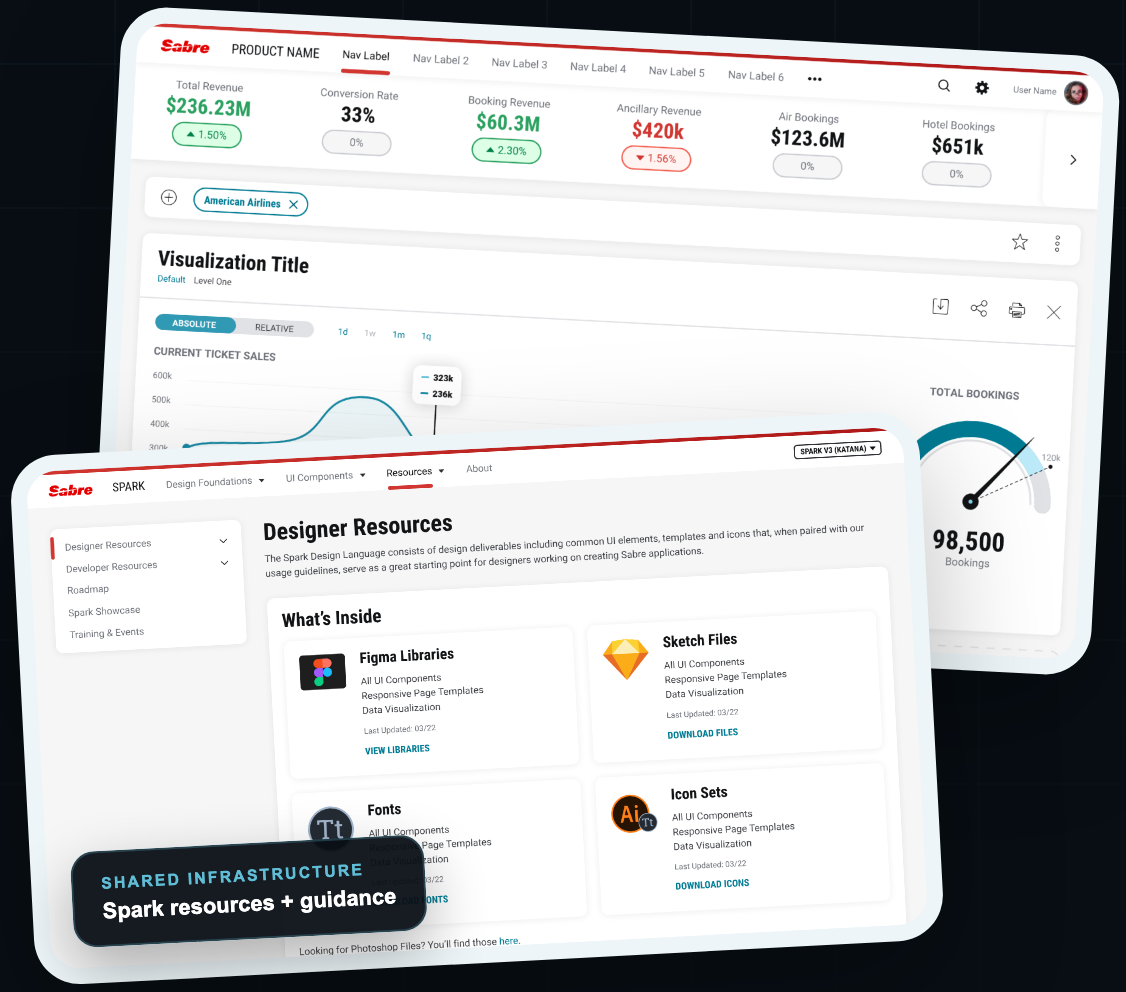
01 SABRE · 2007-2023

I learned the enterprise one product at a time. Then helped build the system that could connect it.

Sixteen years across consumer travel, corporate booking, airline, and hospitality led to a founding role on Sabre Spark - and a design practice built to scale through shared tools, direct support, and trust.



ROLE
Principal Designer
Founding Spark designer

RANGE
Product UX · Systems
Enterprise enablement

LEADERSHIP
Mentored 15+ designers
Cross-team partnership

Different products. Repeated problems. One opportunity to multiply better decisions.

