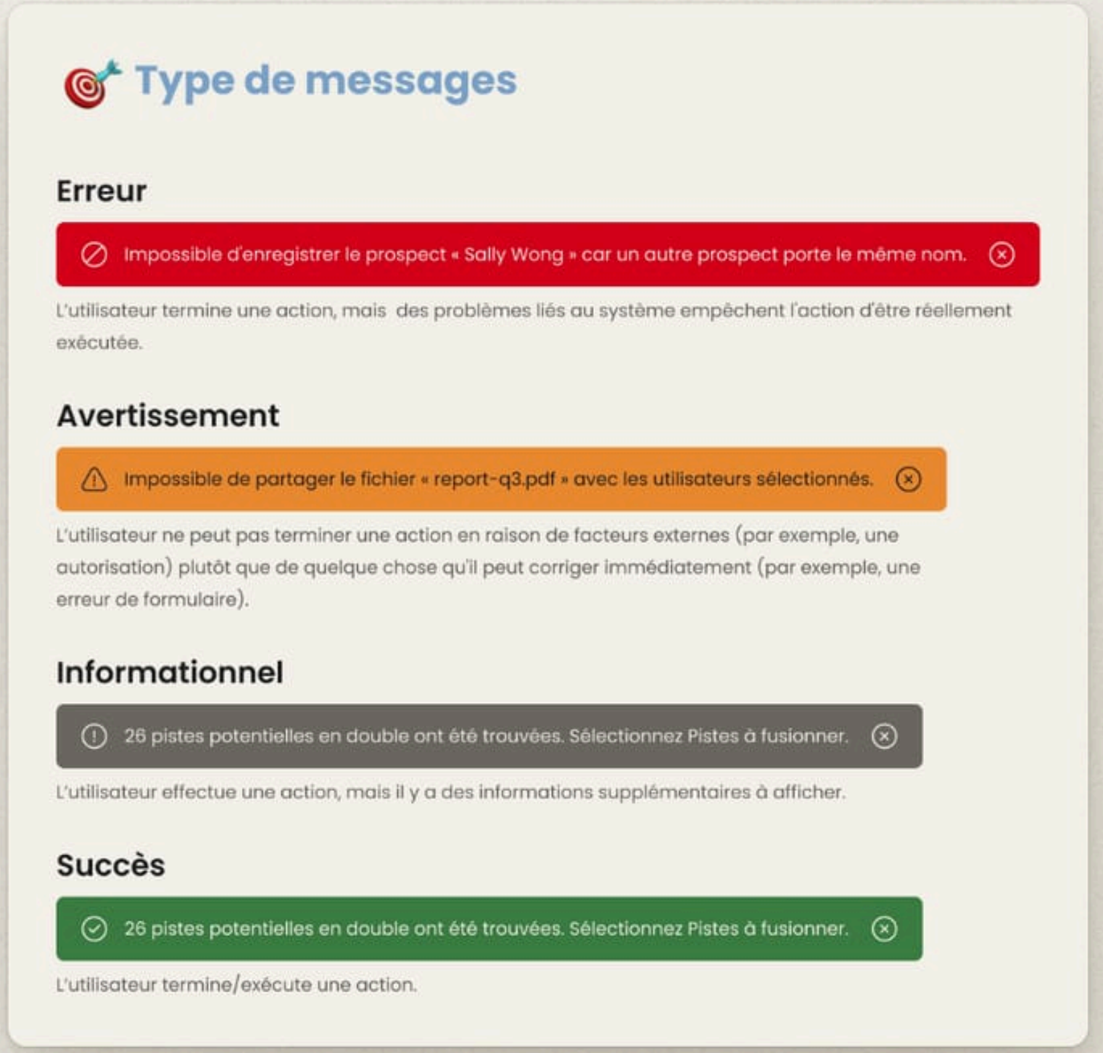
On screen

My board, translated: toasts are brief, temporary notifications with simple messages and low-priority alerts.

Error	The user completes an action, but a system problem prevents it.
Warning	External factors, such as a permission, block the action, not a form error the user can fix.
Informational	There is more information to show.
Success	The user completes an action.

What it led to

I presented the results of my research to the client and the product team in order to set up guidelines on what to do and what not to do.



Board 01 · bar texts from Lightning Design System

Mistakes to avoid: 6 don’ts

On screen

My board, translated. The tip: a toast confirms an action, and rarely carries an error message.

What changed

The No screens of the build show user errors in toasts: “Email non enregistré”, “Invalid credentials”.

1 No toast for user errors, only for system problems.

2 No success toast when a creation opens the new item.

3 No auto-dismiss for information that must not be missed.

4 No toast for insignificant feedback: it distracts.

5 No endless feed: that calls for a notification centre.

6 A toast that can cover content must be dismissible.

Board 02 · from benrajalu.net and Lightning Design System

When should a toast be used?

On screen

4 screens of the development build. Yes: a toast that confirms, “Bienvenue à Ynergie” (Welcome to Ynergie). No: toasts that carry a form error, “Email non enregistré” (Email not registered).

What changed

My correction: remove the toasts with error messages, put the error messages back in place as on the mock-ups, and align with the design system.

Sources

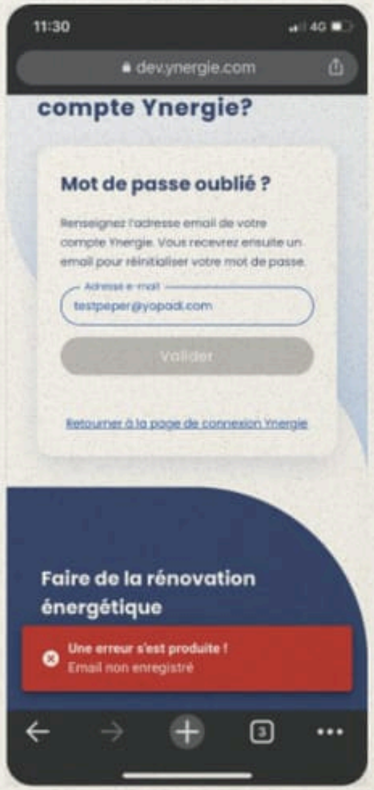
benrajalu.net, UX of notification toasts
Lightning Design System, toasts

Yes





No









Development build, integrated by the developers · toasts enlarged below

2 key steps delivered for desktop in 1 sprint

Pre-diagnosis and diagnosis. Joined at sprint 3 of 9 to reinforce the design team.

