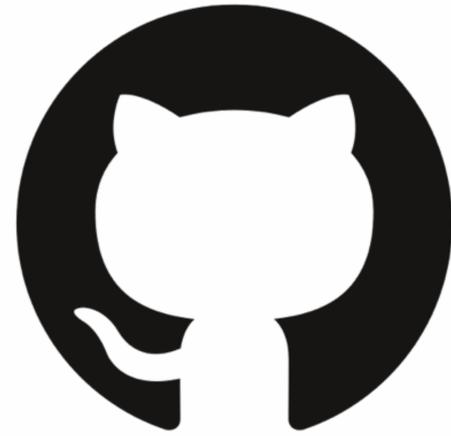


GitHub PR Feature Design Project

Michelle Venetucci

michelle.venetucci@gmail.com



Product Choice: GitHub

Improving Developer <> Designer Workflows

GitHub has the potential to improve cross-functional collaboration throughout the product development process.

Product Opportunity & Research

Conduct basic foundational user research and competitive analysis to provide context and opportunities.

Existing GitHub Product Audit

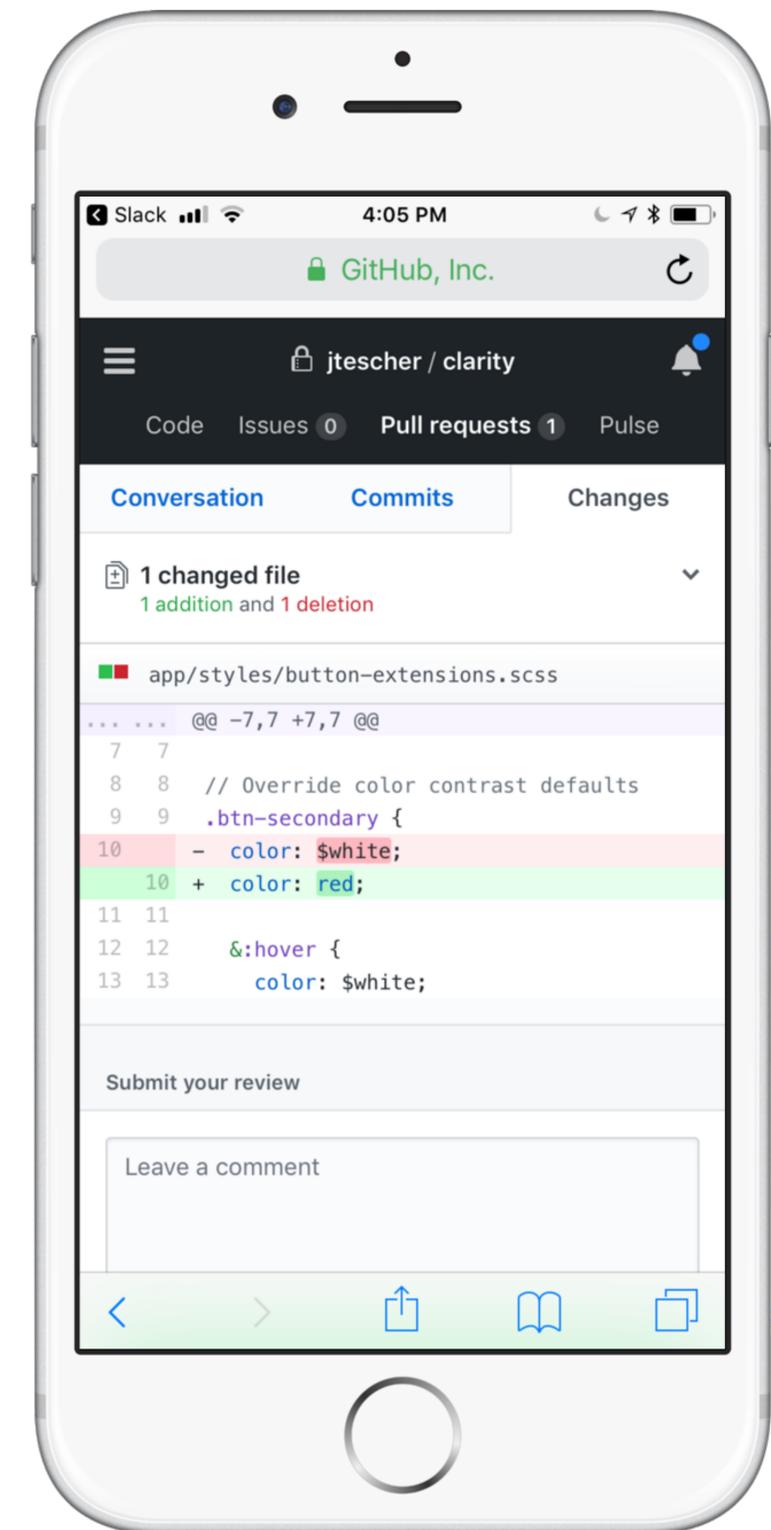
Audit the product to contextualize problems and identify areas of opportunity or improvement.

Brainstorming & Prototyping

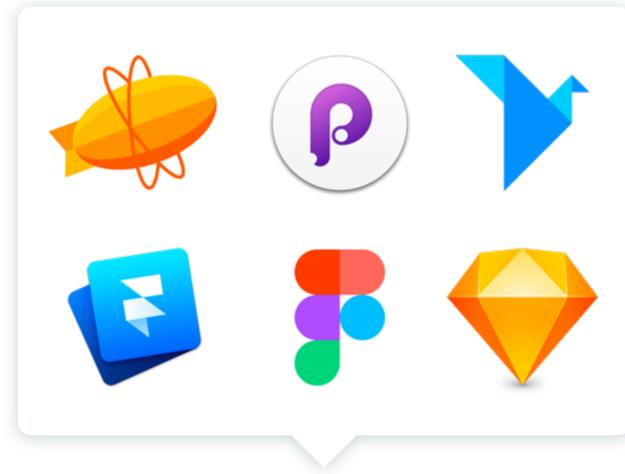
Generate ideas and direction through rapid brainstorming activities, prototyping, and wireframing.

Final Design

Walk through the final solution from a user experience design perspective.



GitHub's existing mobile experience



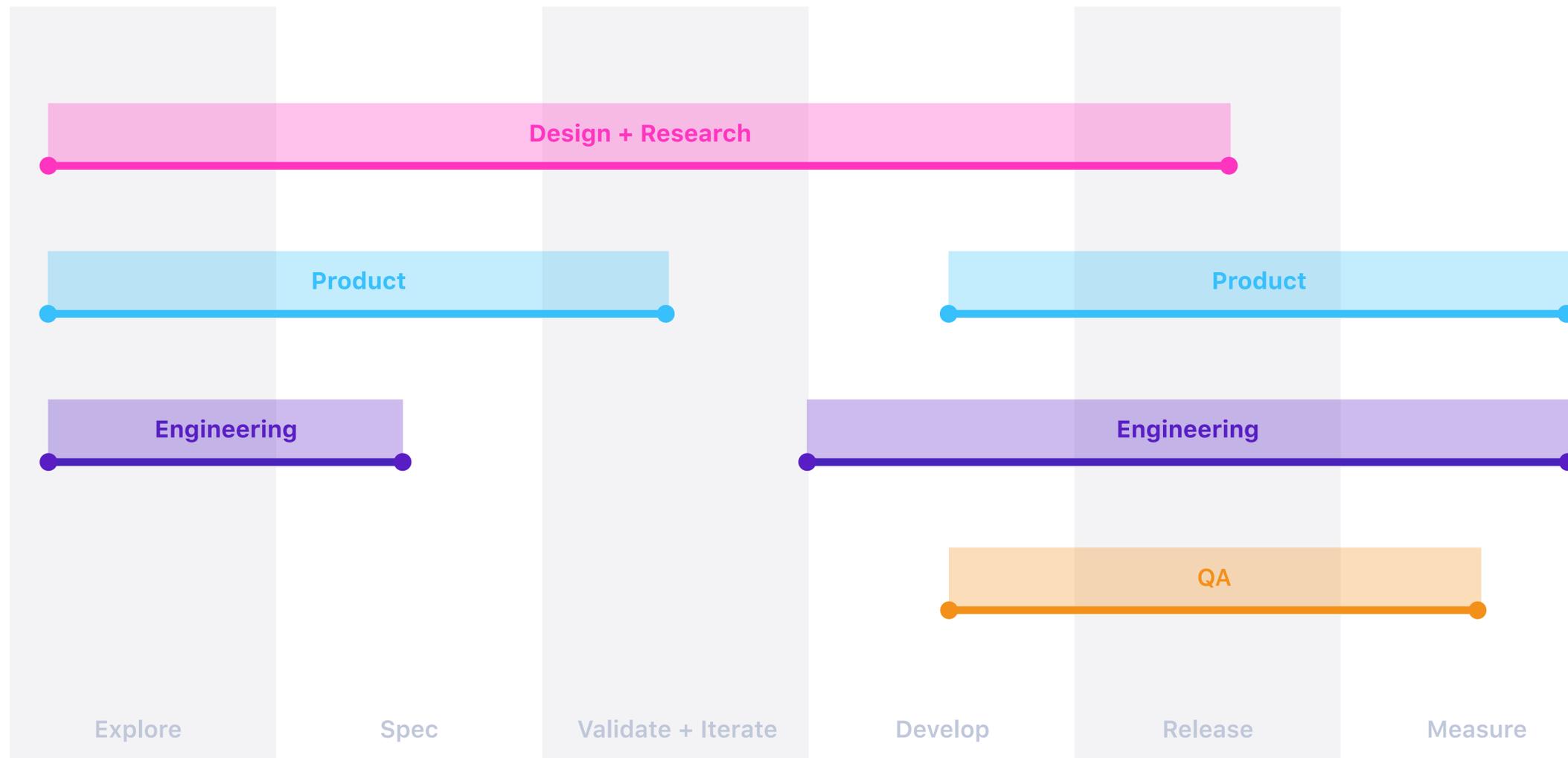
Product Opportunity

Research & Competitive Analysis

I conducted time-constrained foundational user research and looked over the competitive landscape in order to better understand the industry and start identifying the most compelling problem spaces.

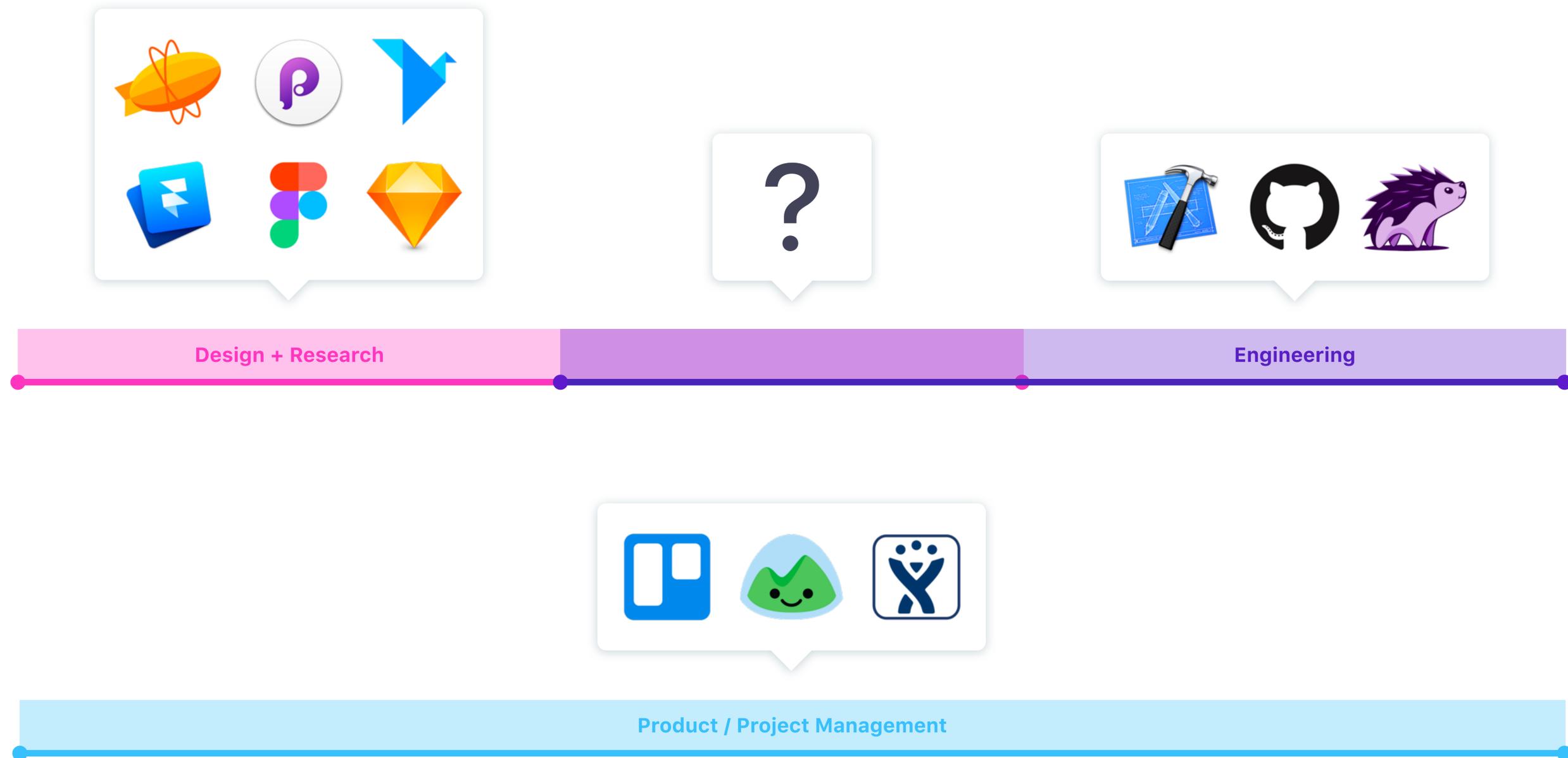
Product Opportunity / Project Lifecycle

There are many cross-functional touchpoints throughout a typical digital project lifecycle.



Product Opportunity / Project Lifecycle

While there are currently many tools that project participants use, there aren't many that are targeted specifically towards cross-functional collaboration and treat cross-functional partners equally.