OBSERVE

Product depth

Years inside distinct travel products exposed different users, devices, operating environments, commercial models, and technical realities.

CODIFY

System responsibility

As a founding Spark designer, I helped translate recurring product needs into three generations of shared foundations, patterns, documentation, and resources.

MULTIPLY

Participatory adoption

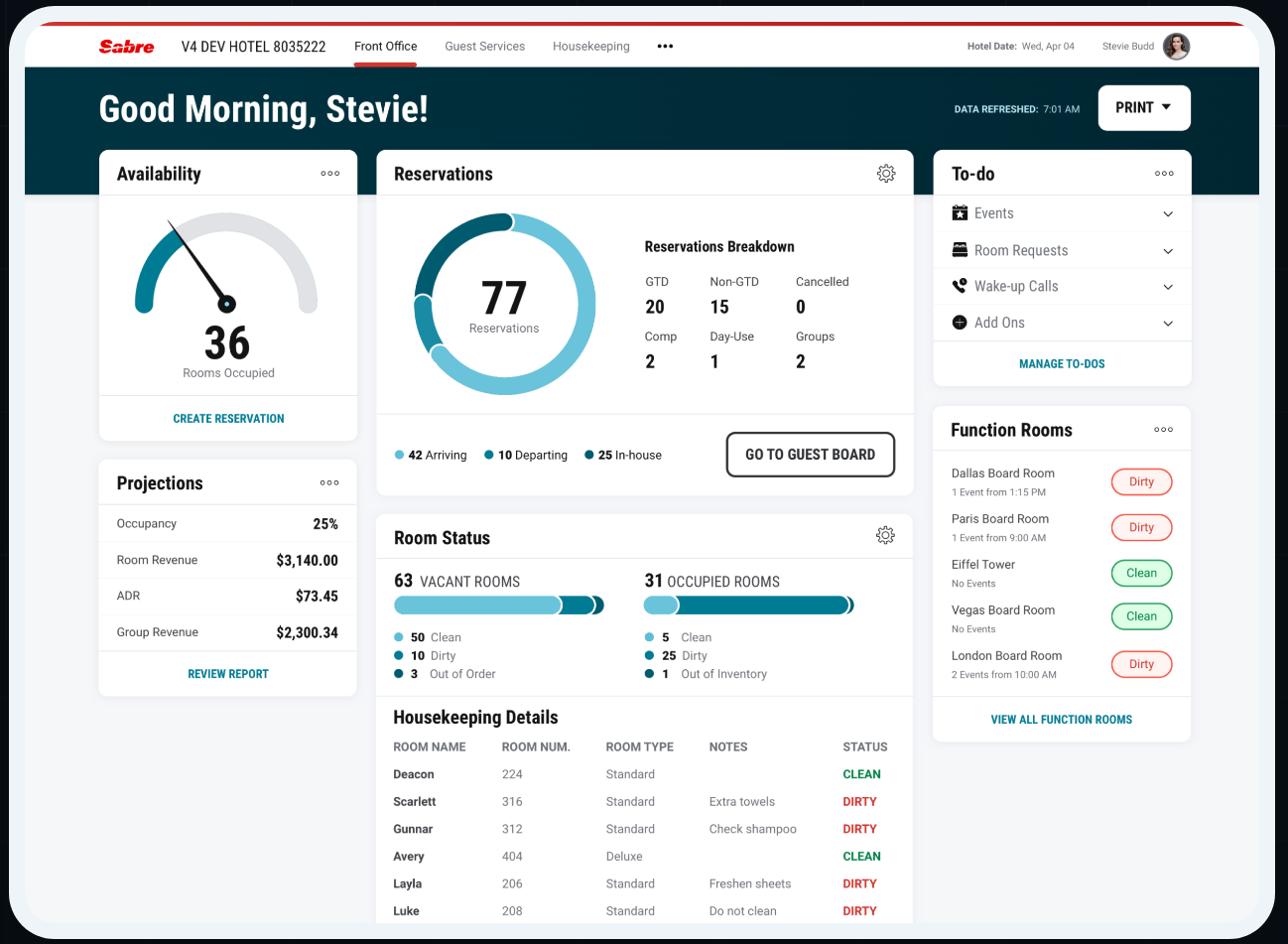
I mentored designers, supported teams directly, and helped create the Spark Showcase so implementation quality became visible, coached, and celebrated.