- Mature UI Kit: the desktop components, in the team's design system.
- PO reviews on both sprints, every update in Jira tickets for the developers.
- User tests helped us validate our delivery and reassured the customer that we keep a user-centric approach during the construction of the MVP.
- Evangelization of UX/UI best practices within the product team: toast guidelines presented to the client and the product team.
- The new B2C site has been launched.



Aleia



Delivery Park
Create a B2B app

DeliveryPark is a product designed by Aleia which lists the stops of urban delivery people and which therefore allows local authorities to manage their planning and urban regulation policies.

My role	Product Designer (Solo Designer)
Crew	Solo Designer, 1 PM, 1 Digital and Marketing Director, 1 PO, 1 Dev
Problem	Today delivery people park in unregulated spaces in Paris and complain that they do not have access to a sufficient quantity of delivery space. I was asked as a designer to build from scratch a delivery area location application for the B2B sector.

Mission 01 **Understand** 4 sub-missions

Mission 02 **Graphic Identity and UI kit** 2 sub-missions

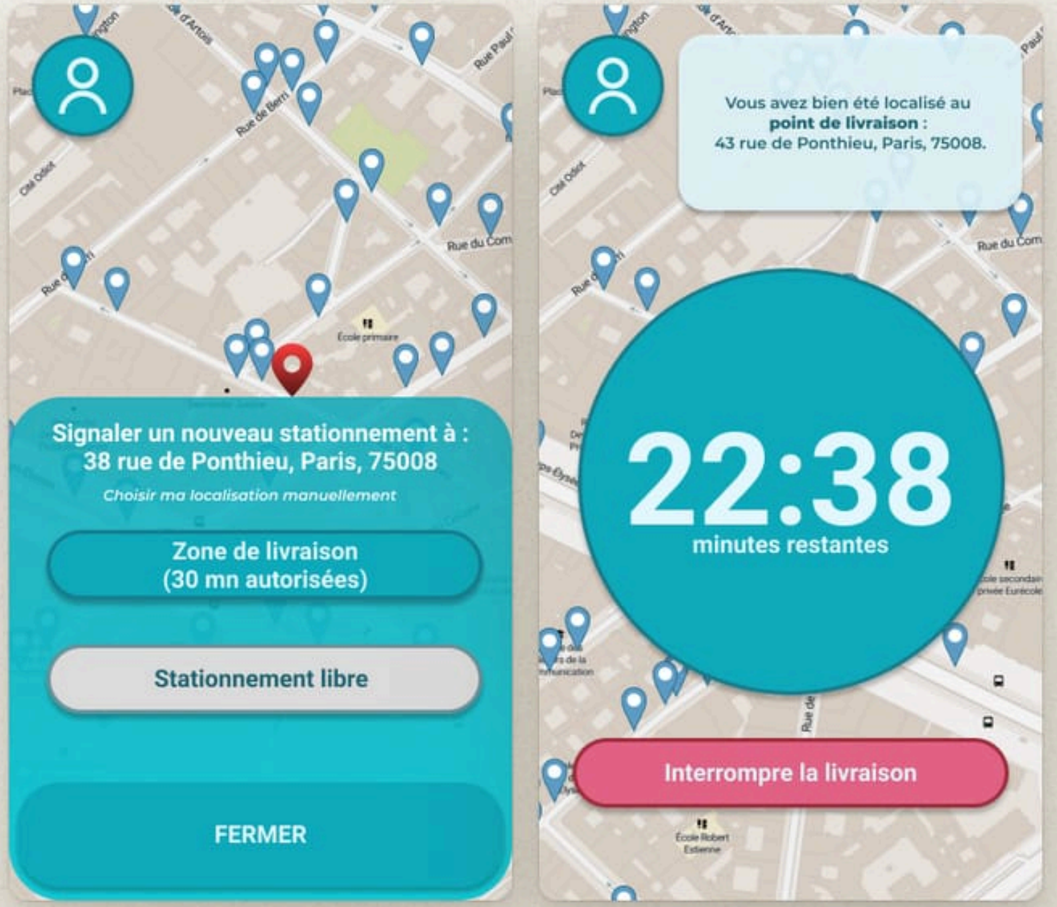
Mission 03 **App design** 6 sub-missions

Understand

A specification with wireframes had already been made by the client: he wanted a UX/UI redesign of the paths and a design system for the developer. I did a quick audit of the existing system and a user-flow, then organized a meeting with the developer to know the technical limits of the project.

Project Planning	Sprint 1: kick-off, graphic charter, mock-ups, client presentation.
Quick audit of the existing system	The client's spec and wireframes.
User-flow of the delivered paths	To define the missing screens.
Analysis of technical constraints	With the developer, upstream.