“

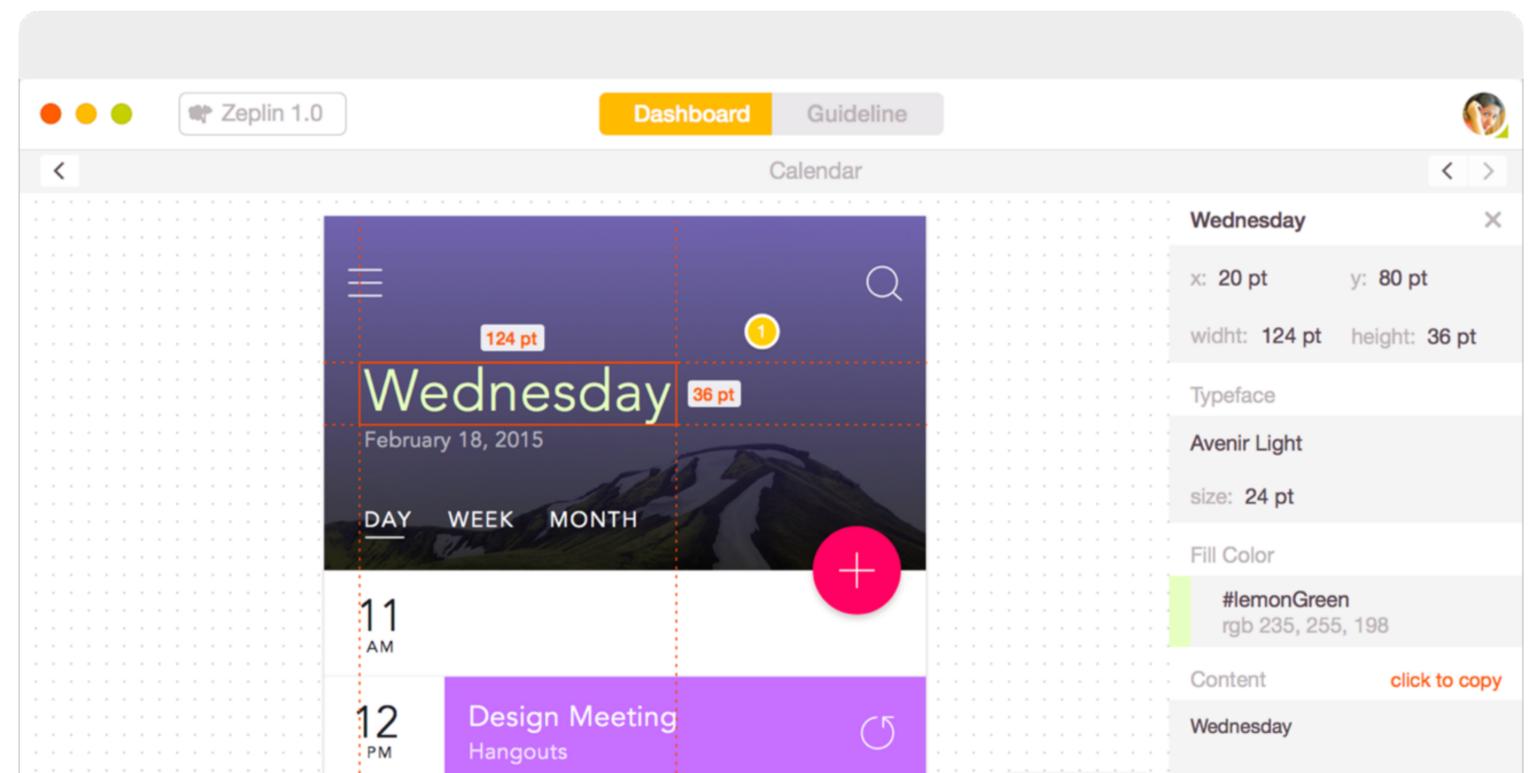
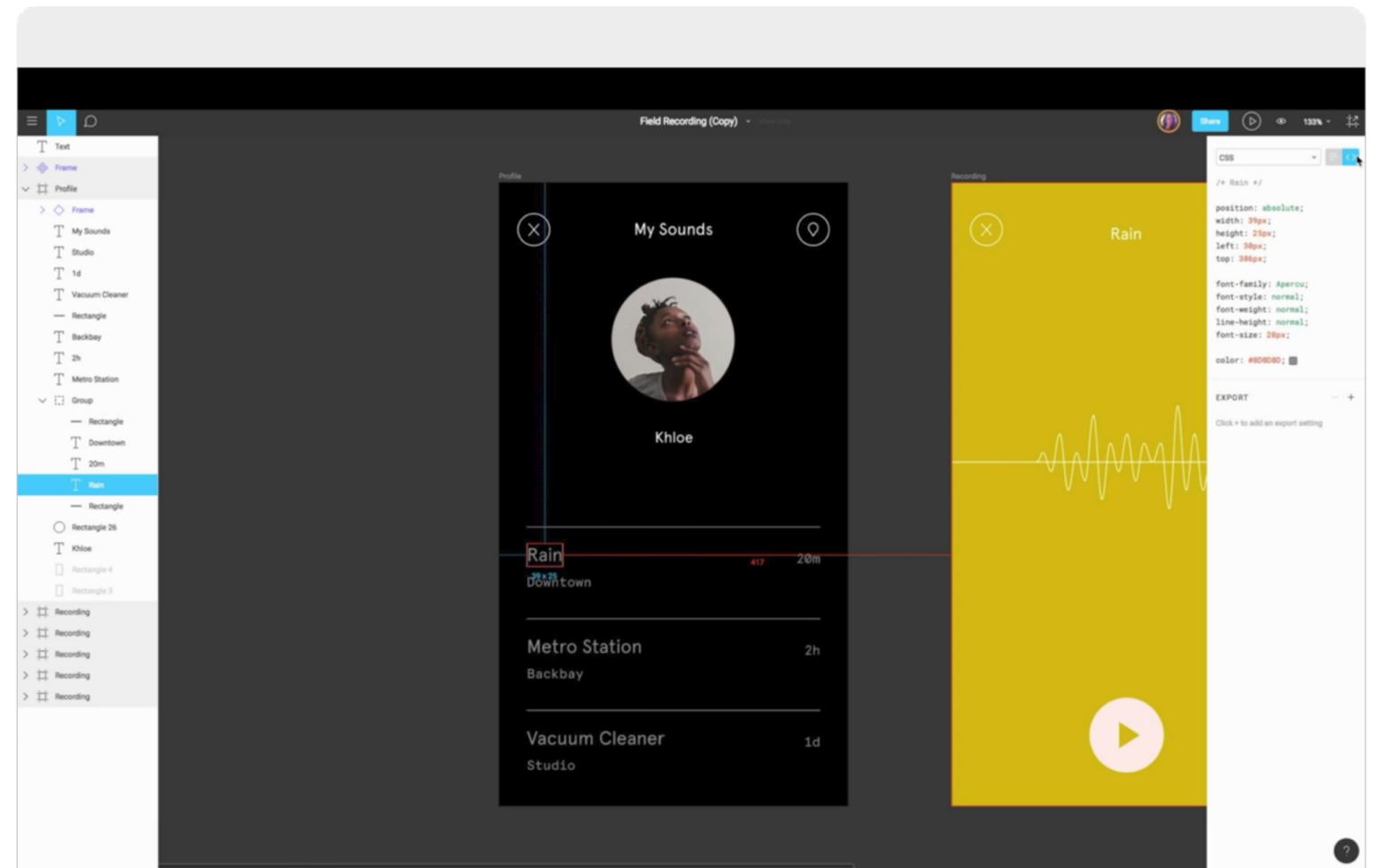
When handing off work to engineers, I always think the videos and mocks communicate the design perfectly. But what I get back often looks nothing like my intention, which ends up being an awful experience.

- Harrison, Product Designer

Competitive Analysis / Design Tools

Many design tools have popped up recently with a main goal of improving communication between designers and developers.

These are generally seen as helpful, but remain somewhat separated from the engineering workflow, which can cause communication breakdowns.





“

I Use Zeplin for redlines, but it's not very efficient. I also use Dropbox, and it's okay. It gets the job done for sharing files, but it's hard to keep track of things. Google docs and Paper could be useful for comments but overall things are hard to track.

- Lin, Product Designer

Competitive Analysis / Visual Regression Testing Tools

There are a host of visual regression testing tools made for engineering workflows.

Most compare proposed CSS to previous CSS visually, and focus on engineers as the end user.

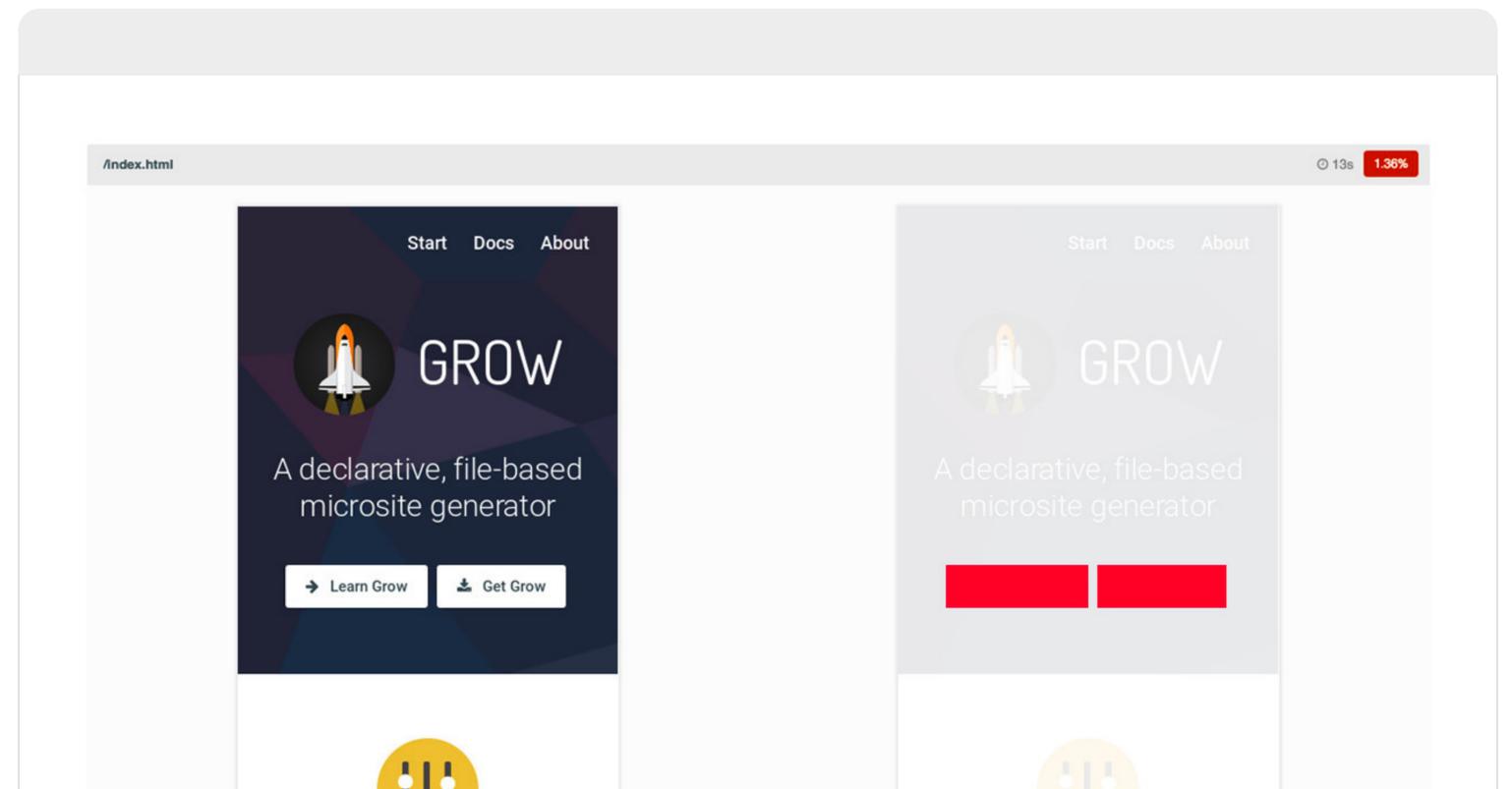
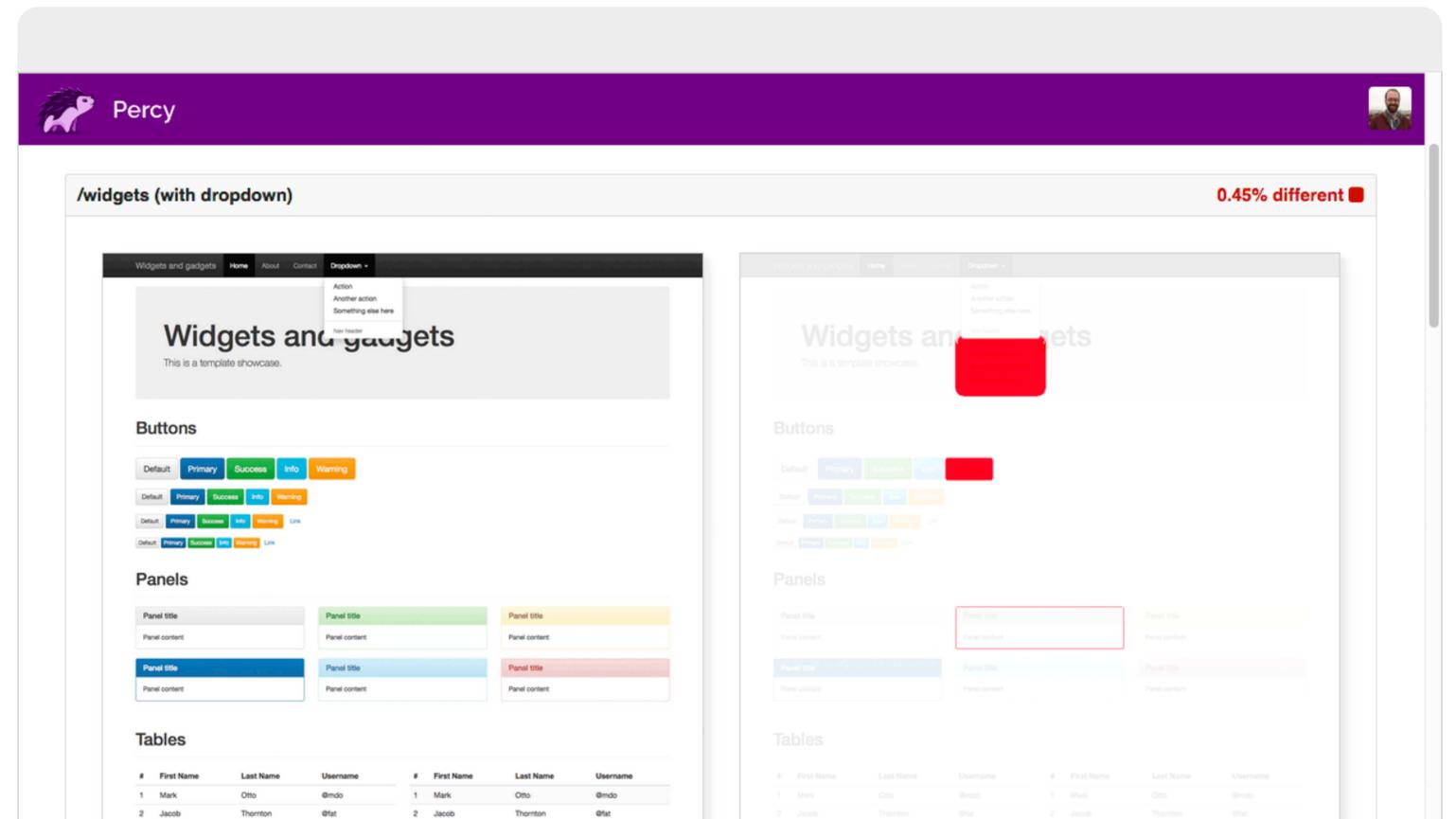
The screenshot displays the Spectre visual regression testing tool interface. At the top, the Spectre logo is visible. Below it, the breadcrumb navigation shows 'Projects > Project B > Components > #4'. The main heading is 'Run #4 (Components)'. A filter bar includes a search input, a 'Browser' dropdown, a 'Size' dropdown, and an 'All' dropdown. The interface is organized into a table with five columns: 'Test name', 'Baseline', 'Comparison', 'Diff', and 'Result'. Two test entries are shown. The first entry is for 'Timetable' at 'Firefox, 1200px'. It shows three side-by-side screenshots: the baseline, the comparison, and the difference (Diff). The 'Result' column for this entry shows a '0.4% difference', a red 'Fail' button, and a 'Set as baseline' button. The second entry is for 'Timetable' at 'Firefox, 768px'. It also shows three side-by-side screenshots and a '0.42% difference' with a red 'Fail' button and a 'Set as baseline' button.

