UX WRITING CASE STUDY

Designing for the User Nobody Planned For

Writing a two-sided survey experience for a healthcare compliance platform

At a glance	
Product	In-app survey tool within an established healthcare compliance platform
Primary use	Conflict-of-interest disclosure surveys
My role	UI/UX Writer: owned all on-screen copy, end to end
Team	Product managers, product designers, engineering
Timeline	9 months from discovery to initial launch
Outcome	90 customer organizations actively using the tool within one year of launch

Overview

A healthcare compliance platform was adding a native survey product so customer organizations could run conflict-of-interest disclosures and other surveys without leaving the application. I was the UI writer for the project and owned every word on the screen: buttons, labels, tooltips, warnings, confirmations, and system messages.

Early in the project, I noticed that the plan covered only one of the two people who would use the product. Survey administrators were fully mapped out, but the employees who would actually answer the surveys were not. I worked with product to add that second user journey, which shaped the experience we shipped.

The Problem

Customer organizations needed to collect conflict-of-interest disclosures from their staff on a regular basis. Before this project, they relied on third-party survey tools outside the platform. That created real friction for compliance teams:

- **Scattered data.** Responses lived outside the system of record, so they had to be exported, reconciled, and re-imported.
- **Harder audits.** Proving who was surveyed, who responded, and when took manual effort and was prone to error.
- **Extra work.** Admins had to switch between tools and maintain separate distribution lists.

The goal was to bring surveys into the platform so that distributing, tracking, and auditing them happened in one place.

The Gap I Found

The initial product plan focused entirely on the survey administrator: building surveys, sending them out, and reviewing results. But every survey has a second user, the respondent. In a conflict-of-interest survey, that respondent is an employee being asked to disclose potentially sensitive information about outside relationships, finances, or affiliations.

If the respondent experience was confusing or felt intimidating, response rates and data quality would suffer, and the admin experience would fail with it. I raised this with product and took responsibility for defining the respondent persona and journey.

Competitive research

To ground the respondent journey in real patterns, I did extensive competitive research on how other survey tools handled both sides of the experience. I looked at:

- **Entry points.** How respondents are invited, and what they see first
- **Orientation.** Progress indicators, saving drafts, and returning later
- **Trust and privacy.** How tools explain who sees answers and why the information is collected
- **Closure.** What happens after submission, and whether answers can be edited

Based on this research, we created a customizable post-submission experience. Instead of displaying a standard thank-you page, survey administrators could provide a personalized message with relevant information, next steps, or resources.

Two Journeys, Two Voices

With both personas defined, I worked with product and design to plan the on-screen journey for each. The same product needed to speak to two very different people, so I tuned the voice for each while keeping terminology consistent across both. Explore how copy guides user journeys.

	Survey administrator	Survey respondent
Who they are	Compliance professionals who manage surveys regularly	Employees across the organization who may take one survey a year
What they need	Efficiency, control, and accurate tracking	Clarity, reassurance, and a quick path to finish
Voice	Precise, direct, and action-oriented	Plain language, calm, and supportive

	Survey administrator	Survey respondent
Key moments	Building a survey, sending it, monitoring completion, closing it	Opening the invitation, understanding why they are being asked, answering, submitting
Copy priorities	Clear consequences on destructive or irreversible actions	Explaining purpose and privacy up front, and confirming what happens next

Writing Within Constraints

Working within a design system

The survey tool had to use the platform's existing design system so it would feel native to the rest of the application. That meant working with standard components and their built-in limits, including fixed button widths, character limits on labels, and tooltip length restrictions. Some messages had to be rewritten to fit components that were not designed with this content in mind.

Accessibility

We aimed to meet as many WCAG accessibility guidelines as possible. Parts of the existing design system were not fully compliant, so I used copy to close the gaps where I could:

- **Descriptive actions.** Button labels that make sense out of context for screen reader users (for example, "Delete survey" instead of "Delete")
- **Text, not just color.** Error and warning messages that state the problem in words rather than relying on color or icons alone
- **Persistent labels.** Field labels and instructions that stay visible rather than living only in placeholder text
- **Plain language.** Short sentences and common words, especially for respondents

Partnering With Engineering

Once designs were released to engineering, I became the contact for any copy-related issue that came up during the build. When a design could not be implemented as planned because of technical limits, component behavior, or edge cases nobody had anticipated, I adapted the copy, and sometimes the structure, so the experience still made sense to users.