Client's wireframes · before my redesign



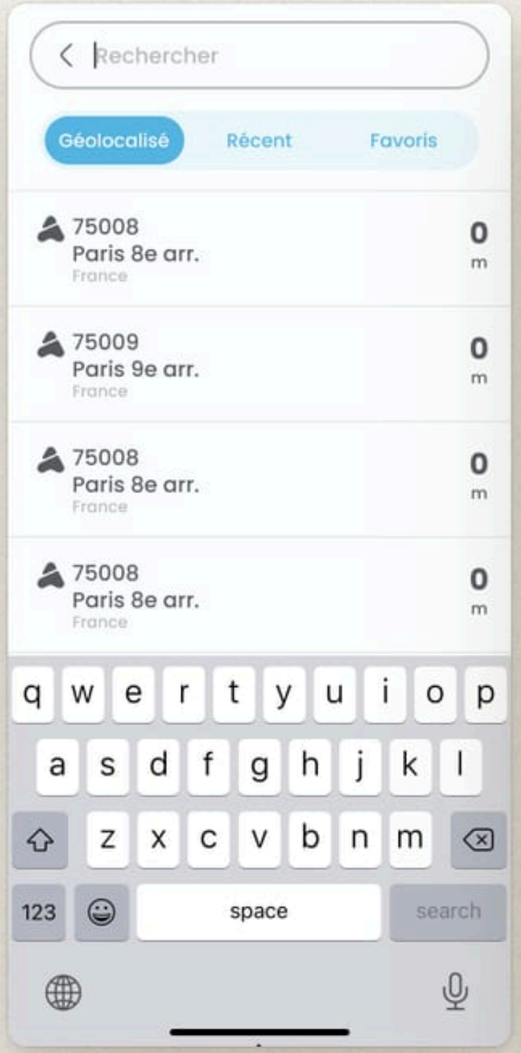
A list of delivery spaces in a key district

On screen

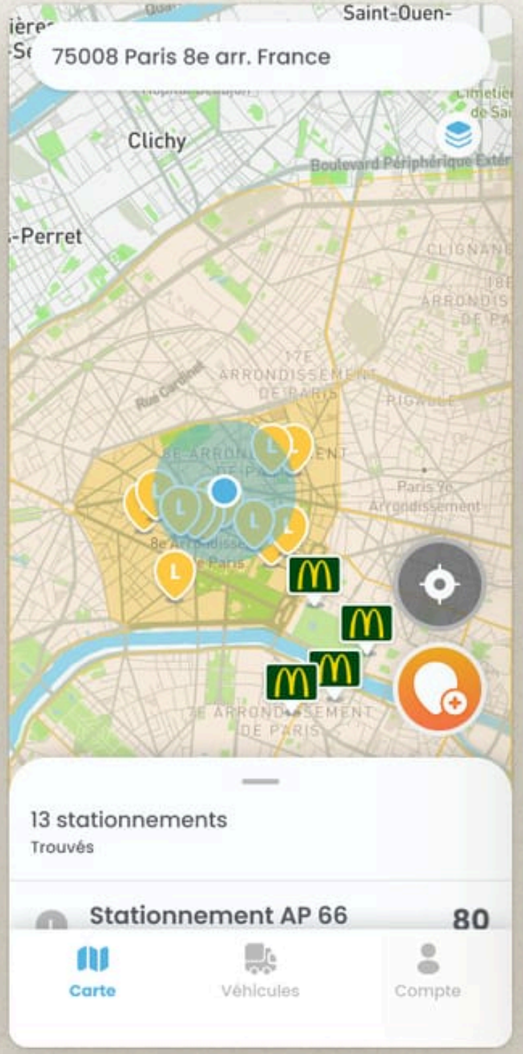
The driver chooses a district (75008, Paris 8th), the map highlights it, and a drawer lists its 13 parking spaces with their distance.

Why

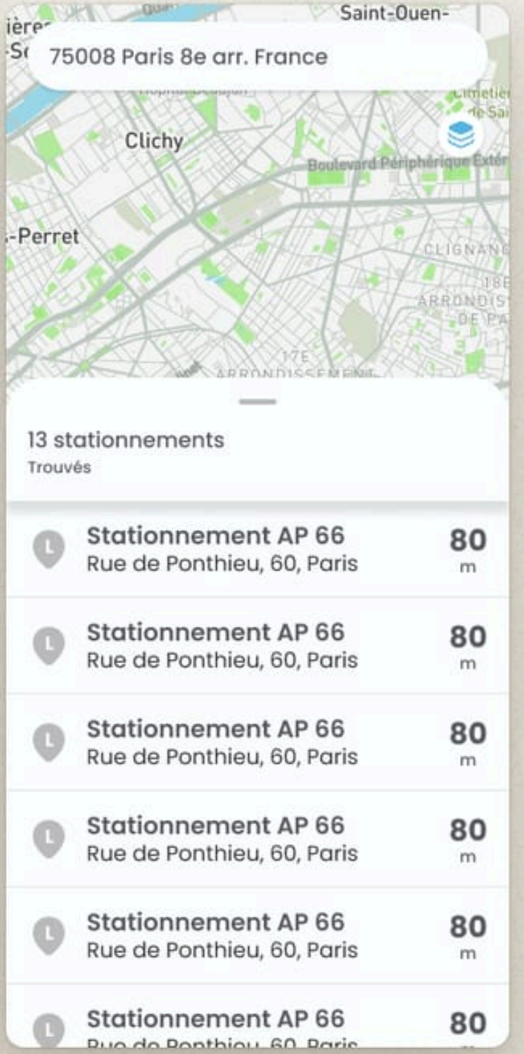
We could not display all the delivery spaces on all the map of Paris at the same time: too many elements to load, so a considerable waiting time for the user. The solution chosen was a list of delivery spaces in a key district, to gain in performance.