Test name	Baseline	Comparison	Diff	Result
Timetable Firefox, 1200px				0.4% difference Fail Set as baseline
Timetable Firefox, 768px				0.42% difference Fail Set as baseline

One example is percy.io, a continuous visual review tool for engineers building web apps.

“Percy integrates directly into your development workflow to provide iterative and fast feedback about visual changes. This is what we mean by automated visual reviews.”

The fact that so many of these tools exist shows a growing need from users within this problem space, but fragmentation leads to low adoption.





“

There's always a lot of miscommunication in the designer / developer handoff period. It's really challenging to keep track across all the different tools we're trying to piece together.

- Alex, Engineer

Keeping track of updates

Designers and engineers often struggle to keep track of cross-functional work.

Polish

Zeplin has improved polish for static mocks, but it's still challenging to communicate animation values.

Communication is a barrier

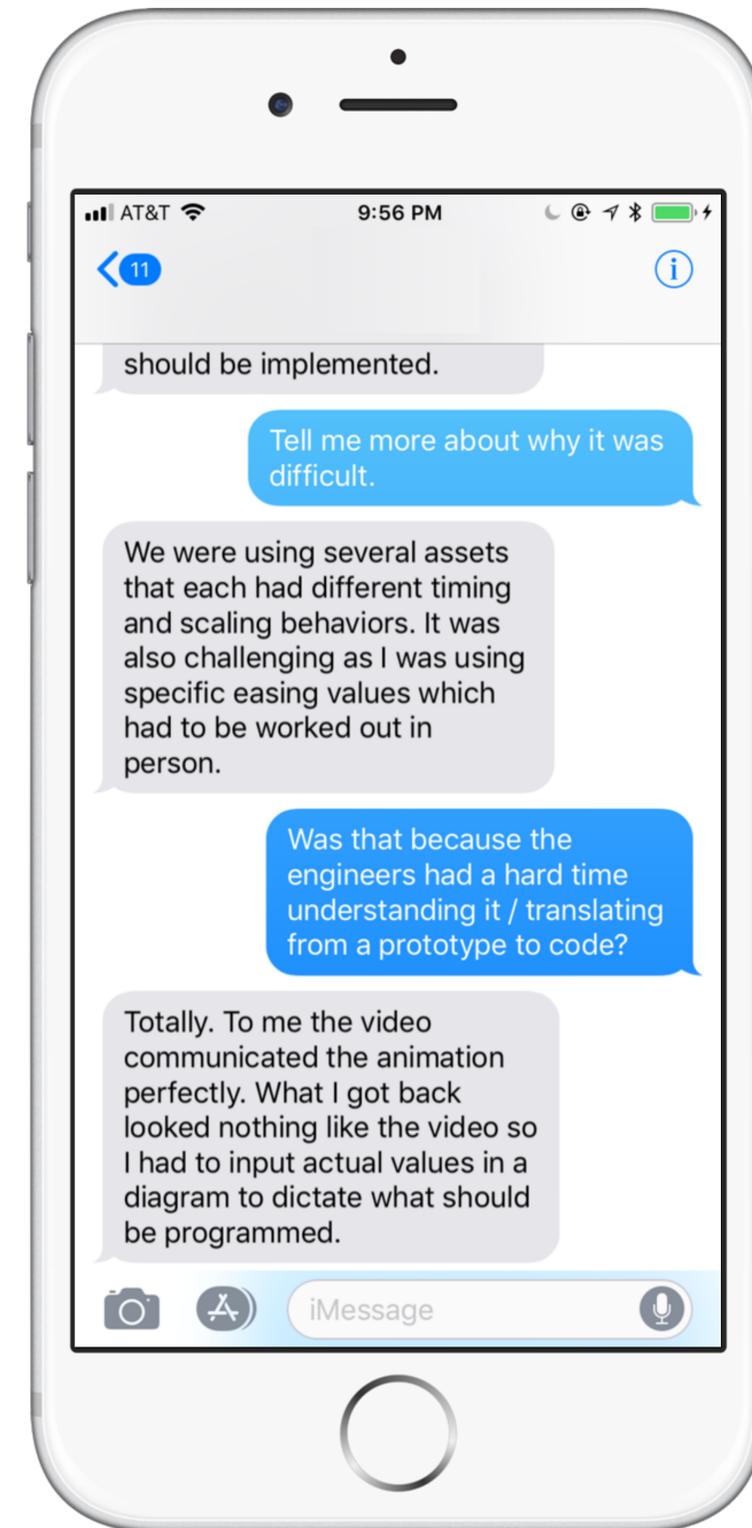
Designers and engineers feel like they communicate clearly to their cross-functional partners, but often the results don't reflect the original intention.

Too many tools

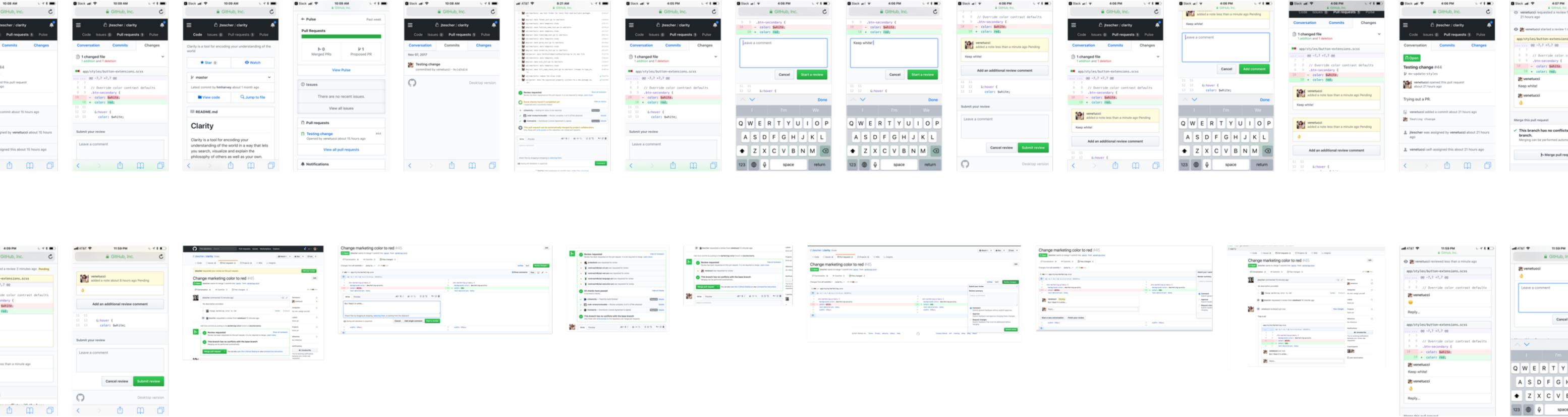
Designers and engineers are using too many tools to communicate with each other.

GitHub as a collaboration tool

GitHub feels very intimidating to everyone except engineers.



Screenshot of SMS-based user interview



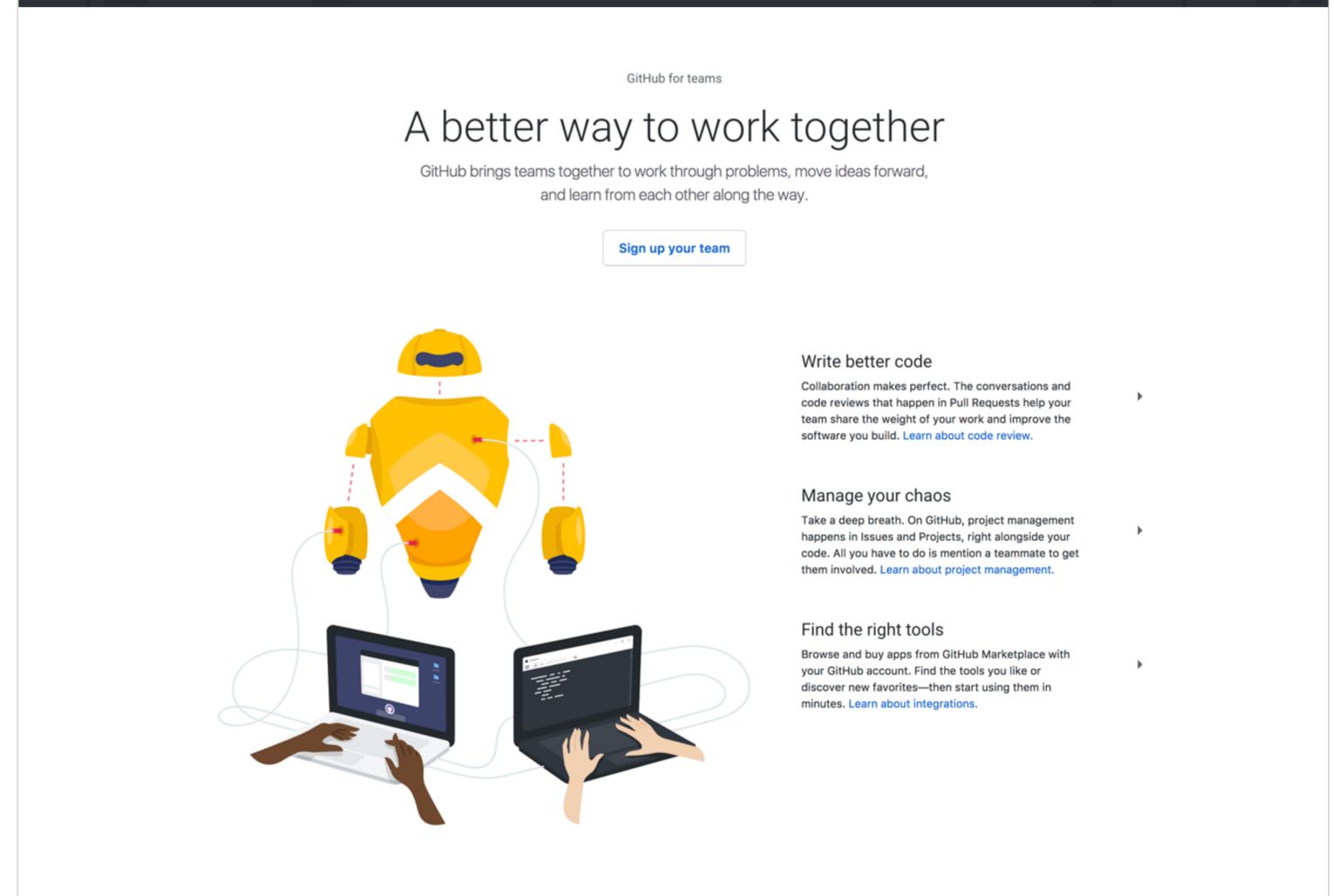
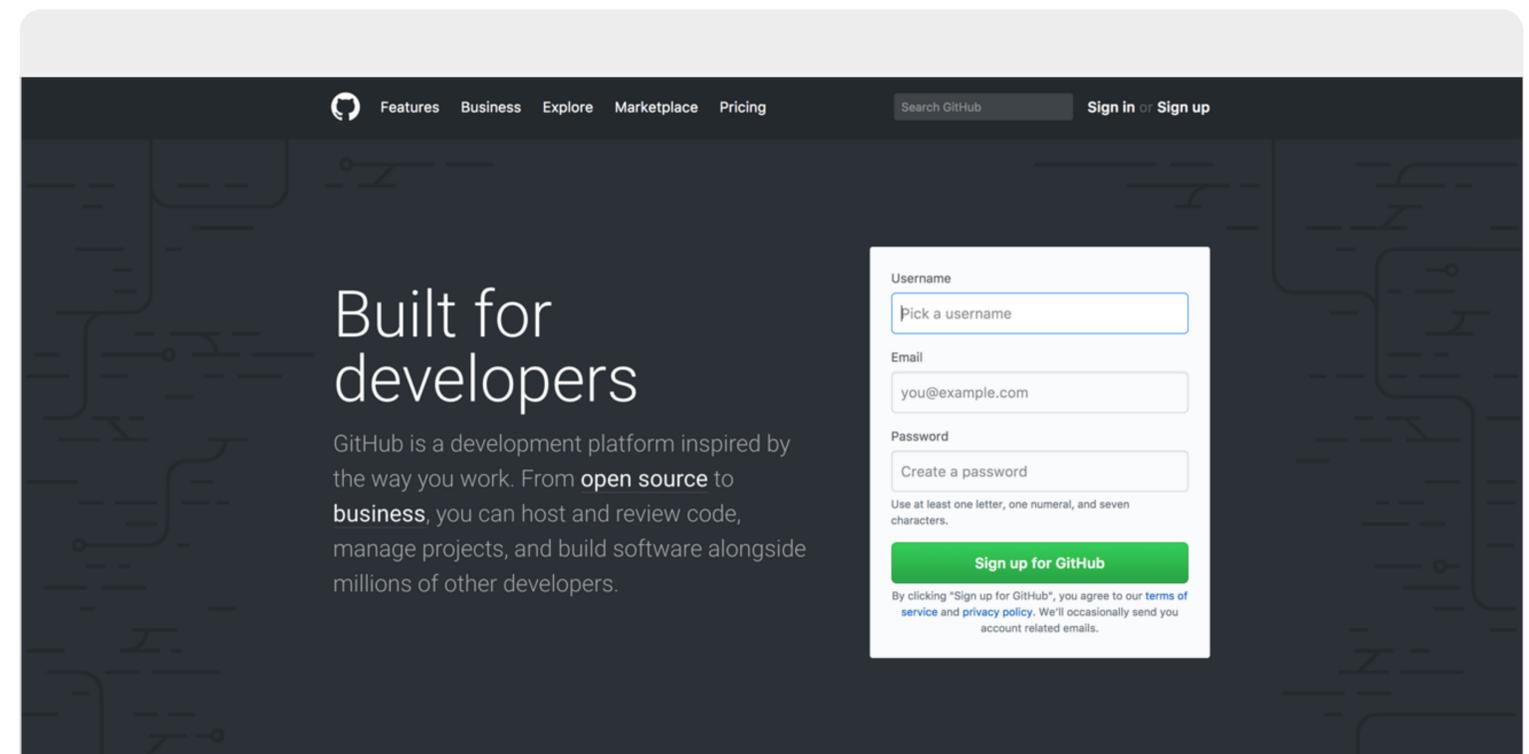
Product Audit