Three completed Spark generations	**40+ coded UI elements in V3**	**Responsive product guidance**	**Mentored 15+ designers**

03 PRODUCT RANGE BECAME RESEARCH

Before I helped shape the system, I had to understand the realities it would serve.

One enterprise contained distinct users, workflows, devices, operating contexts, and commercial realities. Product depth made the shared opportunity visible.



RESPONSIVE PRODUCT EVIDENCE

Hospitality operations across desktop, tablet, and mobile

Improving one interface at a time could not create an enterprise experience.

Independent product work could improve local experiences while the portfolio continued to diverge. Strong shared decisions had to become available wherever they genuinely applied.

PRODUCT REALITIES

Different work had to stay different.

Booking · Operations · Analytics · Traveler tools · Hospitality

REVEALED

REPEATED CONCERNS

The same questions kept returning.

Hierarchy · Navigation · Status · Data density · Responsiveness · Accessibility

MULTIPLIED

ENTERPRISE COST

Related problems. Uneven outcomes.

Local progress could not compound without shared infrastructure.

Enterprise consistency cannot be imposed abstractly. It has to earn its place inside real product constraints.

We were not building a style guide. We were building shared product judgment.

As one of Spark's founding designers, I worked with another founding designer and a broad network of partners to turn product lessons into a shared design language.

01 FOUNDATIONS
Color · type · spacing · grids · accessibility

INFORMED

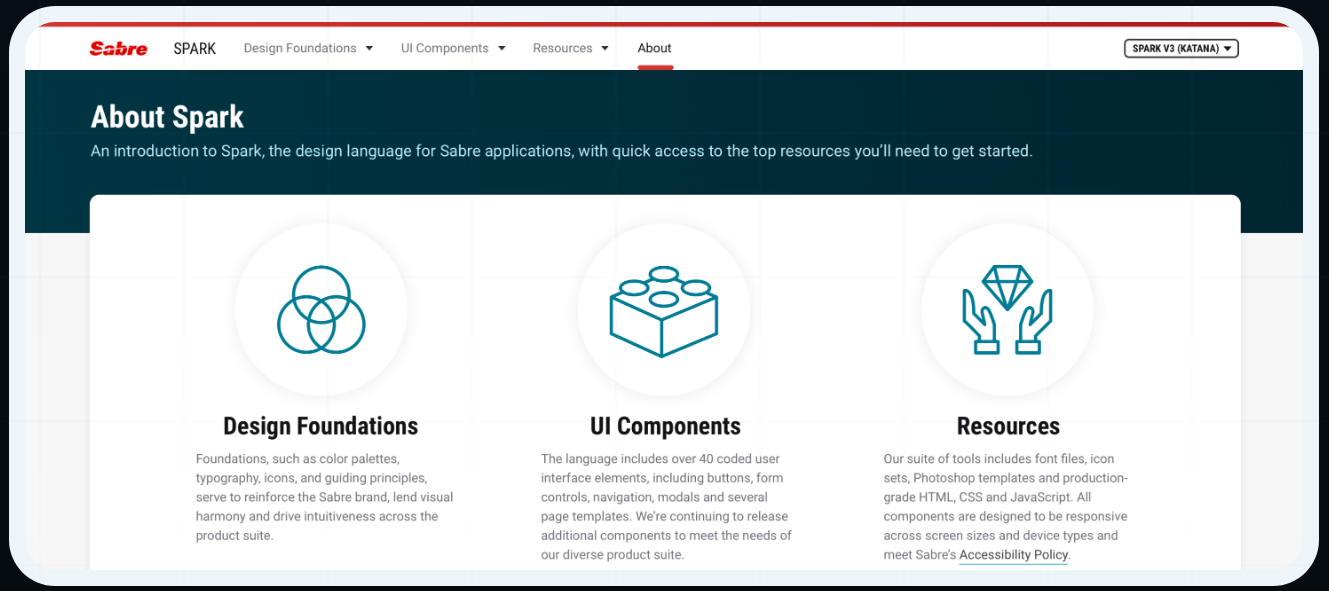
02 INTERACTION LANGUAGE
Navigation · controls · feedback · hierarchy

SHAPED

03 PRODUCT GUIDANCE
Templates · data visualization · operational patterns

SUPPORTED BY

04 ENABLEMENT
Libraries · documentation · training · direct support



CO-CREATED SYSTEM EVIDENCE

Foundations, coded components, resources, and responsive guidance

Each generation expanded what the system had to solve.