1 · Choose a district



2 · The district on the map



3 · Its 13 parking spaces

Graphic Identity and UI kit

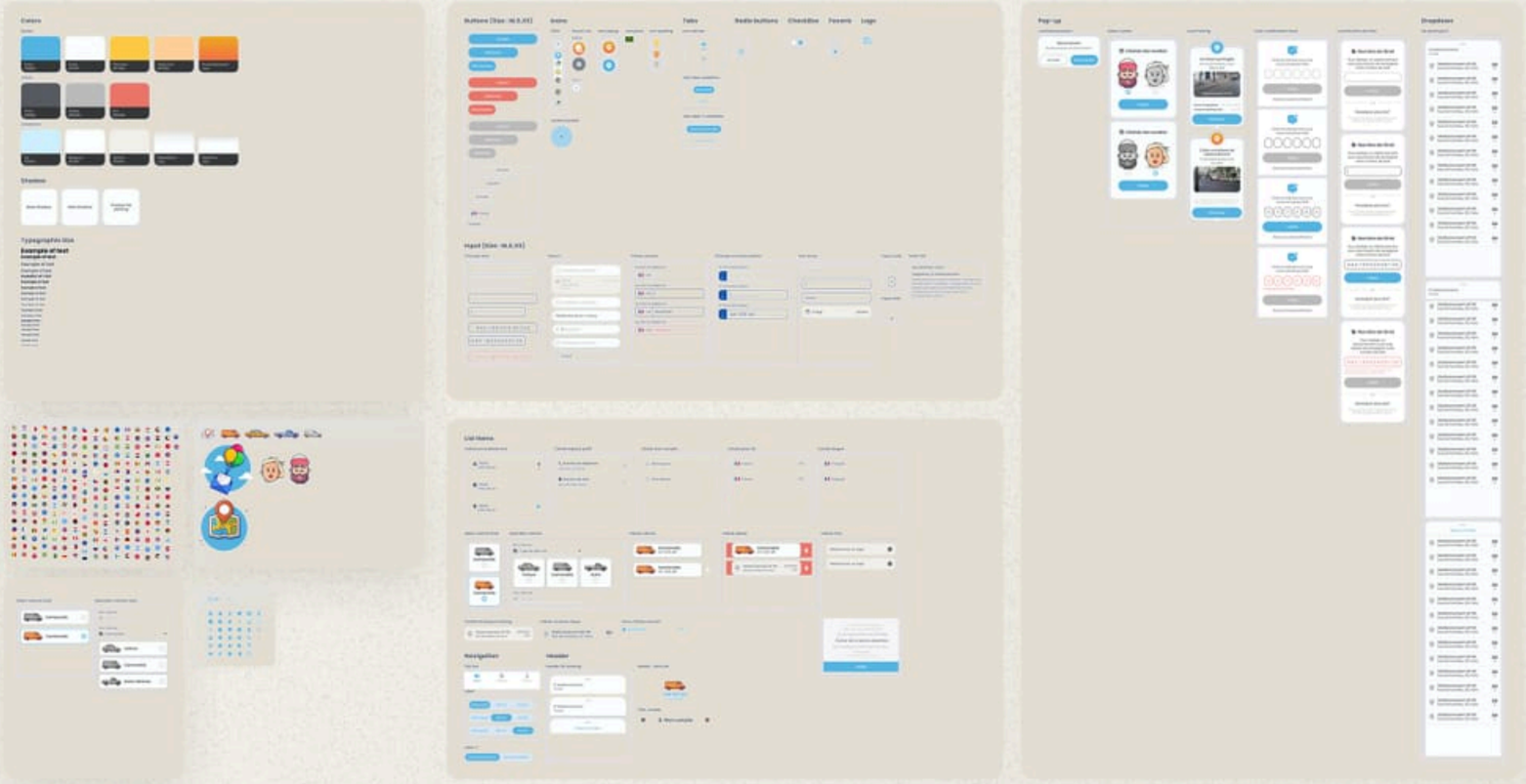
The product had no graphic charter. I first started by creating the graphic identity of the product before building a UI kit.

Creation of a graphic identity

From scratch: colours by role, shadows, type scale.

Creation of a UI kit

An actionable UI kit for developers, each component with its sizes and states.