GitHub's Existing Experience

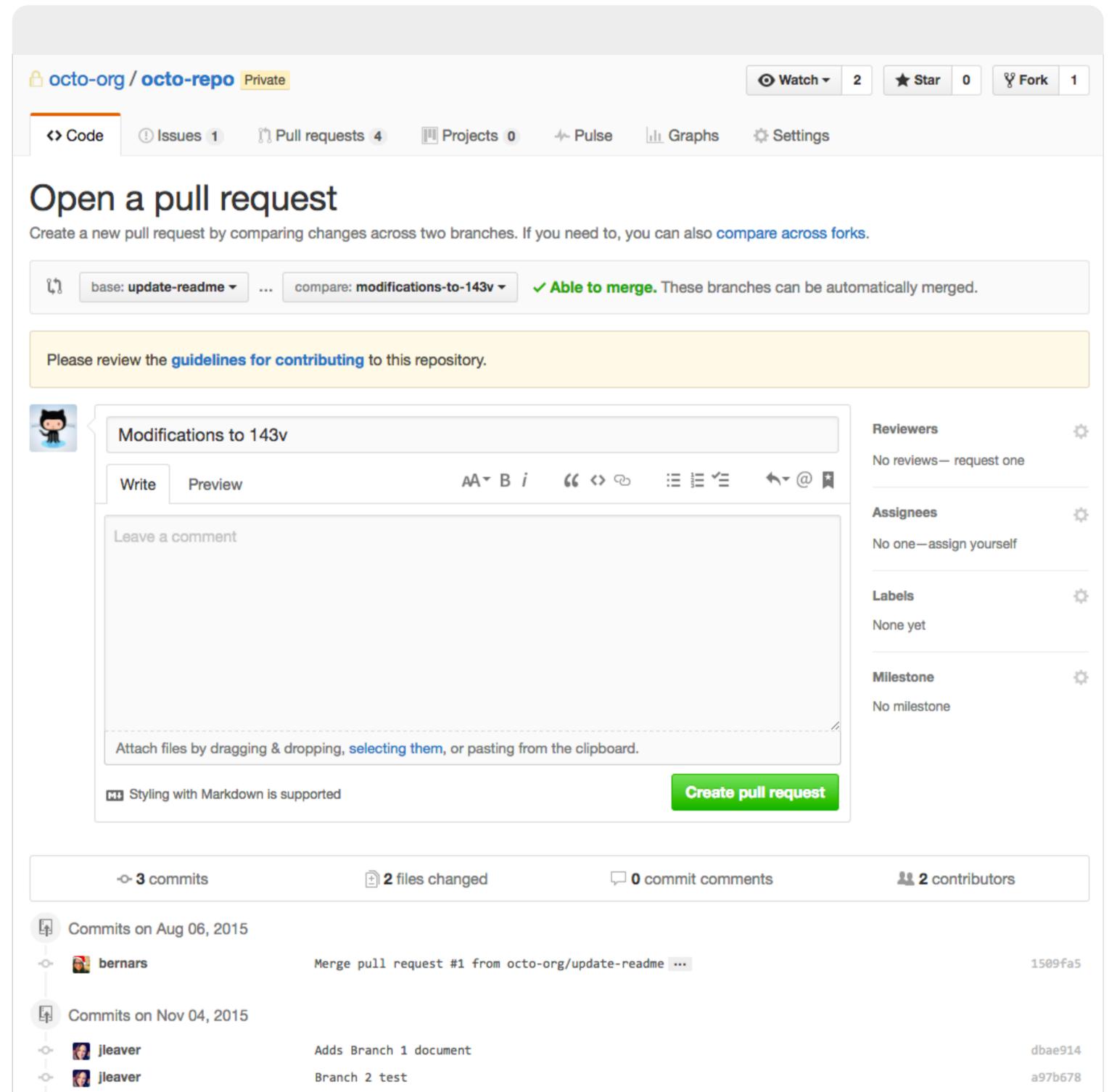
Looking over the existing desktop and mobile user experience helped give me context into existing problem spaces and opportunities for strategic improvement.

GitHub is a great tool for developer collaboration, but doesn't currently support much cross-functional collaboration. Some examples:

- Pull Requests (PRs) only include code reviews, even when there are visual updates that need reviewing.
- The UI is intimidating and there is no user experience geared towards designers, who may want to be involved in the review process.

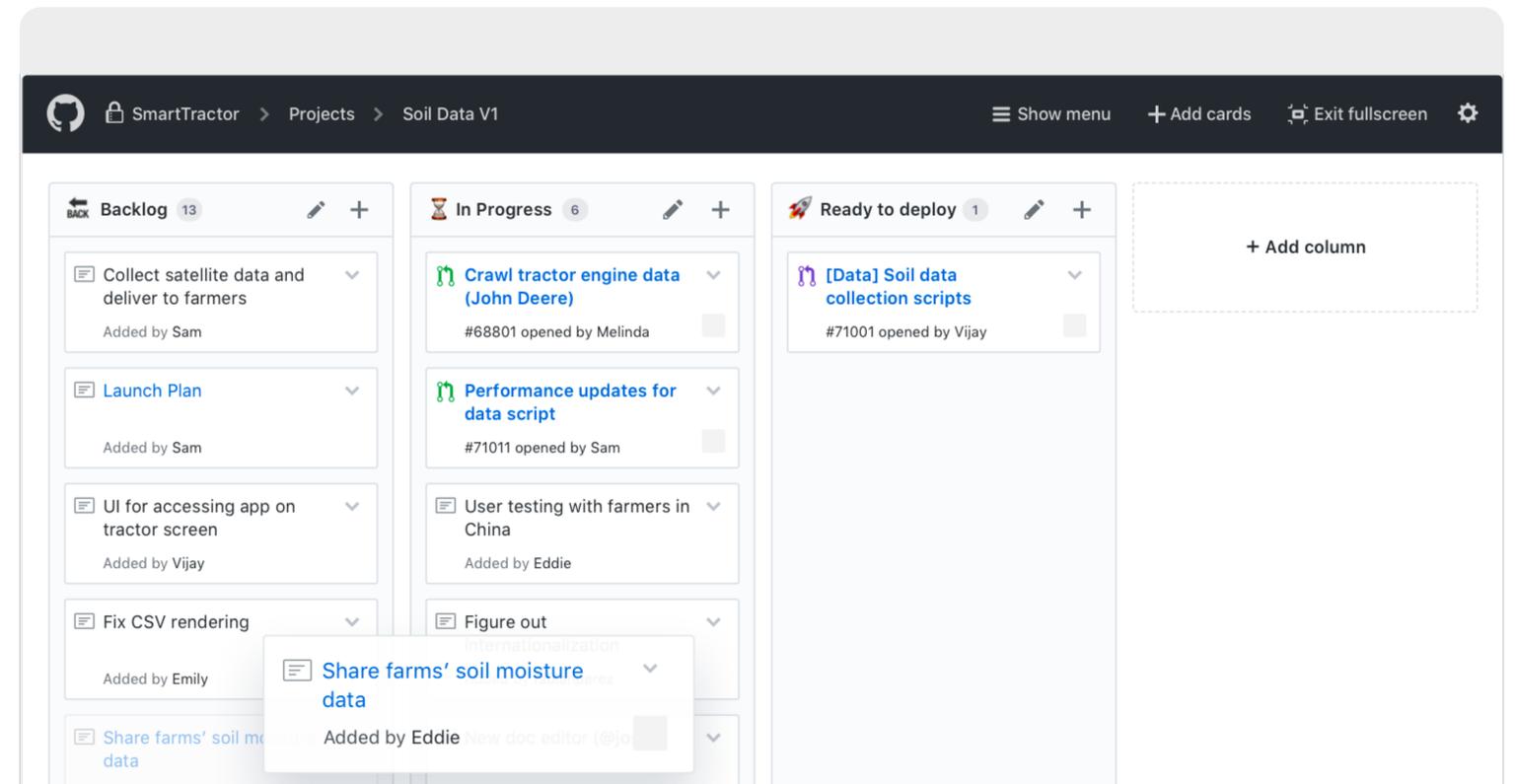


Pull Requests (PRs) are a great opportunity for design QA or as a tool for visual comparisons, but the current view doesn't include useful features for visual reviews.



Product Audit / GitHub Project Management

GitHub has started rolling out project management tools, but they are still very developer focused and don't provide collaboration features for designers.



Update Primer contribution guidelines and license

303 opened 9 days ago by broccolini

CSS brand status: wip effort: low

10

github/site Post-ship To-Dos #252 opened on Mar 1 by sophshep

1 of 2

- Refactor Jumbotron CSS: [PR here](#)
- Clean up site colors from primer-core

Thoughts?

I would love to hear what everyone thinks!

cc/ @github/design @github/creative @github/brand

6 10 9

Style guide Release 1: documentation

Due by April 30, 2017 Updated 1 day

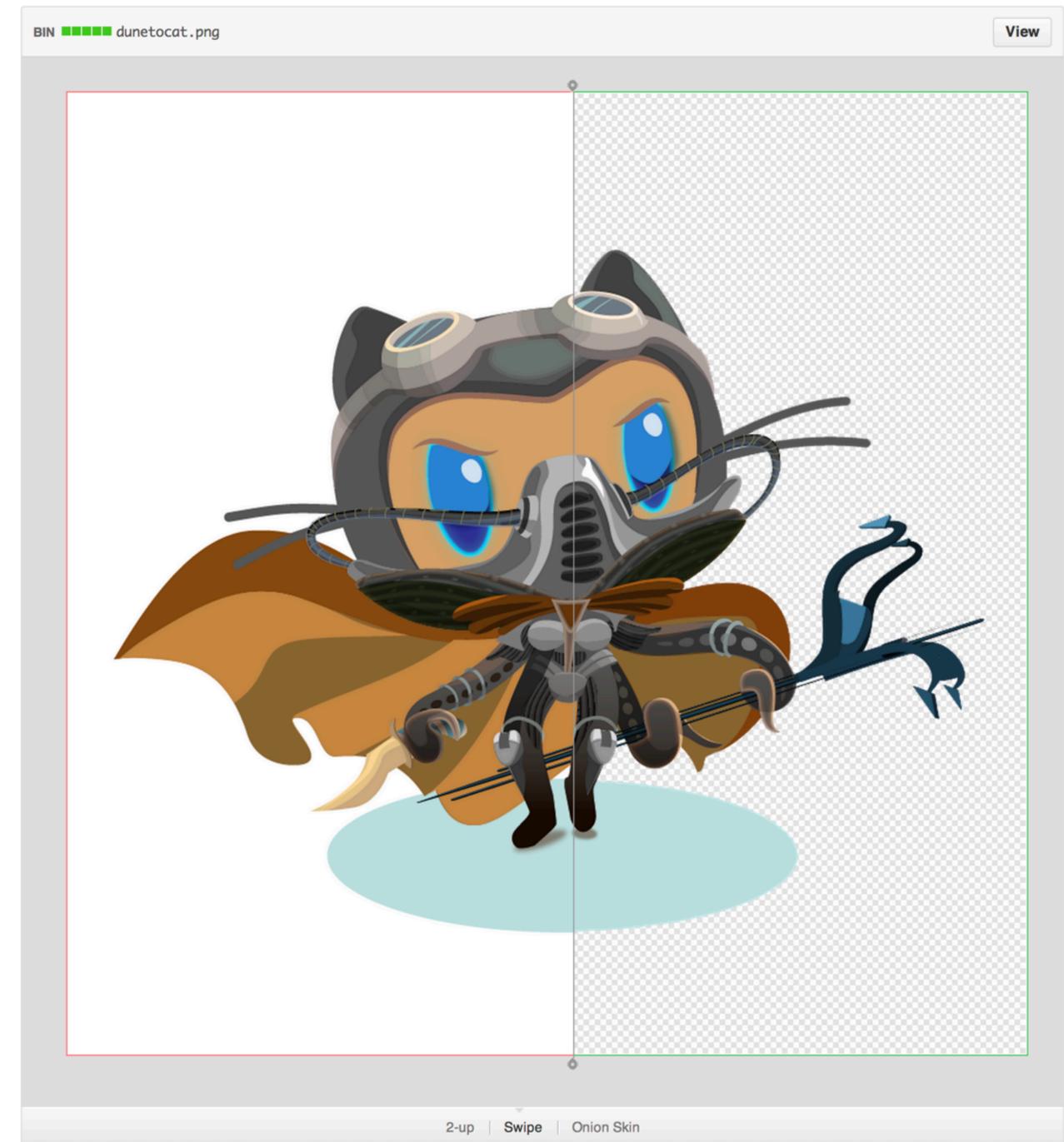
42% complete 11 open 8 closed

Edit Close Delete

Product Audit / GitHub Image Diffs

GitHub already has features to show differences in images between different code commits, which many developers find helpful.

This creates the potential to apply similar concepts to full product screenshots, or to compare against mocks instead of previously committed code.



Read about image diffs on GitHub [here](#)



“

I don't see any value in GitHub for me. I would love to get a GitHub link that I could open and actually gather some utility from, but I just don't see a use right now.

- Harrison, Product Designer

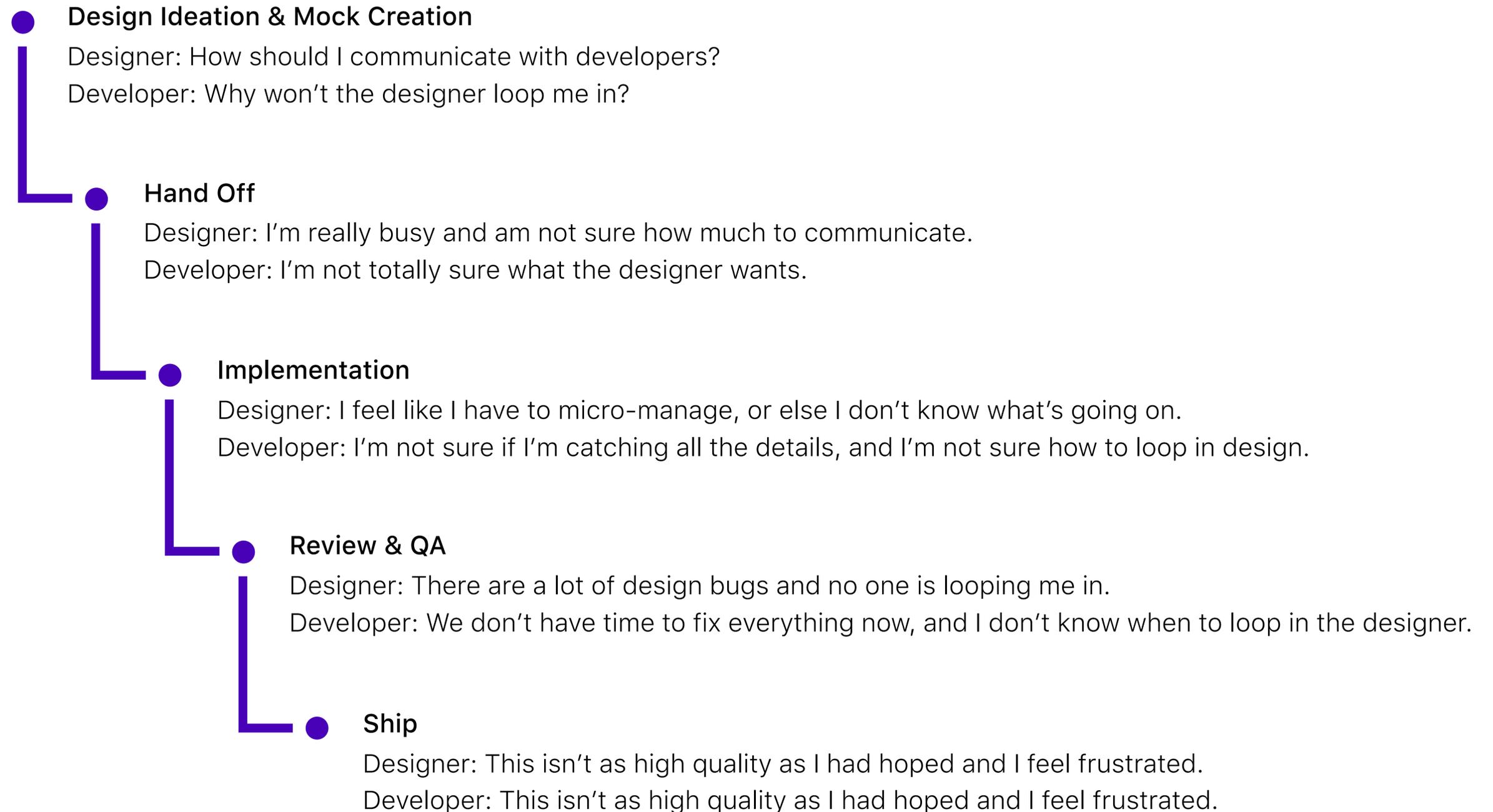


Brainstorming Strategy & Prototyping

For this project, I did a variety of brainstorming activities to make sure I explored divergent options, and then narrowed down on a solution using sketching and prototyping.