Spark grew from a recognizable language into deeper product guidance—and then into a service teams could more easily understand, access, and apply.

GENERATION 1

Establish the language.

Create a coherent foundation teams could recognize and begin using.

RECOGNITION

GENERATION 2

Broaden the system.

Add component depth, responsive behavior, and clearer application.

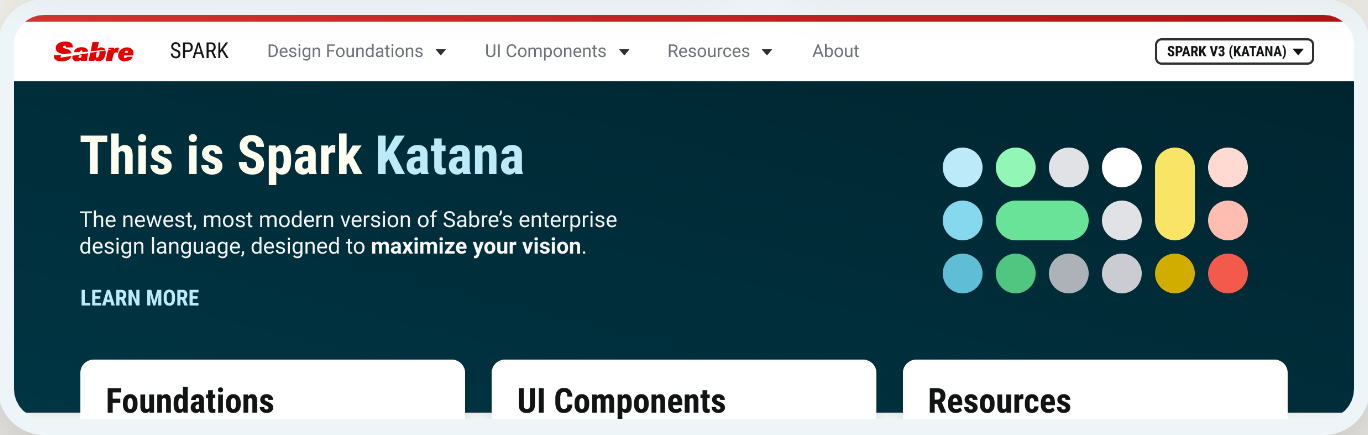
REACH

GENERATION 3 · KATANA

Make the system easier to use.

Strengthen documentation, accessibility evidence, templates, resources, and support.

ENABLEMENT



GENERATION 3 EVIDENCE

Katana made the system—and the service surrounding it—more visible and approachable.

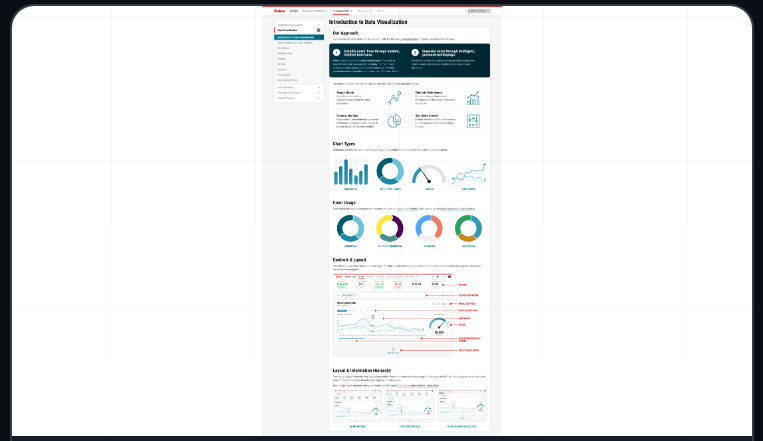
Shared patterns only mattered if they held up inside dense, consequential work.

Spark addressed state, hierarchy, semantic meaning, responsive behavior, product context, and implementation guidance - not appearance alone.

ACCESSIBLE COLOR

Contrast was documented.