UI kit · overview: colours, type, buttons, inputs, list items, pop-ups

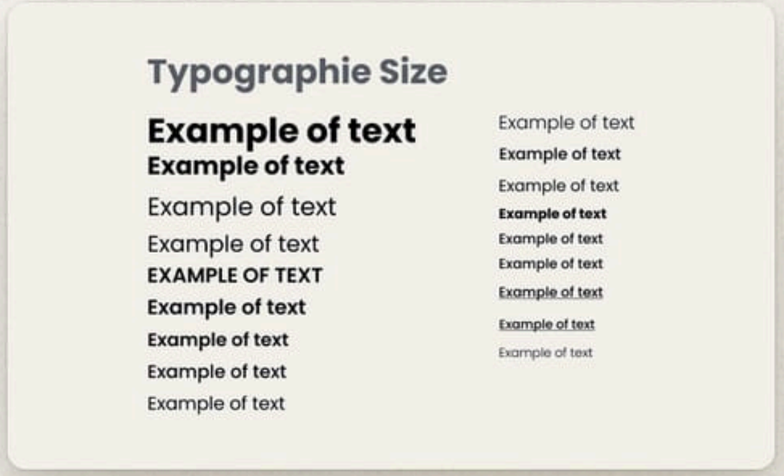
Creation of a graphic identity

On screen

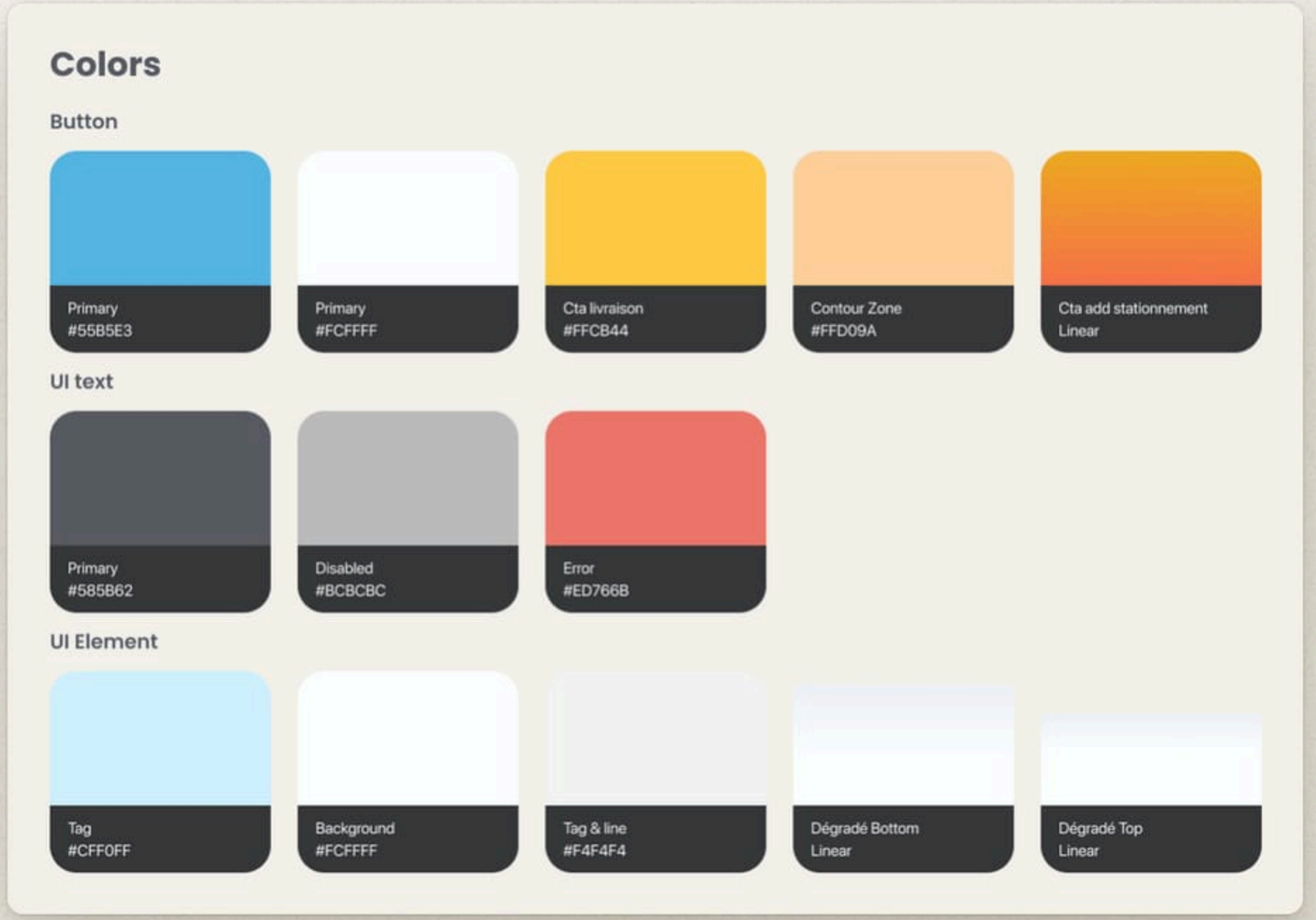
The colours of my graphic charter, named by role, with their codes: button, UI text, UI element. Its type scale: 18 text styles.

What changed

The product had no graphic charter: I created it from scratch.