User Journey / Critical Path

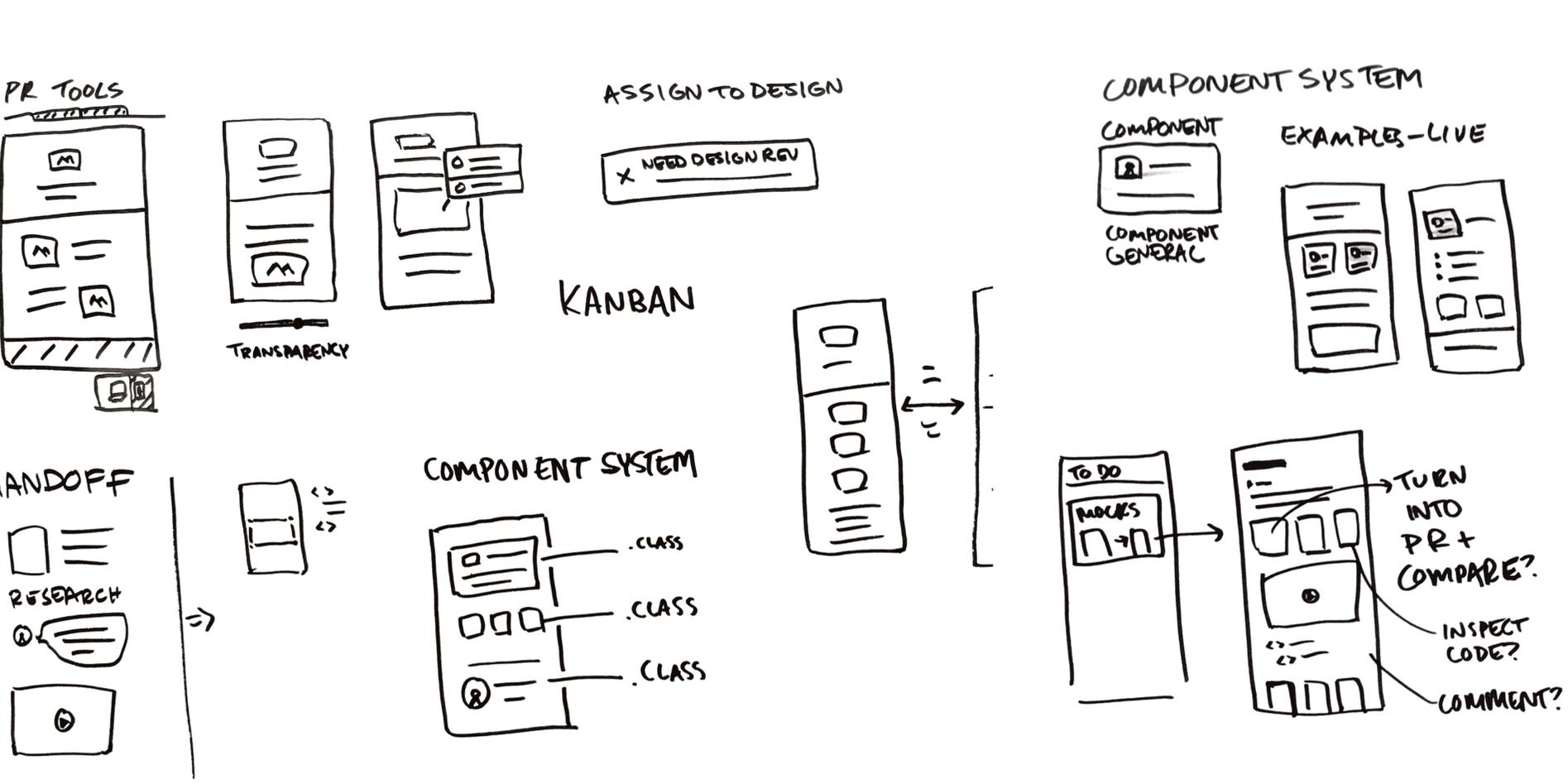
I mapped a critical path to start thinking about pain points throughout the user journey.



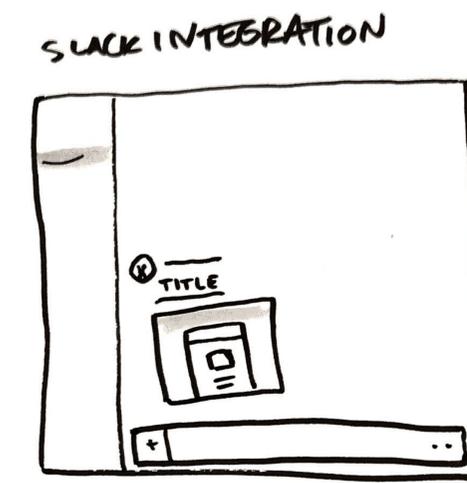
Ideation / Crazy Eights Divergent Brainstorming

I did a series of rapid timed sketching activities to produce a lot of ideas in a short amount of time.

I noted which ideas seemed the most interesting to me, and chose three concepts to sketch out in more detail.



- ### IDEAS
- * - ASSIGN REVIEWERS FLOW
 - MOCKS IN BOARD (KANBAN) - CREATE PER HANDOFFS
 - MOCKS AUTO CREATE IN PR CSS + COMMENTING + RESPONSIVE?
 - * - IMAGE DIFFS ANIMATION DIFFS
 - COMPONENT LIBRARY UI TO EDIT W/OUT CODE?
 - ZEPLIN IN GITHUB + WALKTHROUGHS
 - MOBILE / WEB APP VIEWING W/OUT DEVEL. ENVIR.

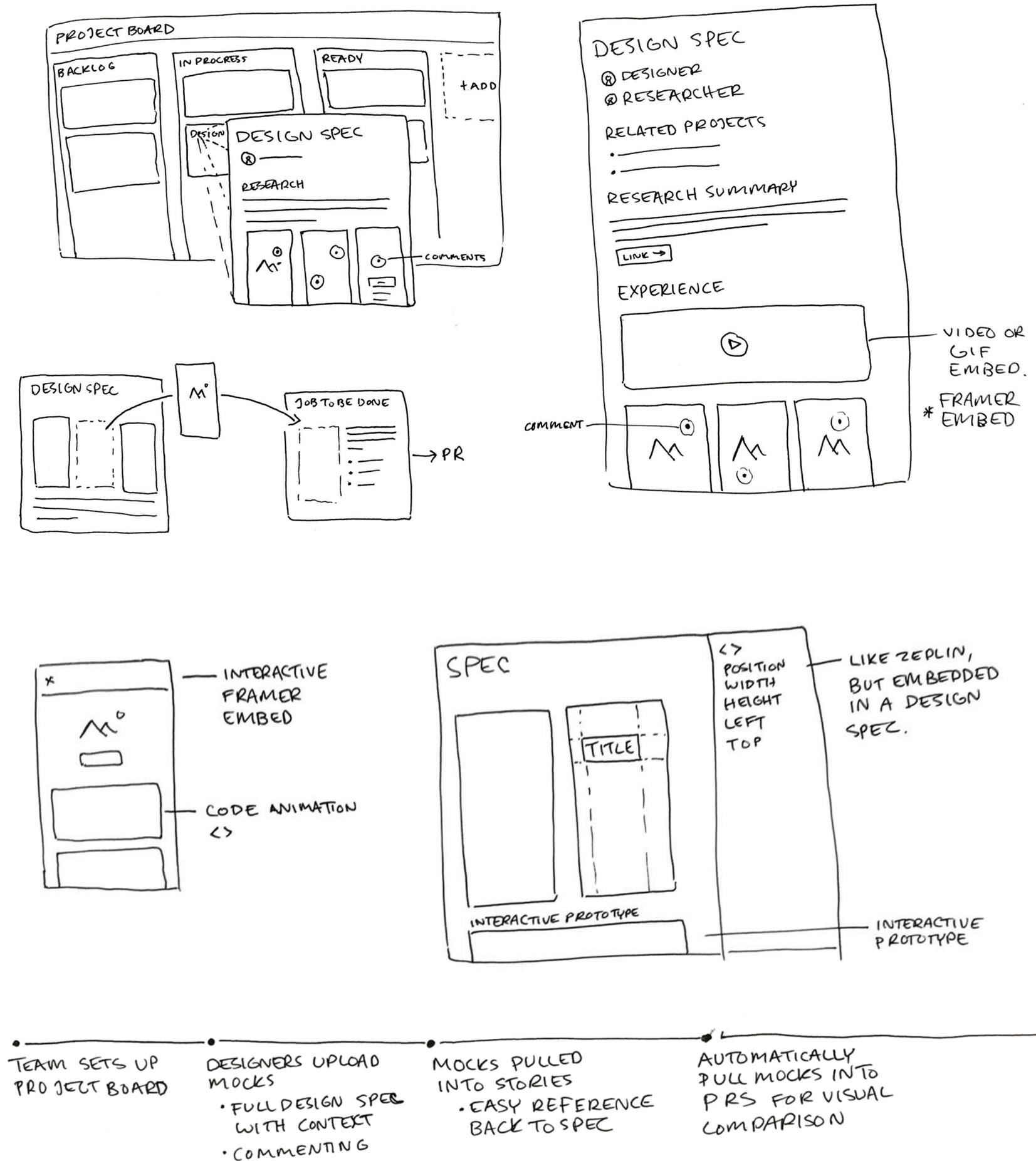


- QUICK TAP / VIEW ON PHONE
- FORCE TAP COMMENTS
- PREVIEW LOOKS MORE LIKE A MEDIUM POST / NOT TECHNICAL
- REVIEW NEEDED - DESIGN

Ideation / Concept 1: Design Tools in Kanban Boards

Since GitHub recently introduced kanban-style project boards, I thought it would be interesting to incorporate design into the process. Some ideas include:

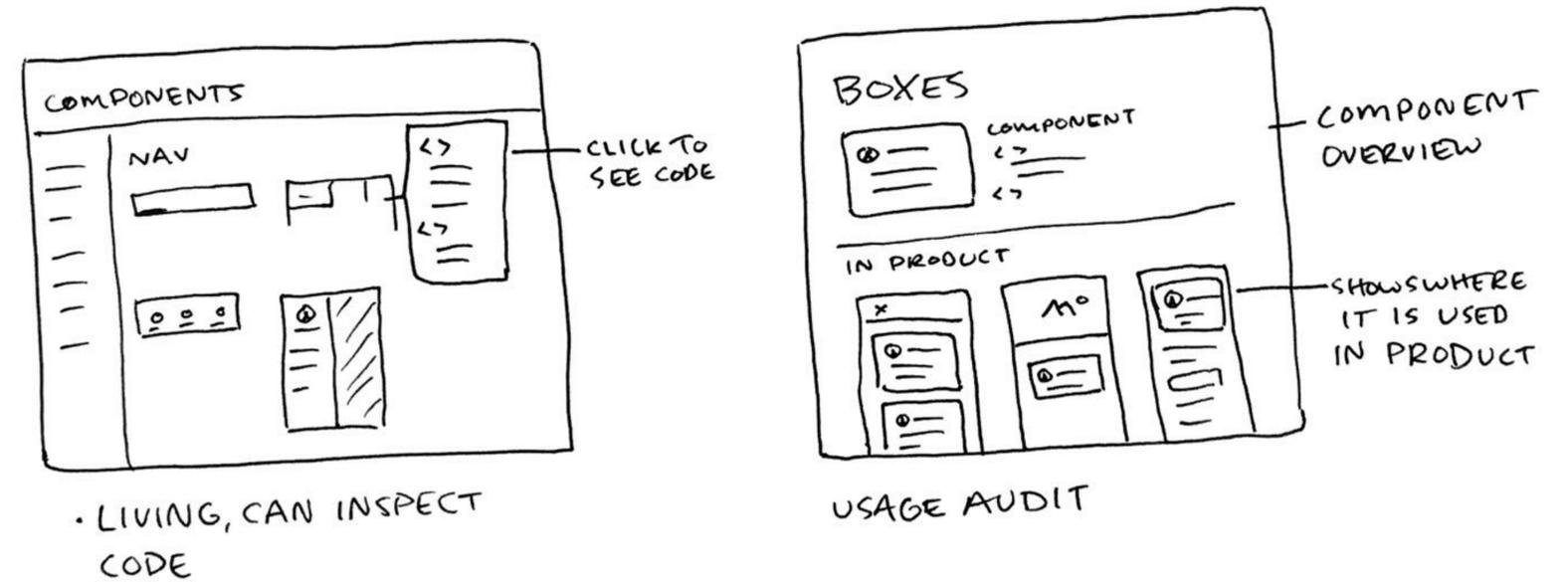
- Create a "design spec" card for the project board.
- Design specs can provide redlining features, as well as show the full intention of the design through diagramming and research.
- Mocks can be pulled from design specs and attached to other cards.
- Cards that get converted into Pull Requests could attach the mocks as a reference.



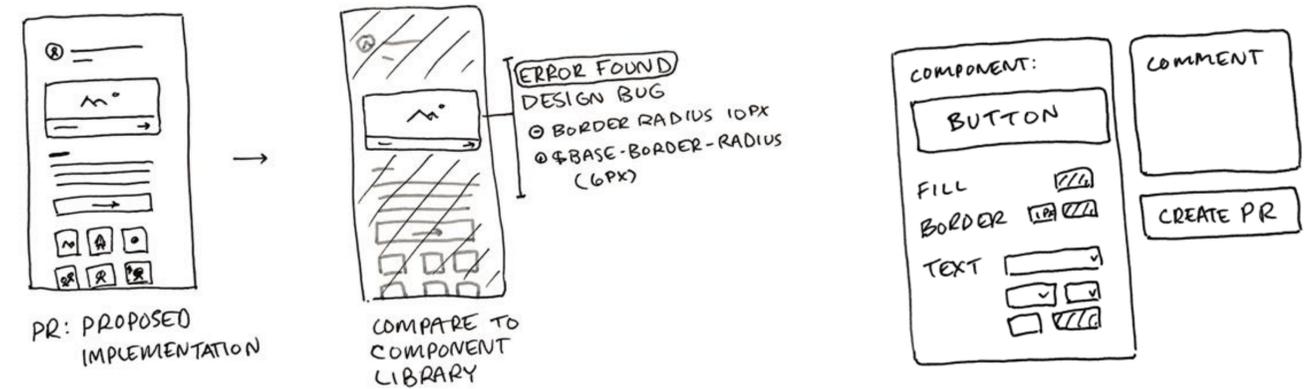
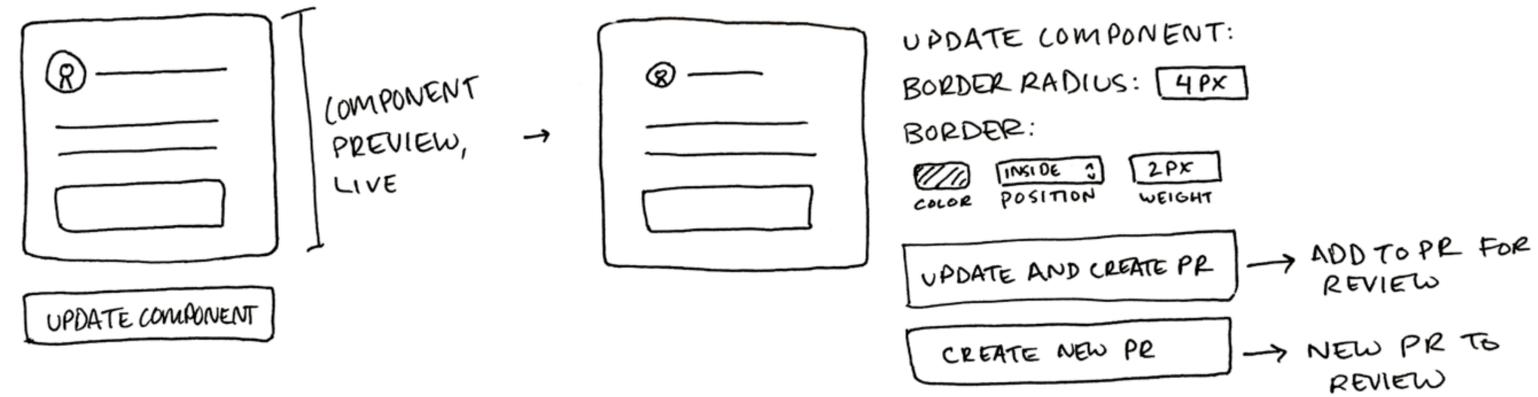
Ideation / Concept 2: Integrated Component Library

Creating component systems is always a challenge, often because designers and developers have a hard time keeping everything updated.