Example: Making room for clear tab labels

The design for survey creation took administrators through eight tabs, each covering one step of building and deploying a survey. The design system technically supported eight tabs. During the build, though, we found that our descriptive tab labels took up too much horizontal space, and only six tabs would fit.

One option was to shorten the labels until all eight fit. I didn't recommend it. Clear, descriptive labels help admins understand where they are in a multistep process, and vague one-word labels would have made the workflow harder to follow, especially for screen reader users.

Instead, I looked at how admins would actually use each tab. Two of them, Settings and Permissions, were typically set up once. Organizations usually reuse the same settings and permissions when they create each year's new version of a survey, so admins wouldn't need to come back to these steps often.

I brought a recommendation to product and design: replace those two tabs with buttons that open a modal where admins can set the conditions. The recommendation was accepted. The change:

- Kept the full, descriptive labels on the six remaining tabs
- Kept the tab bar within the design system's limits
- Put the most frequent tasks front and center, while keeping set-once options one click away

This showed me that a UX writer can sometimes solve a space problem without cutting words. Understanding how people actually use a feature can point to a structural fix that keeps the language clear.

Being available after hand-off kept the copy accurate to what actually shipped, not just to what was originally designed.

Writing Samples

The examples below are recreated and anonymized to respect confidentiality. They show the reasoning behind key copy decisions.

Annotated microcopy

Element	Copy	Why it works
Destructive confirmation (admin)	“Close this survey? Respondents won’t be able to submit answers after it’s closed.” Buttons: “Close survey” / “Keep open”	States the consequence clearly. Button labels name the action instead of “Yes/No.”

Element	Copy	Why it works
Tooltip (admin)	“Changing the order of questions may affect survey rules that show, hide, or skip questions. Review and test these rules after reordering questions.”	Warns admins about a hidden consequence before it happens and tells them exactly what to do next.
Survey landing page (respondent)	Shows the survey name, past versions, due date, and an option to reuse previous responses. “Reuse previous responses” button was clearly displayed for users that completed a previous version.	Gives respondents what they need to know before they start and saves time on repeat surveys.
Question completion checkmarks (respondent)	A checkmark appears next to each completed question.	Helps respondents who return later see what's left at a glance. For respondents using assistive devices, ‘completed’ screen reader text was added.

Before and after

Draft	Final	What changed
One date field: **End date**. The survey closed automatically on this date and stopped accepting responses.	Two separate date fields: **Due date:** The date admins ask respondents to finish the survey by. The survey stays open after this date. Helper text “Proposed deadline for respondents. Allows responses past the date.” **End date:** The date the survey closes automatically and stops accepting responses. Helper text: Survey closes. No more responses past this date.	In early user testing, admins said they expected some respondents to miss the deadline. A single date forced them to choose between closing the survey on time and collecting every response. Separating the two dates let admins set a target date for respondents and still accept late submissions.

Terminology decisions

Consistent terms across both journeys kept the product predictable. I documented decisions like these:

Use	Instead of	Reason
Survey	Questionnaire, Form	Users already knew and used “survey.” “Form” and “questionnaire” referred to other features across the platform.
Submit	Send, Finish	Signals a final action with a clear outcome.
Respondent	Participant, User	“User” already referred to platform customers, and “participant” has a specific meaning in healthcare. “Respondent” clearly identifies the person completing a survey.

Any wording decisions were added to our internal style guides, so these decisions could be implemented across the platform.

Results

- **A native survey tool.** Customers could run surveys inside the platform instead of relying on outside tools.
- **Better auditing.** Keeping responses in the system of record made tracking and auditing easier and more accurate.
- **Strong adoption.** 90 customer organizations were actively using the tool within one year of launch.
- **A complete experience.** Adding the respondent journey meant the product served both sides of every survey, not just the people creating and sending them.
- **Time saved and trust gained.** Admins report that keeping survey results in their compliance platform saves time. Respondents say they feel more secure answering surveys that come from inside their own organization's system.

What I Learned

- Look for the missing user. Mapping every person who touches a feature can reveal gaps before they reach production.
- Words can carry accessibility when components can’t. Clear, descriptive copy can partly make up for design system limits.
- A writer’s job doesn’t end at hand-off. Staying involved during engineering kept the shipped experience coherent.