Typography · 18 styles



Design system · Colors

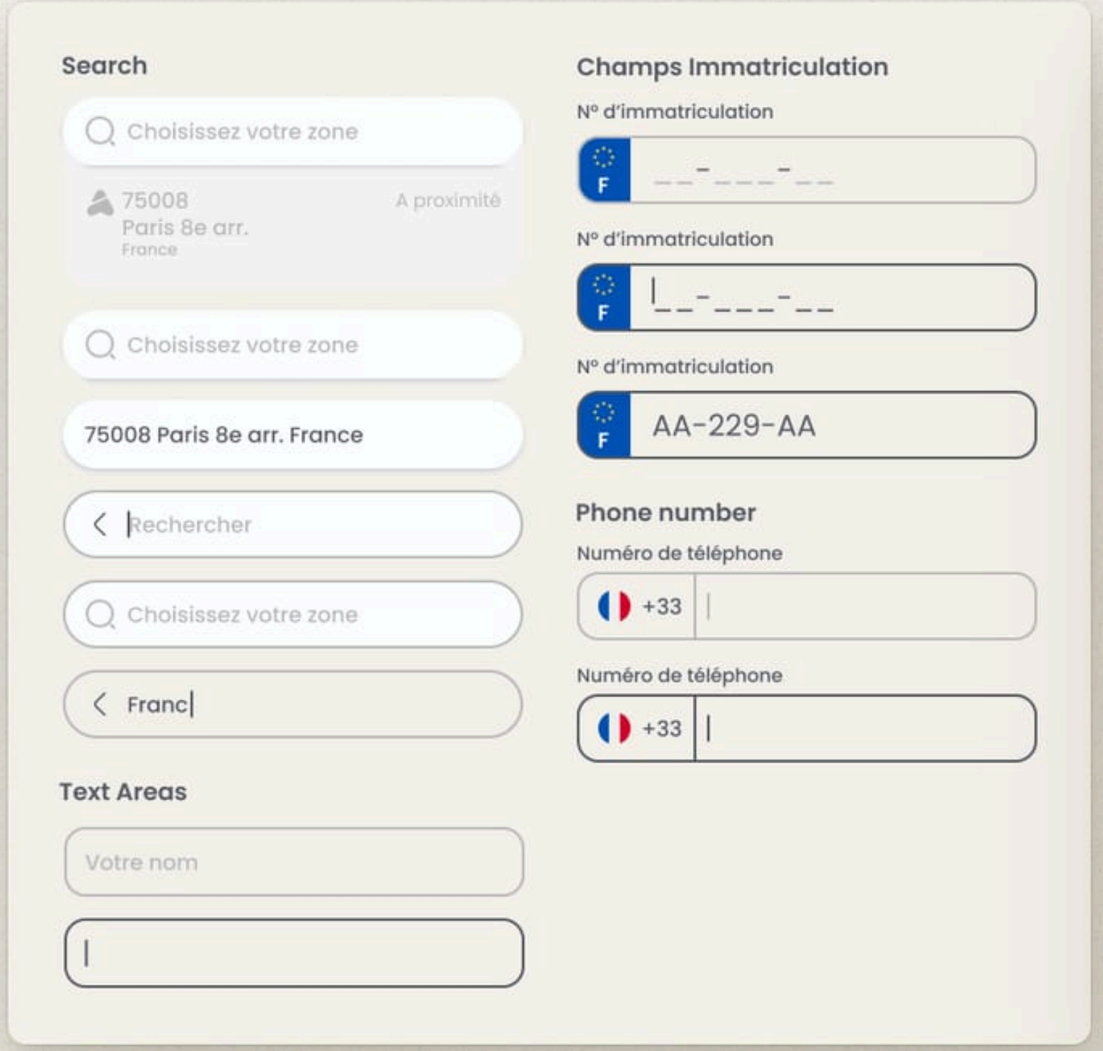
Creation of a UI kit

On screen

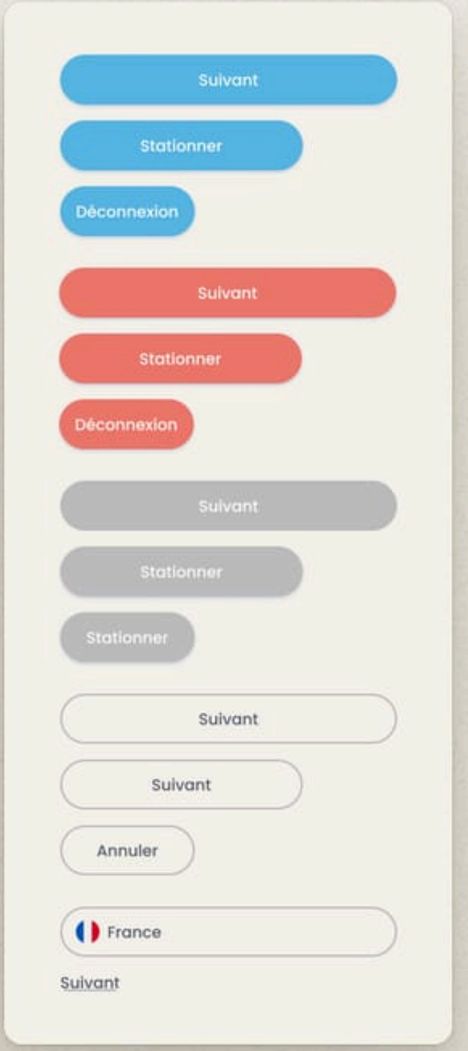
Inputs: the district search in 6 states, the French licence plate in 3, the phone number and the text areas, empty and active. Buttons in 3 sizes (M, S, XS) and 4 styles.

What changed

A design system for developers: each component with its sizes and states.



Input · search, licence plate, phone, text areas

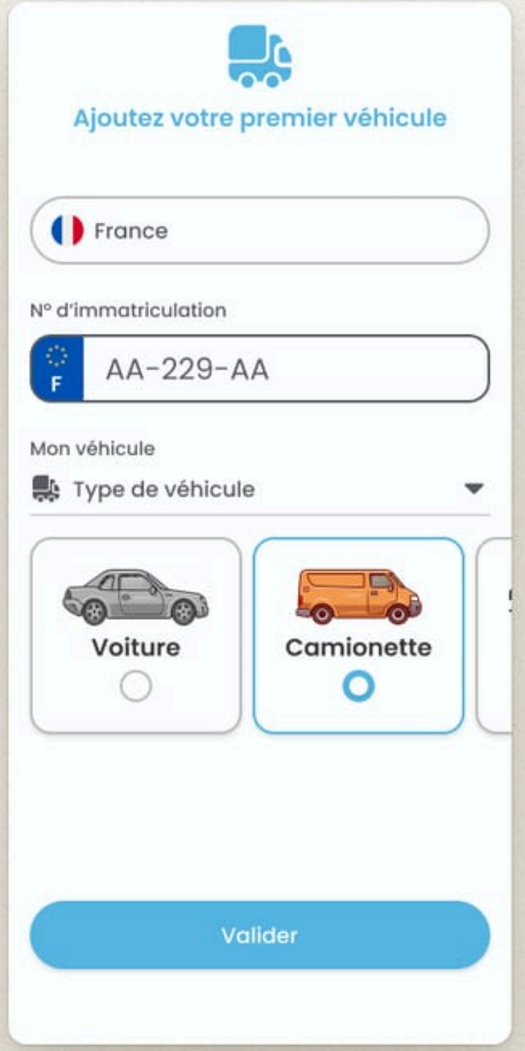


Buttons · M, S, XS

App design

Once my benchmark with the competing applications was done and my UI kit was ready, I moved on to the UX/UI redesign of the wireframes: 4 journeys, validated with the client.

Benchmark	Before the first screen.
Onboarding	Phone, SMS code, first vehicle, GPS.
Declare a parking in a shared delivery zone	Zone card, Siret, 30-minute timer.
Create a parking space	Where no delivery zone exists.
Manage your account	Profile, history, report, FAQ, vehicles.
Handoff management	Mock-ups, design system, demo videos.



Onboarding · first vehicle



Onboarding · GPS access

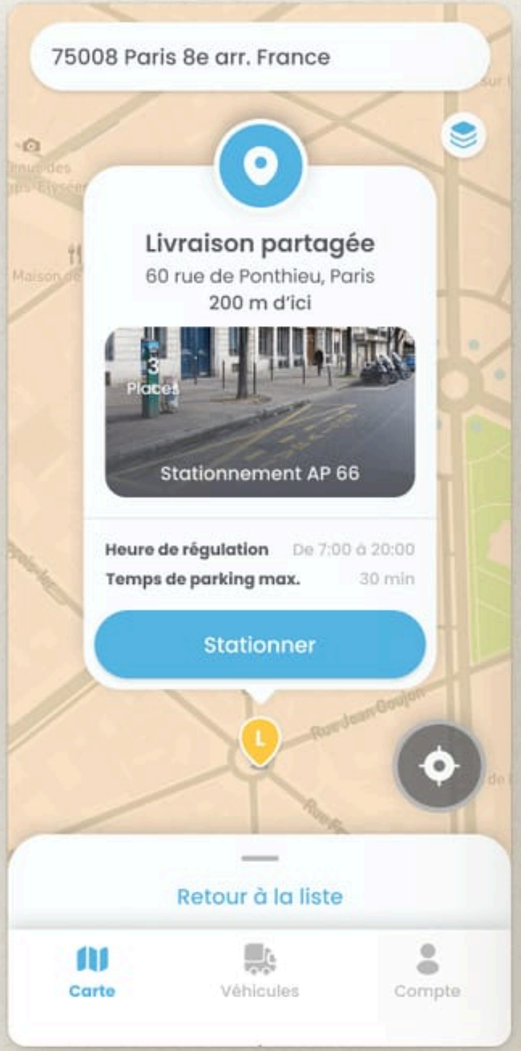
Declare a parking in a shared delivery zone

