

## Story UFC-329: Feedback Form

The following document will go over the design of the feedback form, as outlined in Story ticket UFC-329.

### Acceptance Criteria

For reference, the following are the acceptance criteria for this ticket:

- I want to be able to quickly provide feedback about an issue, idea, or other about the app.
- I want to be able to anonymously provide the feedback if I don't want to be contacted about the feedback.
- I should be informed that submitting feedback non-anonymously means that I might get contacted about the feedback.
- I don't (yet) need to be able to manage (i.e. read, update, or delete) feedback that I've previously submitted.
- I (the developer) want to receive the feedback in a Slack channel so that I can quickly action it.

### Design Brainstorming

#### Backend

In terms of Backend, we're gonna need a new table for the feedback. In terms of schema, I've already thought of the following:

- id
- userId
- type
- message

type would be (currently) one of "issue", "idea", or "other".

userId would be nullable. This way, "anonymous" feedback would just leave out the userId.

And then message... should be straight forward enough.

In terms of the Feathers service, it should only have `create` enabled externally, and `create` should require authentication. Every other method (especially the fetching methods) should be disabled externally, for obvious reasons.

In terms of the Slack notifications, should be able to just slap a hook on top of `create` and we should be good. I thought about separating the Slack channels used for the messages (so that sign up messages and the like go to one channel, while feedback goes to another), but meh. We can just keep everything in one channel for now.

Should probably put the aggressive rate limiter on the feedback service, to prevent people from massively spamming us.

As for... anything else? I don't know. Should be a fairly straightforward Backend implementation.

## Frontend

### Front-of-the-Front (UI)

I quite like the design of [dulajkavinda/minimal-feedback](#).

UI design has already been (basically) completed at this point. Will be the most 'complex' modal we have so far, but still really straightforward. Just a two-step form.

The mobile design for the form itself will be the same as on desktop (maybe with reduced padding/spacing), but the way to activate the form will be different.

- On desktop, I was thinking we could put a "Feedback" button just floating in the bottom right corner (as oh so many other apps do), but now I'm just thinking that we plop it in the User dropdown. That way, it's consistent with where we put it on mobile.
- Speaking on mobile, the feedback 'button' would just be another item in the Settings menu.
- In terms of menu item ordering, I'm thinking "Send Feedback" should be above "Check for Updates".

### Back-of-the-Front

Since we don't need to store the feedback on the Frontend, this should be a bit simpler than our usual "new resource" tasks.

We'll definitely still need an offline request slice to handle the creation request, but there just won't be a commit phase (or rollback, for that matter).

- Actually, we *shouldn't* use an offline request slice. Just a regular request slice will be enough. Why? Because we want to surface errors immediately.

Shouldn't be much more to it than that.

## Component Breakdown

### Atoms

- DialogContainer

### Molecules

- FeedbackOption
- FeedbackOptions
- FeedbackForm
- [modification] ConfirmationDialog (use DialogContainer)
- [modification] UserDropdown (add "Send Feedback" item)

### Organisms

- FeedbackDialog

### Scenes

- N/A

### Tasks

- Build the Backend.
- Build the Frontend.