This concept was an exploration around creating a living component library. Designers could update components through a user-friendly UI that automatically submits code for review.



CARD COMPONENT



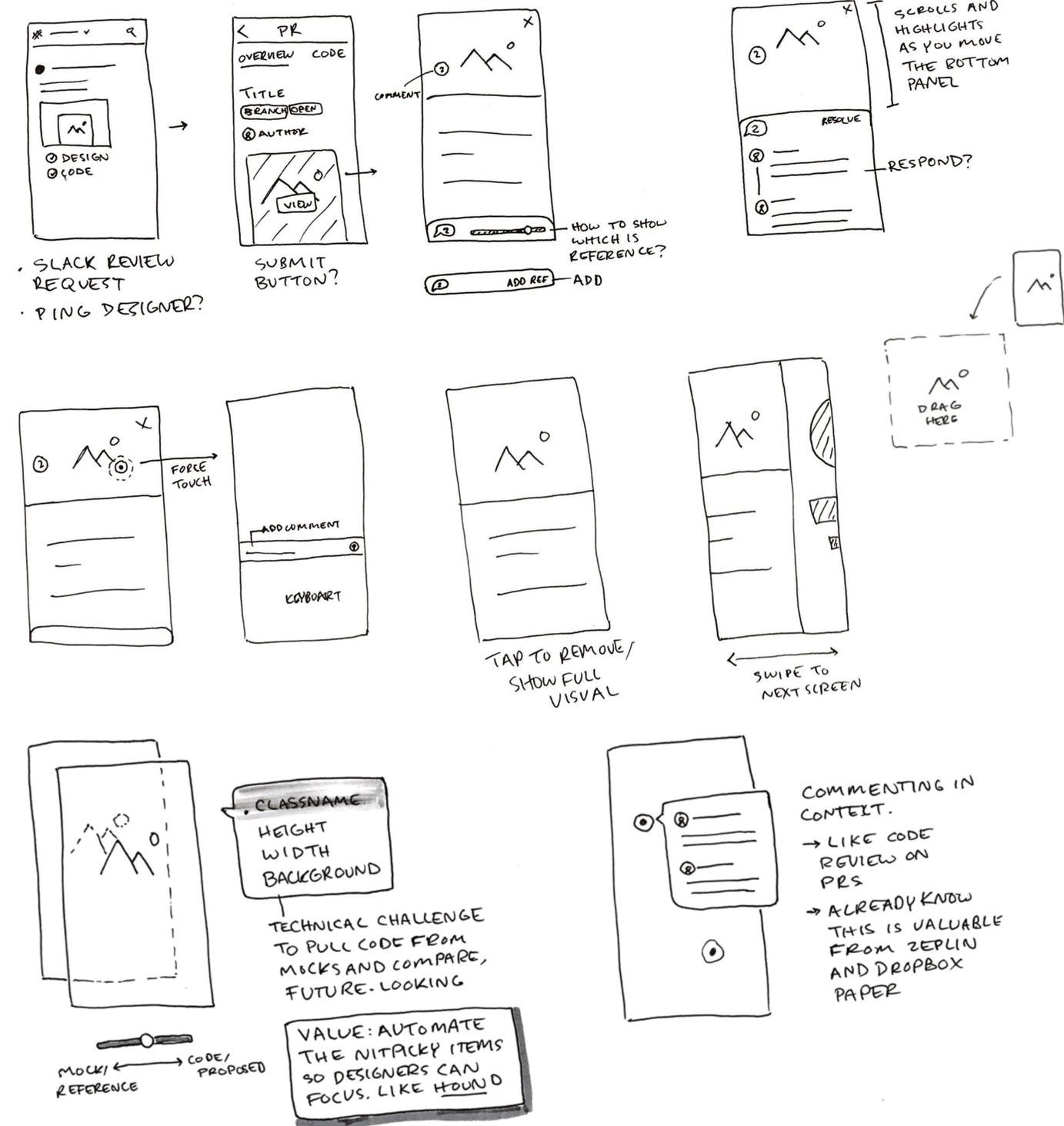
- LIVE COMPONENT LIBRARY
 - DESIGNERS + ENGINEERS CAN TRIGGER SITE AUDITS THAT WILL AUTOMATICALLY SHOW (VISUALLY) WHERE THE COMPONENTS ARE USED.
 - DESIGNERS CAN UPDATE COMPONENTS VIA A SKETCH-LIKE INTERFACE AND CREATE A PR BY CLICKING A BUTTON
 - ENGINEERS + DESIGNERS CAN USE A LIVE COMPARISON TOOL TO SEE DESIGN BUGS IN IMPLEMENTED SCREENS

Ideation / Concept 3: Visual Reviews in PRs

Code review processes currently do not include design checks, even when the code is changing the design.

This concept explores updating the Pull Request flow to include automated and manual visual checks. Some of the value includes:

- Automatically tagging designers for visual reviews.
- Including visual changes alongside code changes.
- Collaboration tools for better communication.
- A potential focus on mobile to improve the quality of feedback. Viewing mobile work on an actual device is closer to the real thing and will improve feedback.



ENGINEER IMPLEMENTS

- NEEDS TO CHECK VISUAL BUGS AS WELL AS CODE
- COMPARES MOCKS TO IMPLEMENTATION

SUBMITS REVIEW

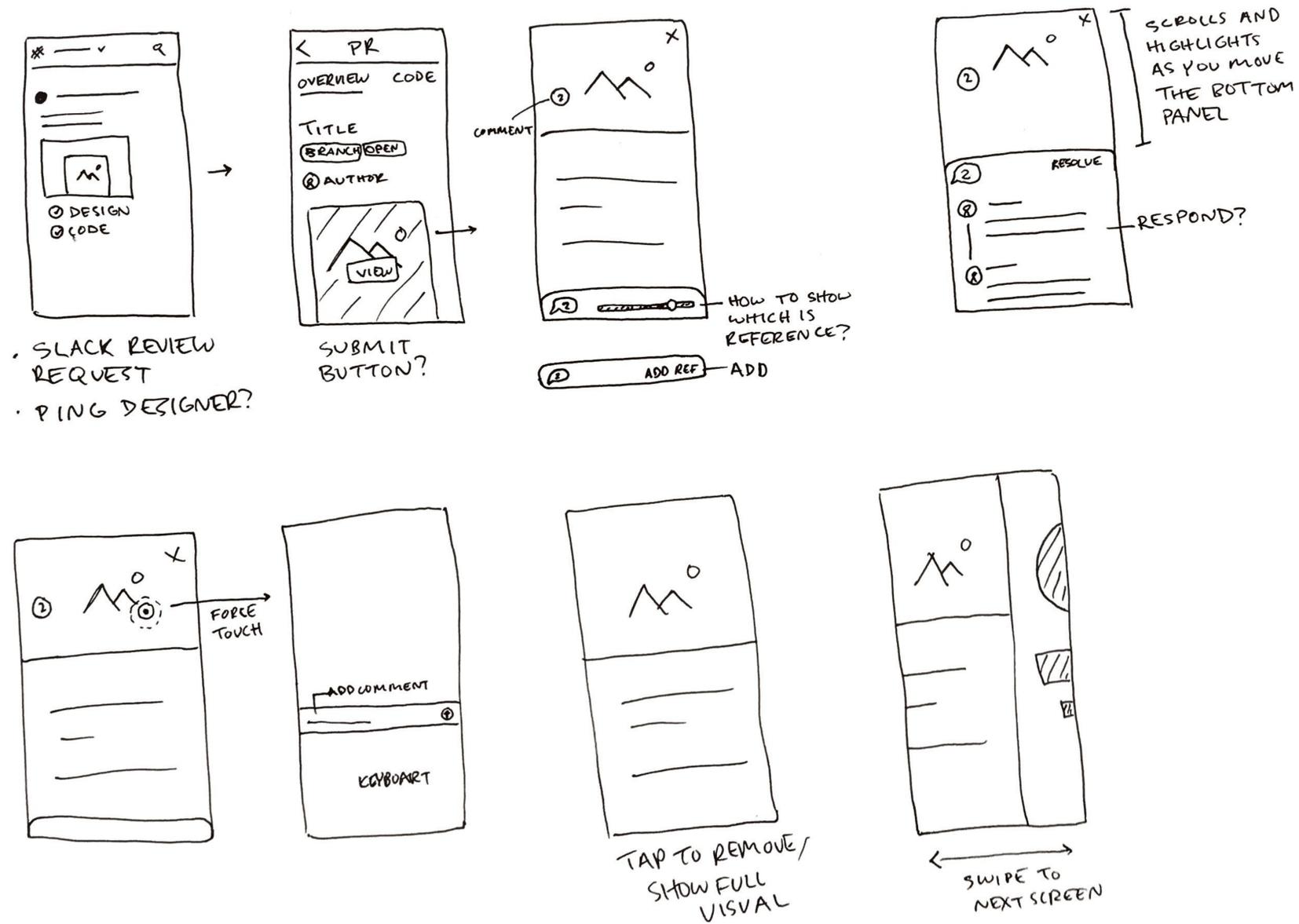
- INCORPORATE VISUAL CHECKS
- PREVIEW VISUALS, INCLUDING MOBILE
- MAKE DESIGN REVIEW AN INTEGRATED PART OF PROCESS
- PINGS DESIGNER AUTOMATICALLY

DESIGNER REVIEWS

- ADD COMMENTS DIRECTLY TO PREVIEWS
- VIEW VISUALS ON PHONE FOR REALISTIC FEEL
- THUMBS UP OR DOWN

I ultimately chose the third concept: Visual Reviews in PRs

I decided to do a native version to show how it could optimize feedback for mobile work.



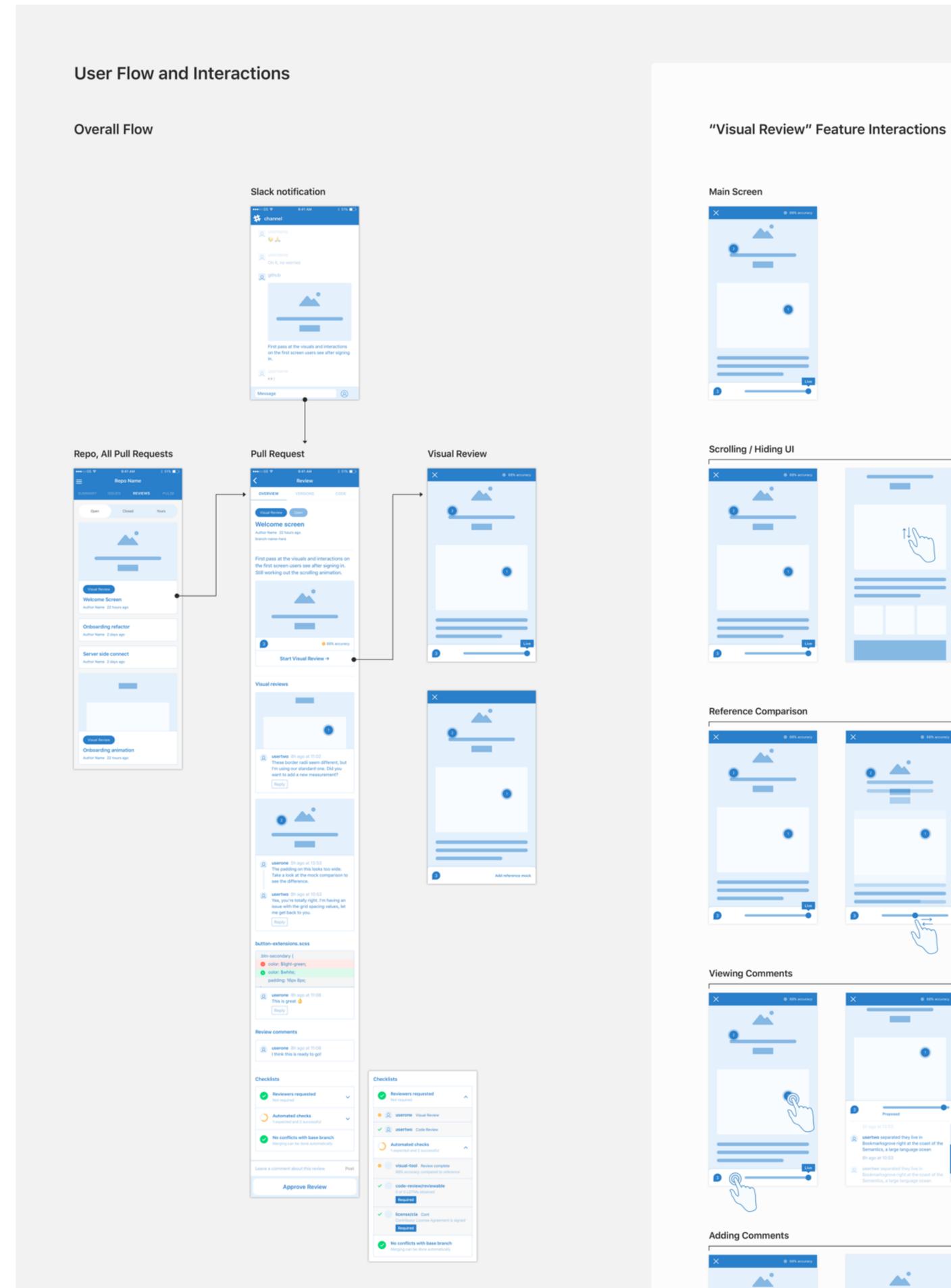
Wires and Flows / Full Diagram

For a first pass at this feature, I scoped the experience to the designer's perspective and established three main points in the user experience:

1. A "hook" to alert the designer that a review needs attention.
2. A redesigned Pull Request flow that includes checks for visuals.
3. A feature that allows designers and developers to check the accuracy of the visuals by comparing the implemented version to the reference mocks.

As an add-on, I decided to redesign the list of PRs in the main repository to apply the new visual styles to the entire flow.