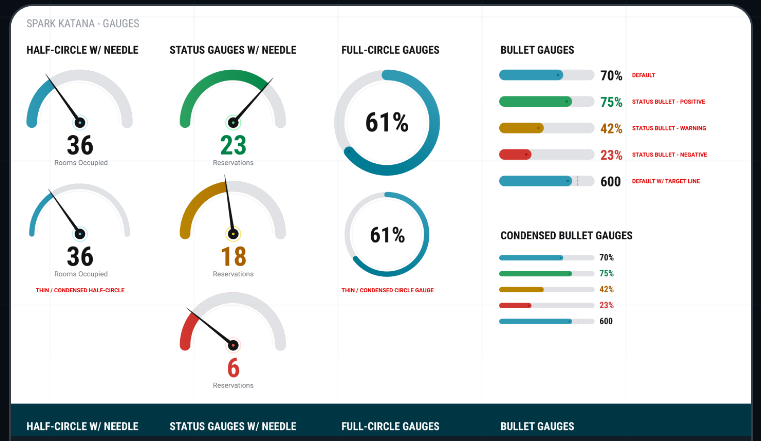
Tonal scales paired exact values with white and black contrast evidence.



DATA VISUALIZATION

Principles met product structure.

Charts, palettes, hierarchy, controls, legends, and layouts worked as one system.



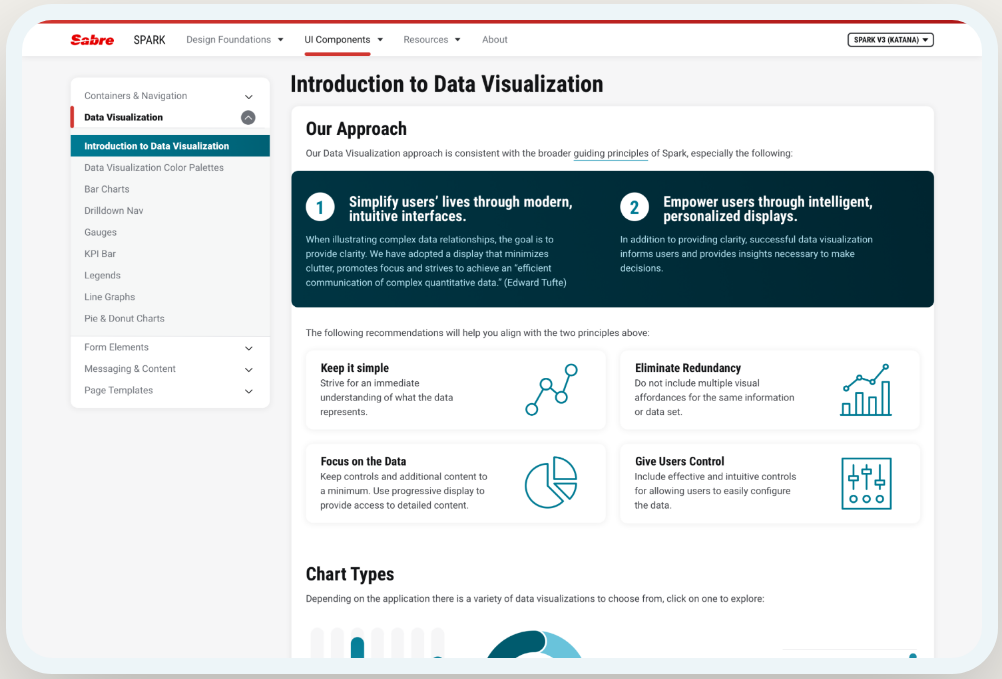
REUSABLE BEHAVIOR

Components carried meaning.

