Manually Add/Remove Tag — Design Spec

> Generated with the `design-spec` Claude Code skill. This is a **design-side** starting point meant to give engineering a head start on the real spec — it is not exhaustive, especially around edge cases. Treat it as the first draft in a conversation with engineering, not the final word.

> *Sample output — illustrates what a completed spec looks like, built for a "Patient Conversations" case study feature.*

Status: Ready for engineering review **Last updated:** 2026-09-14

Progress

Stage	Status
1. Overview & links	Done
2. Components & patterns	Done
3. User flow	Done
4. Behaviors & states	Done
5. Content	Done
6. Known edge cases	Done

---

1. Overview & links

Summary: Care team members can manually add or remove tags (e.g. "Urgent," "Follow-up," "Billing Question") on a patient conversation thread to help triage and organize their inbox.

Problem/goal: Care coordinators currently rely on memory and scrolling to find conversations that need follow-up. Tags let them mark and later filter conversations without leaving the thread.

PRD: Patient Conversations — Tagging PRD

Figma link	Covers
Conversation thread — tagging flow	Main add/remove flow
Tag chip — states & variants	Component states

Figma link	Covers
Empty & error states	Edge-state frames

2. Components & patterns

Tag chip *(new)* — a small rounded label shown in the conversation header. - Variants: default (colored by tag category), hover (shows a remove "×"), overflow (`+2` when more tags exist than fit the row) - Color is assigned per tag category, not per-conversation, so the same tag reads consistently across threads

Add Tag control *(new)* — a combobox opened from a "+ Add tag" button in the thread header. - Typing filters existing tags; if no match, shows a "Create '{{query}}'" option - Closes on selection, on Escape, or on click-away

Everything else in this flow (thread header layout, buttons, toast component) reuses existing components unchanged.

3. User flow

Entry point: Conversation thread header, via the "+ Add tag" button (always visible, even with zero tags).

Add flow: 1. Care team member clicks "+ Add tag" 2. Combobox opens with existing tags listed, most-recently-used first 3. They select an existing tag, or type a new name and select "Create" 4. Tag chip appears in the header immediately; combobox stays open for adding another, or closes on click-away

Remove flow: 1. Care team member hovers an existing tag chip 2. An "×" appears on the right edge of the chip 3. Clicking it removes the tag immediately (no confirmation dialog — see Stage 6 for undo handling)

Exit points: Combobox closes (Escape/click-away) with 0+ tags applied; there's no "save" step — every add/remove is applied instantly.

4. Behaviors & states

- **Loading:** tag list in the combobox shows a skeleton row while existing tags fetch (typically <300ms)
- **Empty:** if the workspace has no tags created yet, the combobox shows "No tags yet — type to create one" instead of an empty list
- **Error (add/remove fails):** chip briefly shakes and reverts, with an inline toast: "Couldn't update tag — try again"
- **Max tags:** after 5 tags on one conversation, "+ Add tag" becomes disabled with a tooltip: "Max 5 tags per conversation"
- **Validation:** tag names are case-insensitive for duplicate matching (typing "Urgent" when "urgent" exists selects the existing tag rather than creating a duplicate); max 24 characters

5. Content

- Button: **"+ Add tag"**
- Combobox placeholder: **"Search or create a tag..."**
- Create option: **"Create '{{query}}'"**
- Error toast: **"Couldn't update tag — try again"**
- Max-tags tooltip: **"Max 5 tags per conversation"**

All copy above is final, pulled directly from the Figma file — no placeholder text remaining.

6. Known edge cases

Proposed by the skill from the flow above, confirmed with the designer:

- **Undo removal:** removing a tag is instant with no confirmation — should there be a brief "Tag removed · Undo" toast? *(Confirmed: yes, 5-second undo toast, matches the pattern used elsewhere in the app.)*
- **Tag deleted at the workspace level while applied to a conversation:** needs a decision on whether it silently disappears from threads or the thread keeps a "ghost" reference. *(Not yet resolved — flagged for engineering.)*
- **Two care team members editing tags on the same conversation simultaneously:** last-write-wins is assumed but not explicitly designed.
- **Very long tag names on narrow viewports:** truncation behavior for the chip isn't specified in Figma.

> Note: **Not exhaustive.** This section captures edge cases already identified during design. The remaining edge cases are expected to surface and get resolved jointly with engineering while writing the final implementation spec — that's a feature of this process, not a gap in it.