[View full wireframe mocks here →](#)



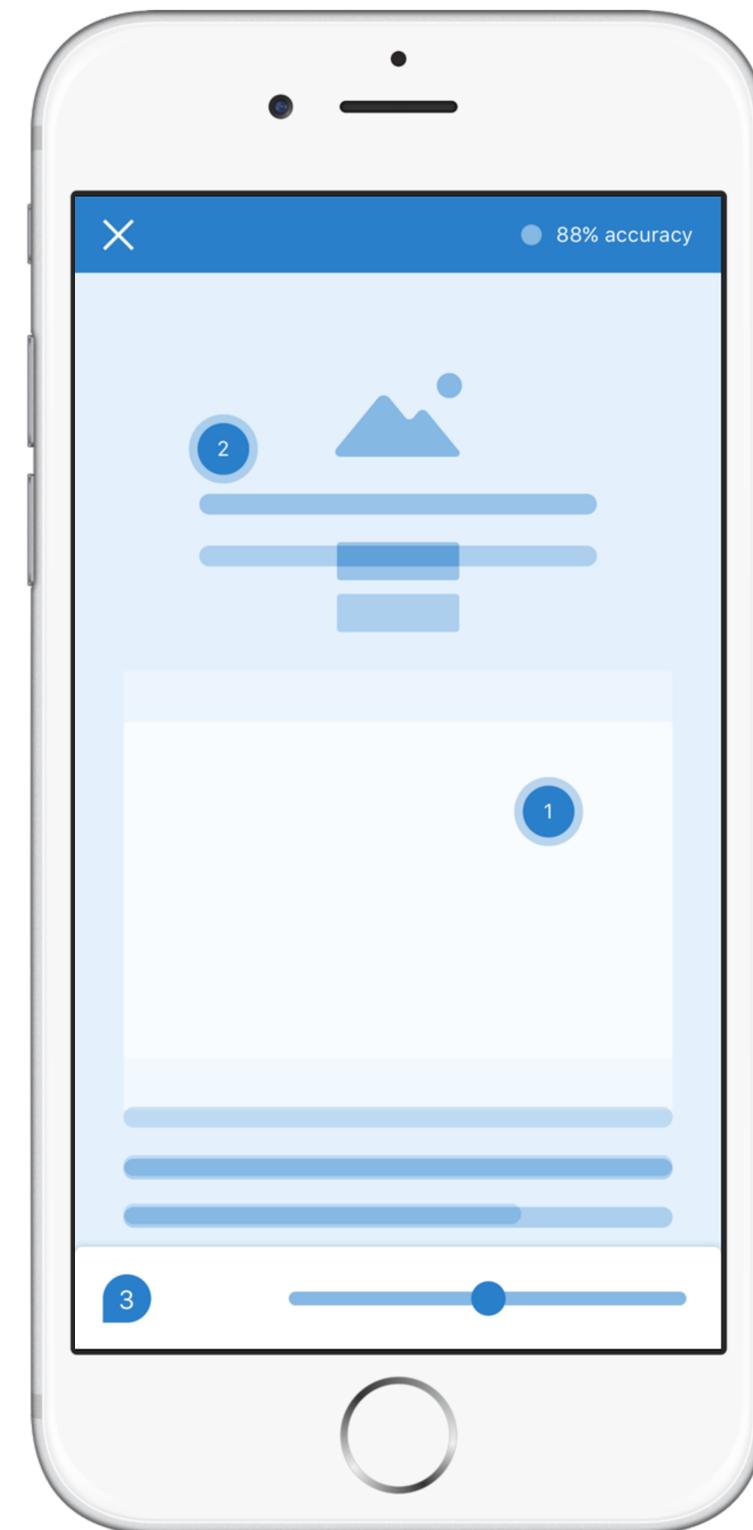
Wires and Flows / Interactions

I decided to prototype the interactions for the “Visual Review” feature before moving on to visual design. Since this feature would rely heavily on interactions, it felt important to work out some of the issues before moving forward.

One improvement I made after prototyping was that I realized that the navigation bar on this screen was too heavy and obscured the view.

I also realized how future-facing the automated “best guess” code feature was when playing around with an actual prototype. I decided to make sure the final design would hold up without this feature and use it to establish future direction.

[View prototype with interactions here →](#)



Automated checks
1 expected and 2 successful

visual-tool Review Complete
92% accuracy comparison

code-review/reviewable
0 of 0 LGTMs obtained
Required

license/cla
Contributor License is signed
Required

No conflicts with base branch
Merging can be done automatically

92% accuracy

Start Visual Review →

Requested reviewers
Not required

Automated checks
1 expected and 2 successful

No conflicts with base branch
Merging can be done automatically

San Francisco

Charlie Ogunse 1m ago at 2:30 pm
Can we keep the route vertically centered?

Overview Versions Code

Visual Review **Open**

Driver Ride History Detail

Jenny Ko 22 hours ago
branch-name-here

First pass at the driver ride history detail view. I'm trying to automate how the map views are generated, so right now it's a placeholder.

Visual Review

Driver Ride History Detail

Jenny Ko 22 hours ago
branch-name-here

Suggested change

```
.route-map {  
- height: 406px;  
+ height: 475px;
```

Comment

Onboarding Refactor

Eunice Lituanas 2 days ago
branch-name-here

Code

Mock

route

Pick-up 11:15 AM

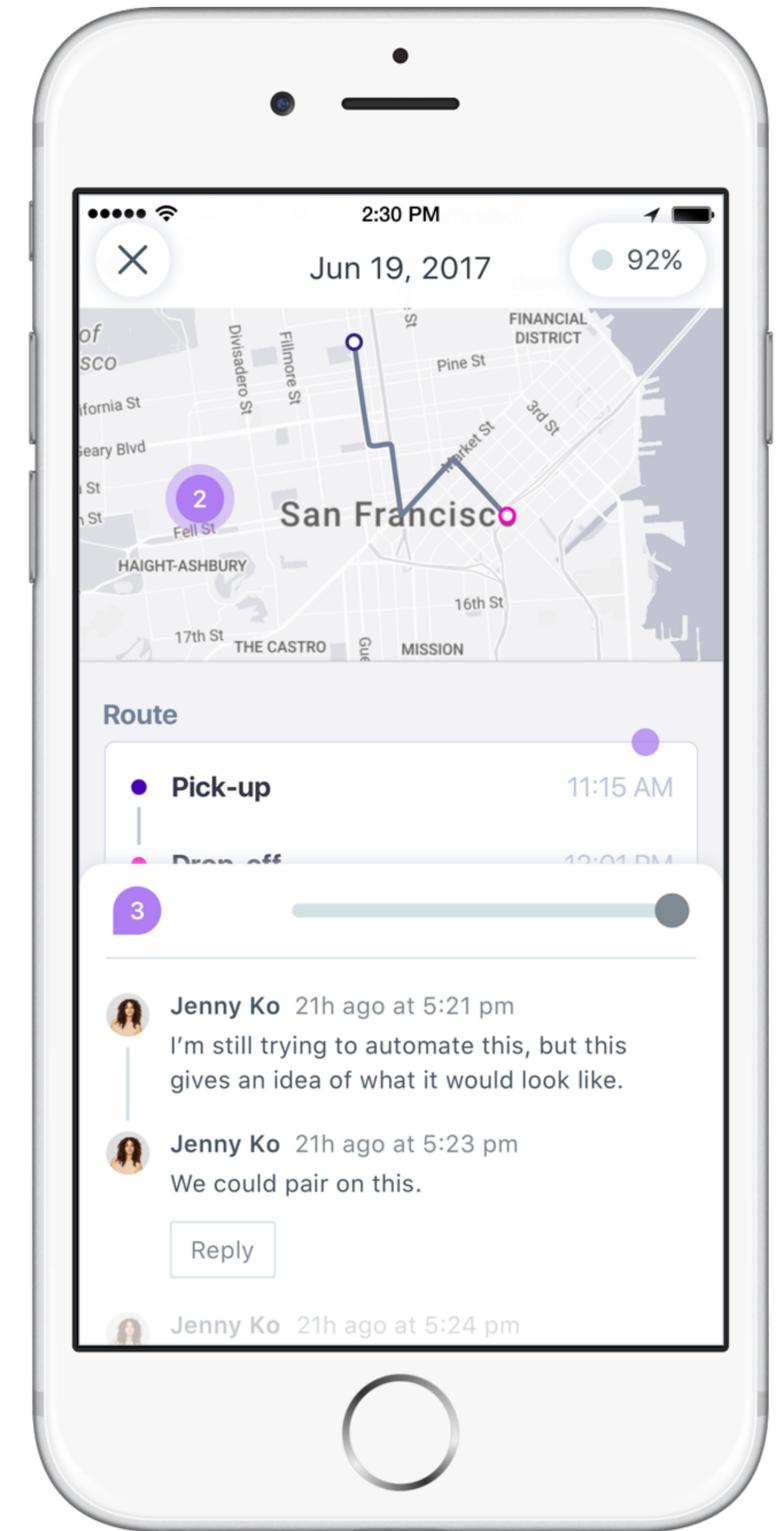
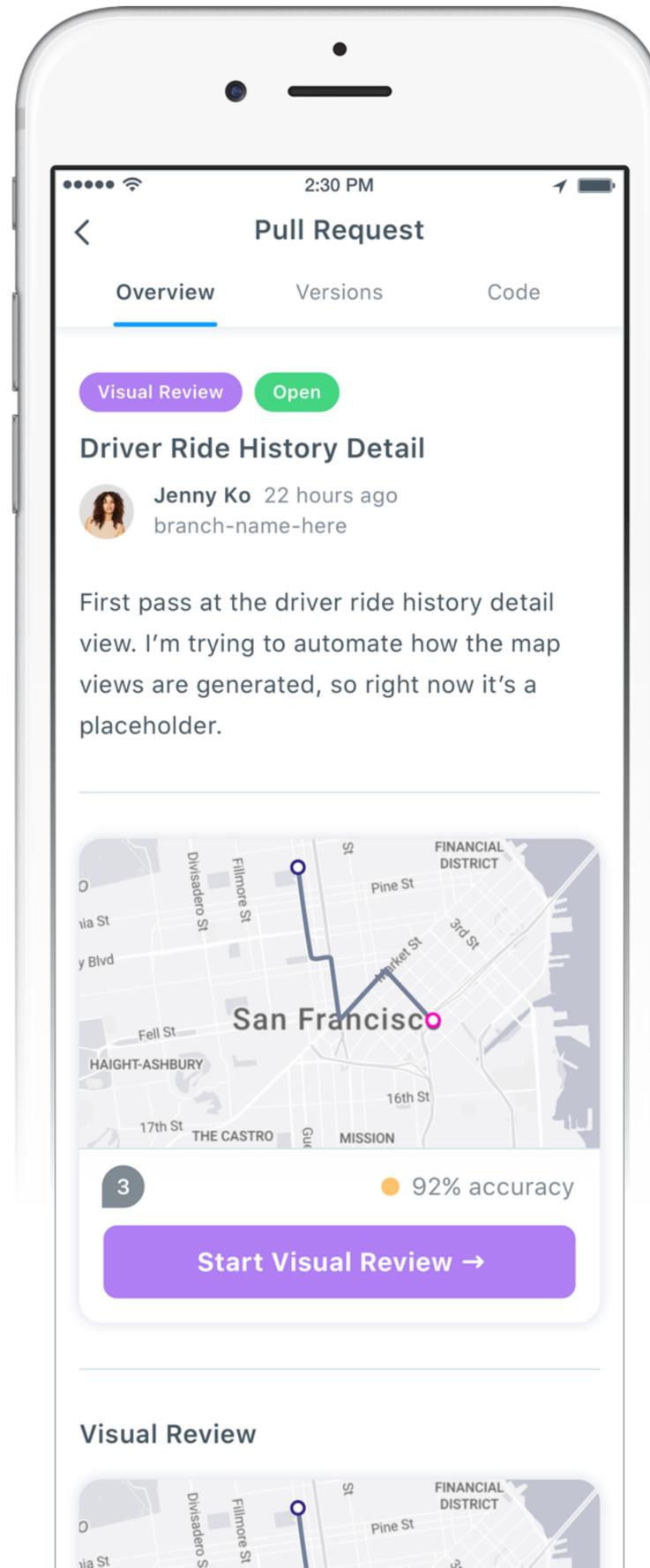
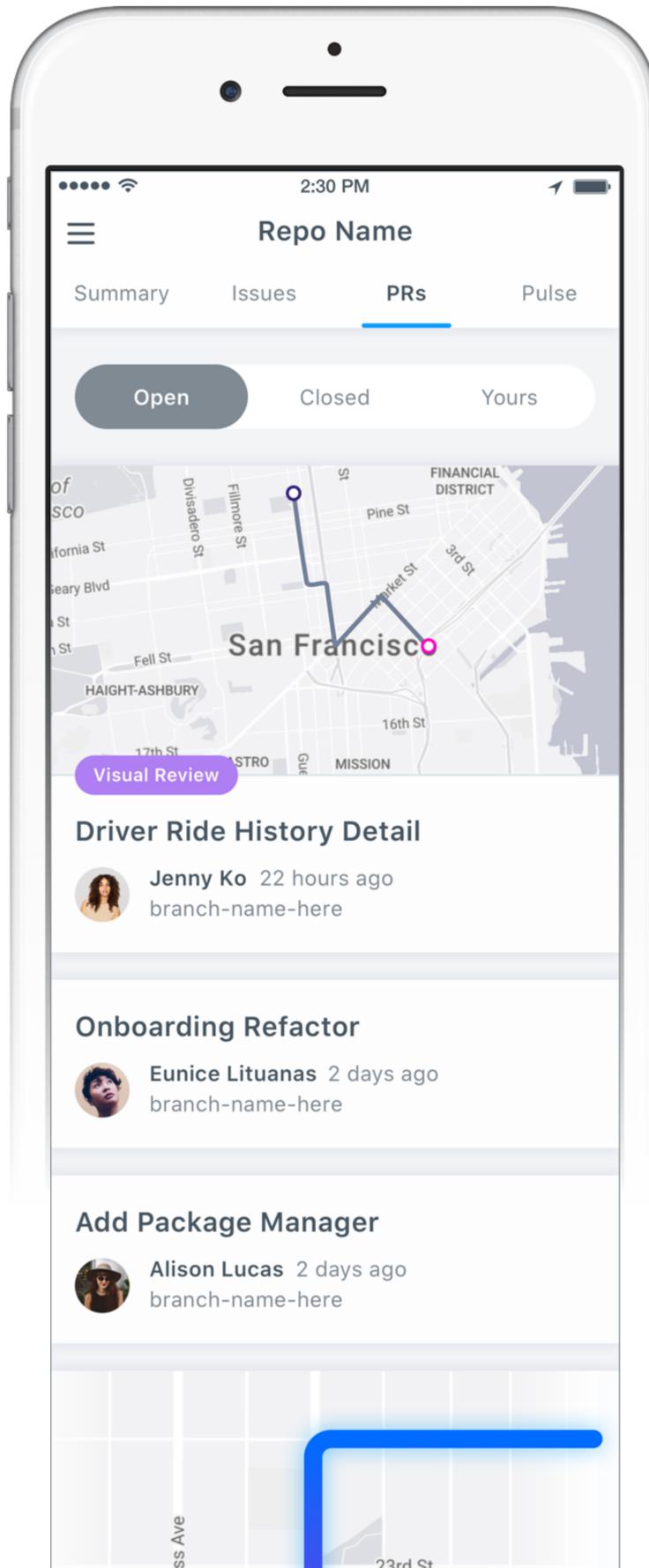
Drop-off 12:01 PM

Jenny Ko 21h ago at 5:24 pm
This is a new border radius unit, should we add this to the style guide?

Final Design

Visual Reviews for PRs

The following is a walkthrough of the final design for the Visual Review feature.



Visual Review feature

Final Design / Slack Entry Point

Scenario: An engineer just submitted a PR for a feature that includes design work. The designer was included in the list of reviewers as a “visual reviewer”.

The designer gets pinged when the GitHub integration posts the PR in Slack. This is the quick entry point to the PR.