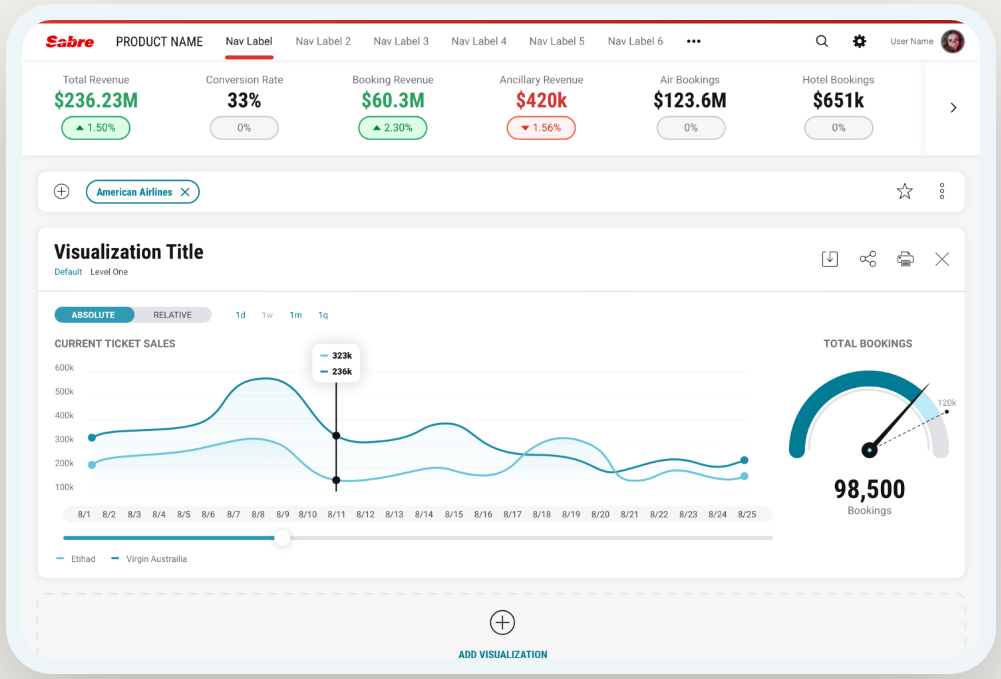
Gauge, KPI, status, and density variants supported operational products.

The system became credible when teams could recognize it inside their own products.

The goal was not visual sameness. Shared structure improved hierarchy, behavior, accessibility, and implementation while preserving product context.