On screen

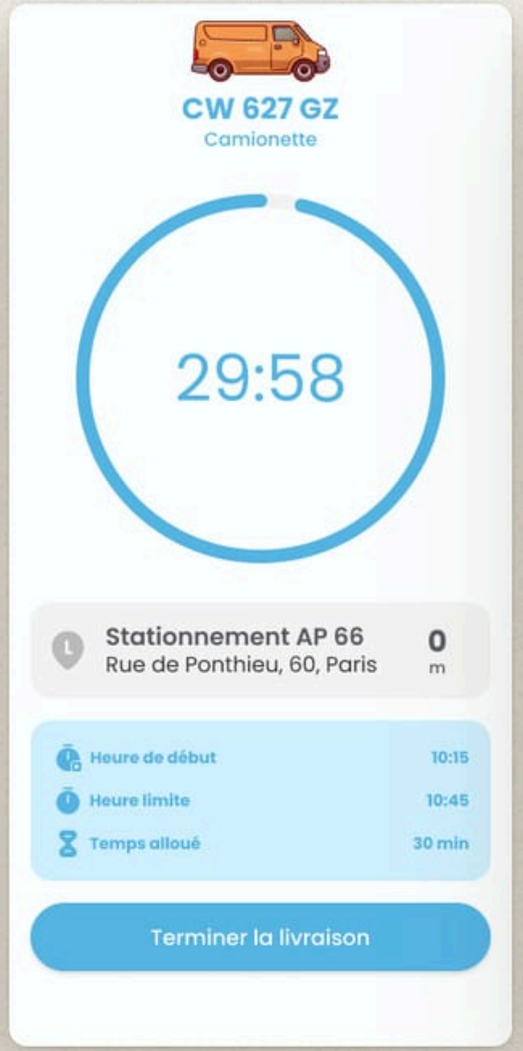
From the district list, the zone card: photo, regulation hours (7:00 to 20:00), maximum parking time (30 min). Then the countdown with start time, time limit and allotted time.

What changed

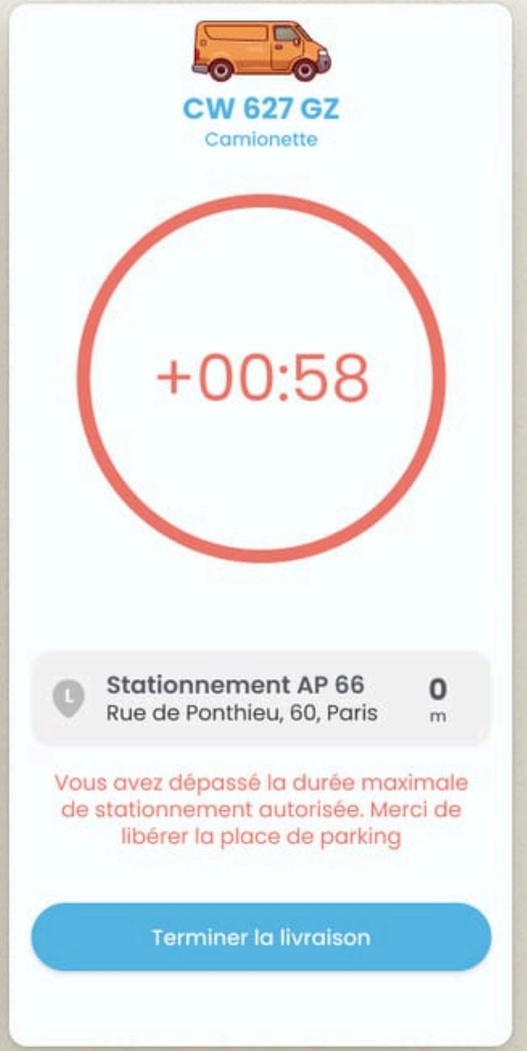
The client's timer showed the minutes and the address. Mine adds the vehicle, start time, time limit and allotted time, and past the limit asks the driver to free the space.



1 · Shared delivery zone



2 · Countdown



3 · Time limit passed

Create a parking space

On screen

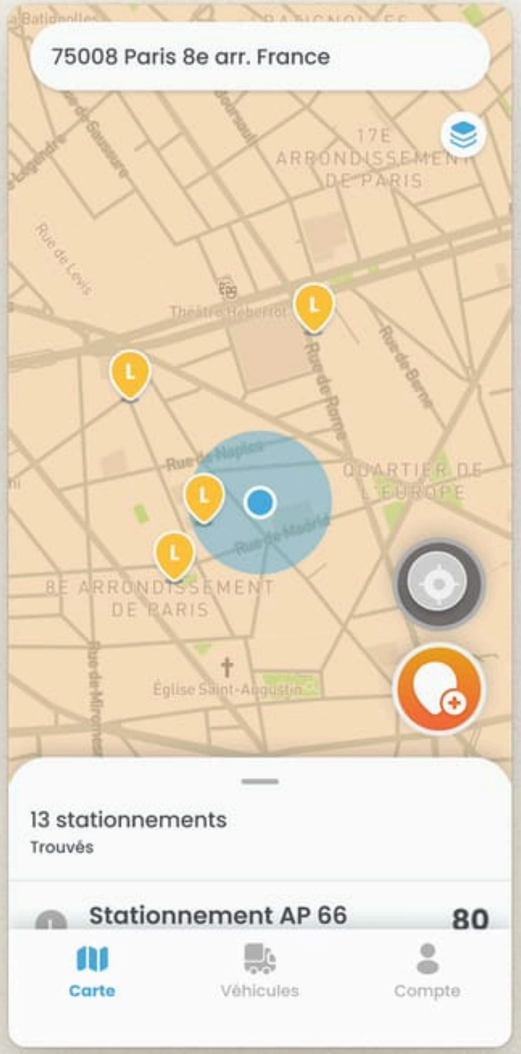
The round orange button on the map opens a card to park at the driver's own position (75 Bd Malesherbes, 0 m away), with a photo and a notice: "DeliveryPark cannot be held responsible for a fine."