SHARED STRUCTURE
CONTEXTUAL
EXECUTION

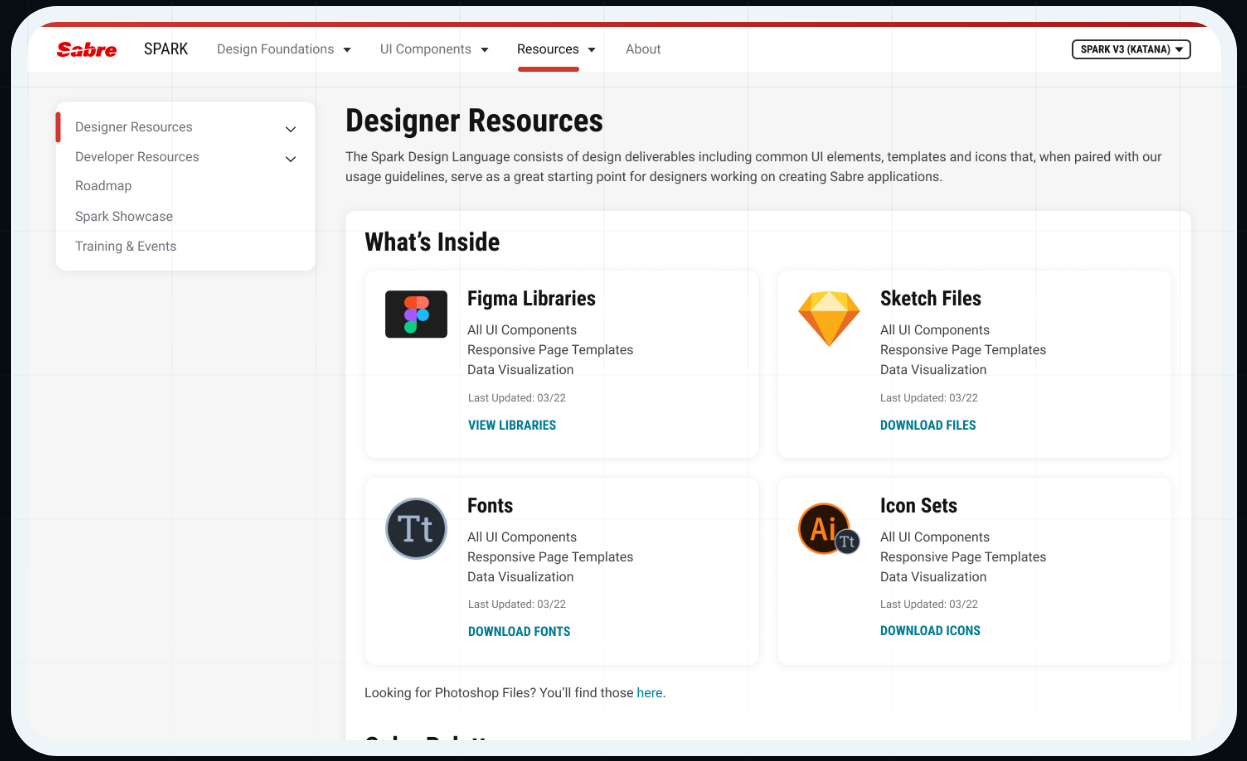


SYSTEM WORK — Shared decision logic

PRODUCT APPLICATION — Implemented by a Sabre product team

Publishing components was only the beginning.

Spark gave teams multiple ways to enter the system, understand it, request change, and apply it. Supporting both Sketch and Figma reduced adoption cost for teams moving at different speeds.



01 LIBRARIES + FILES
Figma · Sketch · fonts · icons · templates

02 GUIDANCE + SUPPORT
Documentation · training · newsletters · direct help

03 OPERATING FEEDBACK
Roadmap · changelog · Jira intake · examples

CO-CREATED ENABLEMENT EVIDENCE

Multiple entry points supported adoption

Evidence boundary: The archive verifies that dual-tool enablement existed. Precise ownership within the Sketch-to-Figma transition is not overstated.

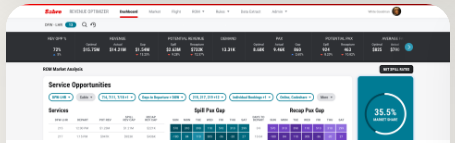
We made adoption visible, participatory, and worth celebrating.

The Spark Showcase gave teams a reason to make the system their own - and made strong implementation visible across the enterprise.

Conceive Shaped the Showcase concept and its adoption role.	**Place** Defined how and where it lived within Spark.	**Recognize** Designed the tiered stickers and rewards.
Evaluate Reviewed and judged team submissions.	**Support** Improved implementations hands-on.	**Mentor** Built trust through critique and partnership.

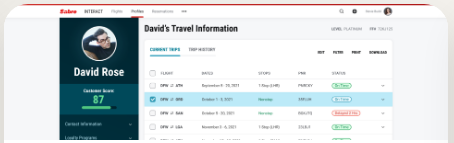
DESIGNED BY TREY

Warrior · Ninja · Master

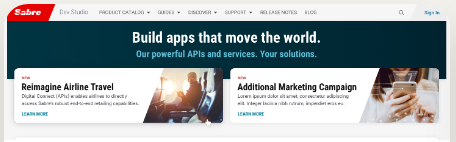


Revenue Optimizer

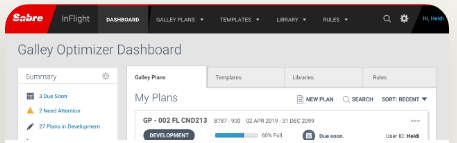
Load Factor



Customer Profile



Developer Studio



Galley Optimizer

My influence grew, but I never stopped working alongside the people applying the system.

I led through a feedback loop: listen to product reality, improve the implementation, then carry the learning back into Spark so the next team started stronger.

01 **Critique + mentorship** Sharpen decisions, not merely conform to a library.	02 **Implementation support** Work with teams when constraints challenged the system.	03 **Submission evaluation** Recognize adoption and make quality tangible.
04 **Pattern development** Turn repeated needs into reusable guidance.	05 **Feedback return** Carry edge cases and learning back into Spark.	06 **Resource creation** Make system judgment easier to apply.

I did not leave product design behind. I helped good product decisions travel farther.

Sixteen years of product depth became a leadership practice grounded in credibility, practical guidance, direct support, and trust.

03	15+	40+	01
Completed Spark generations	Designers mentored	Coded UI elements documented in V3	Participatory Showcase + recognition model

WHAT THE SYSTEM CARRIED

Shared foundations · components · patterns · responsive resources · product guidance

WHAT MADE IT SUSTAINABLE

Implementation support · feedback loops · mentorship · visible participation · trust

I do my best work where product complexity, design quality, and organizational scale meet.

Explore the interactive case study at work.tp3media.com/sabre