What changed

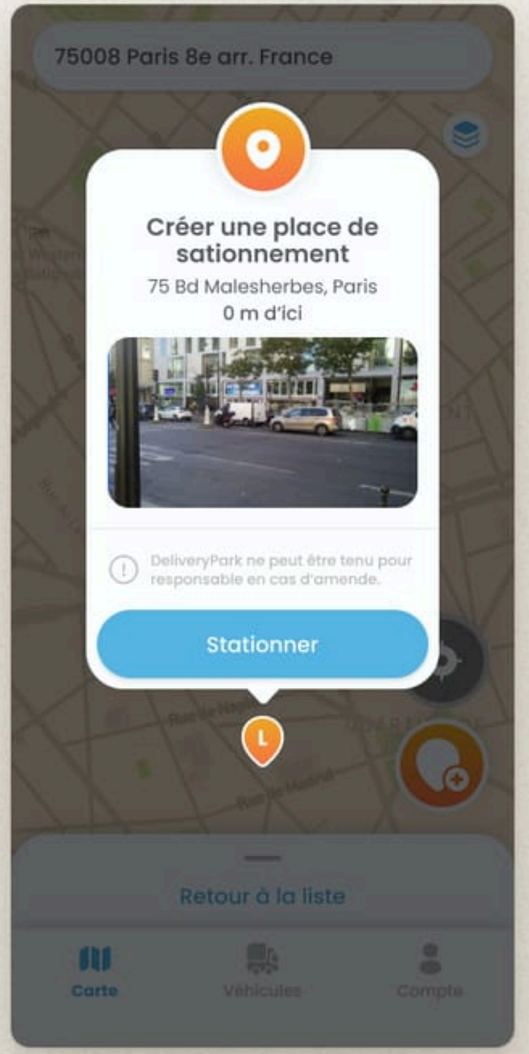
In the client's wireframes, free parking was one choice in the report pop-up. In mine it has its own button and its own card.



Close-up · the notice on the card



1 · The orange button on the map



2 · Create a parking space

Manage your account

On screen

My account: profile, history, report a problem, FAQ, log out. The history in 2 tabs, free parking and delivery zone. The FAQ asks “Are you satisfied with the answer?”

What changed

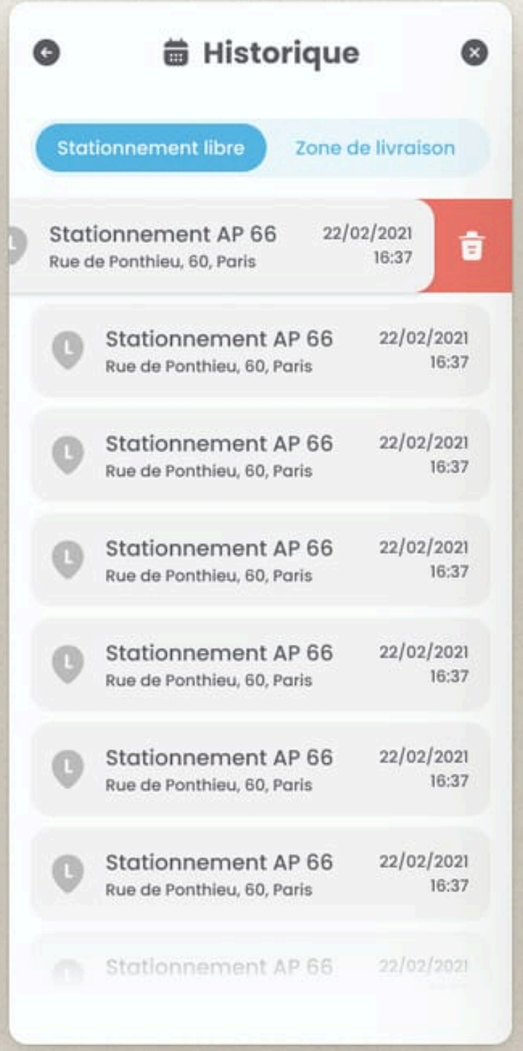
The client had a menu and a FAQ behind a “?”. Mine groups them in My account, adds a profile, 2 history tabs, a FAQ search and feedback; vehicles move to the tab bar.



My vehicles · swipe to delete



1 · My account



2 · History, swipe to delete



3 · FAQ, answer open

Mock-up, Design System, Demo videos

Delivered

Validation of the redesign with the client and delivery of the mock-ups and the design system.

Demo videos

4 videos, 1 per journey: onboarding, declaring a parking, creating a parking space, managing the account.

Frames from my 4 demo videos



Onboarding



Declare a parking



Create a parking space



Manage your account

Key result · Delivery Park, Aleia

1 month to deliver the MVP

From the client’s wireframes, as solo designer. MVP in development at handover.

- A graphic charter and an actionable UI kit for developers.
- Redesign validated with the client, delivered as interactive mock-ups with the design system and demo videos.
- The mock-ups and the design system helped the PM (client side) to build his backlog.
- Exchanging with the developer was extremely valuable: thanks to him I was able to build a feasible application.