The “users” are:



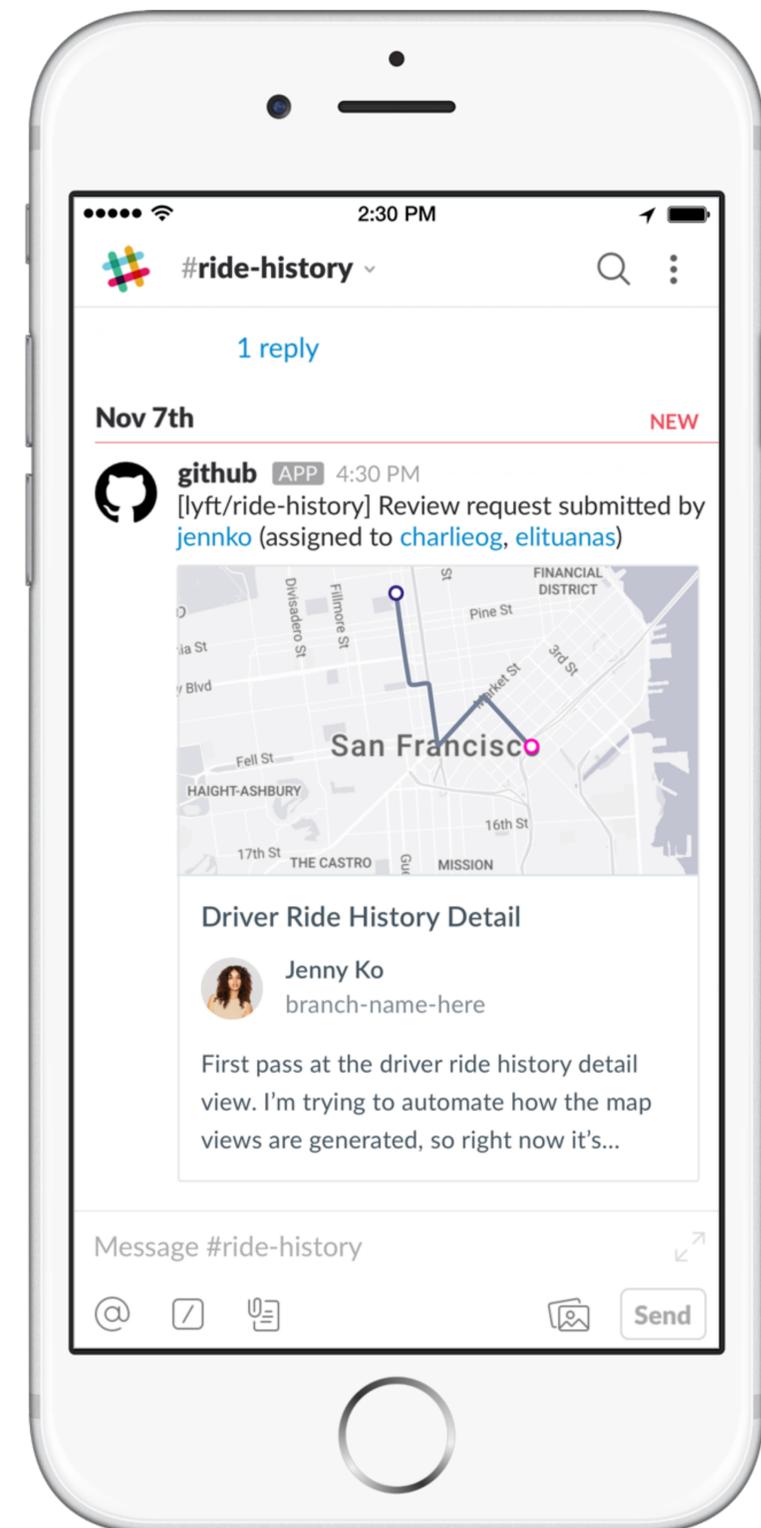
Jenny Ko, PR author



Charlie Ogunse, Designer



Eunice Lituanas, Developer

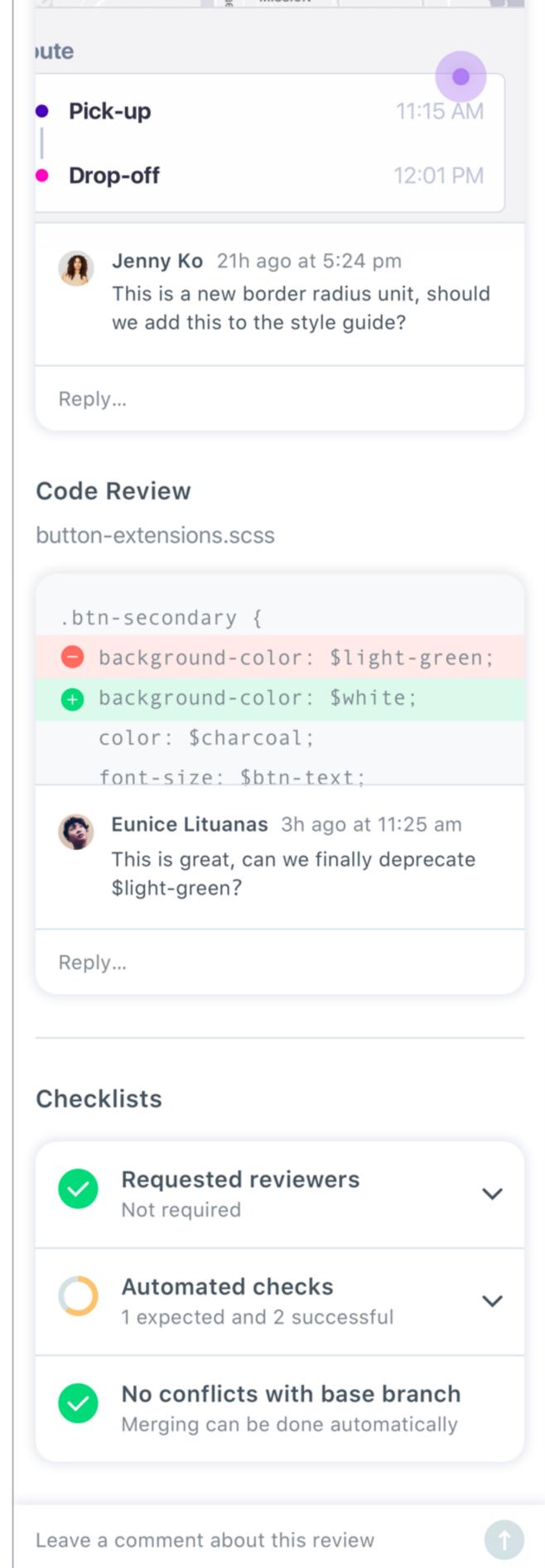
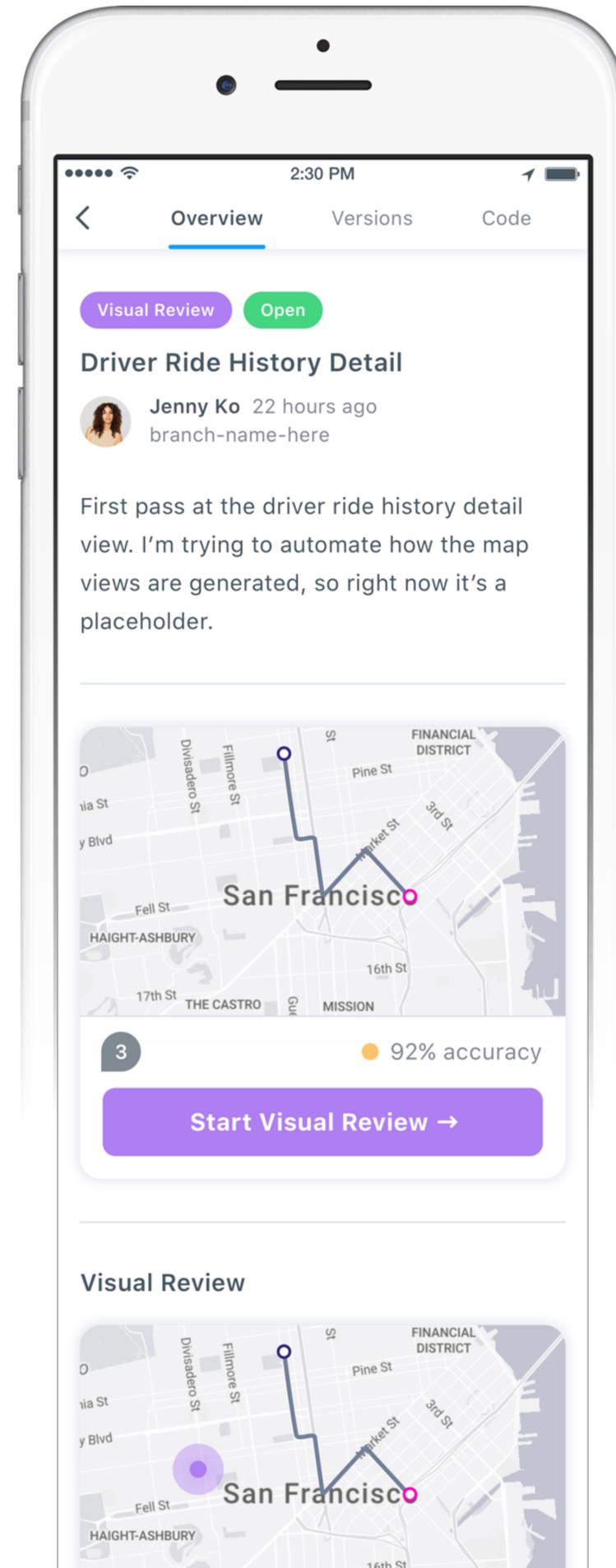


Final Design / Pull Request With Integrated Visual Review

This is the redesigned PR screen that the designer sees. It includes:

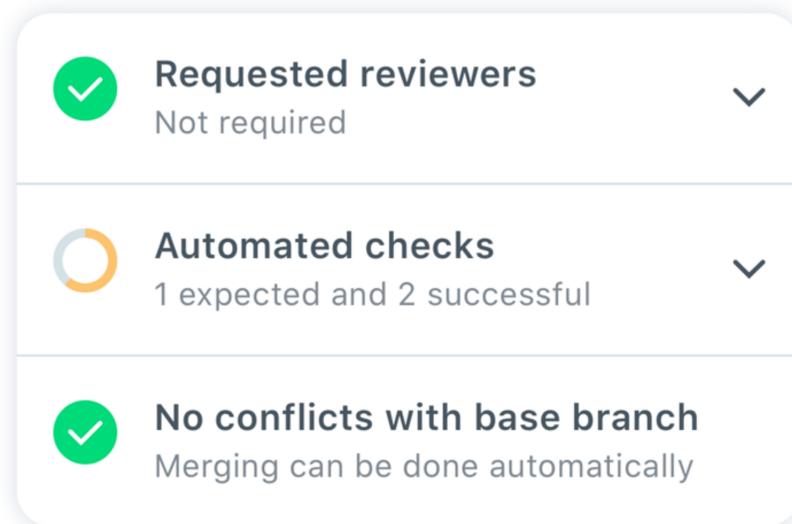
- An up-front summary section with an overview.
- A Visual Review feature incorporated into the main experience with a clear call-to-action, increasing its integration into the development process.
- Updated comment blocks for code and visual feedback.
- An updated checklist module that includes visual review checks.

[View a detailed PR anatomy here →](#)



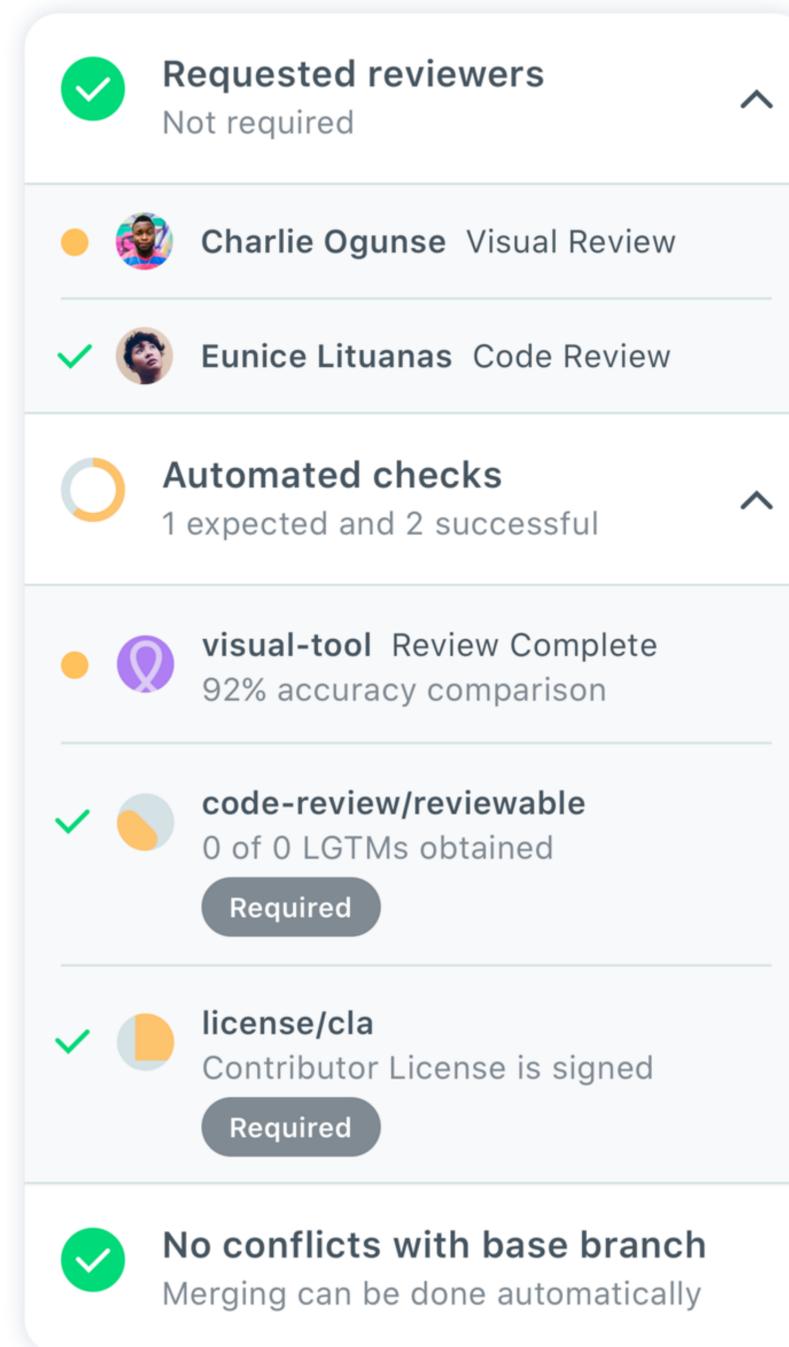
Final Design / Pull Request Checklist

This is a closer look at the updated checklist module, which includes visual review features.



A summary view of the pull request checklist. It consists of three items, each with a status icon and a dropdown arrow:

- Requested reviewers** (Green checkmark icon): Not required
- Automated checks** (Yellow progress circle icon): 1 expected and 2 successful
- No conflicts with base branch** (Green checkmark icon): Merging can be done automatically



A detailed view of the pull request checklist. It shows the same three items as the summary view, but with expanded details for the 'Requested reviewers' and 'Automated checks' sections:

- Requested reviewers** (Green checkmark icon): Not required
 - Charlie Ogunse Visual Review
 - Eunice Lituanas Code Review
- Automated checks** (Yellow progress circle icon): 1 expected and 2 successful
 - visual-tool** (Purple location pin icon): Review Complete, 92% accuracy comparison
 - code-review/reviewable** (Green checkmark icon): 0 of 0 LGTMs obtained, Required
 - license/cla** (Green checkmark icon): Contributor License is signed, Required
- No conflicts with base branch** (Green checkmark icon): Merging can be done automatically

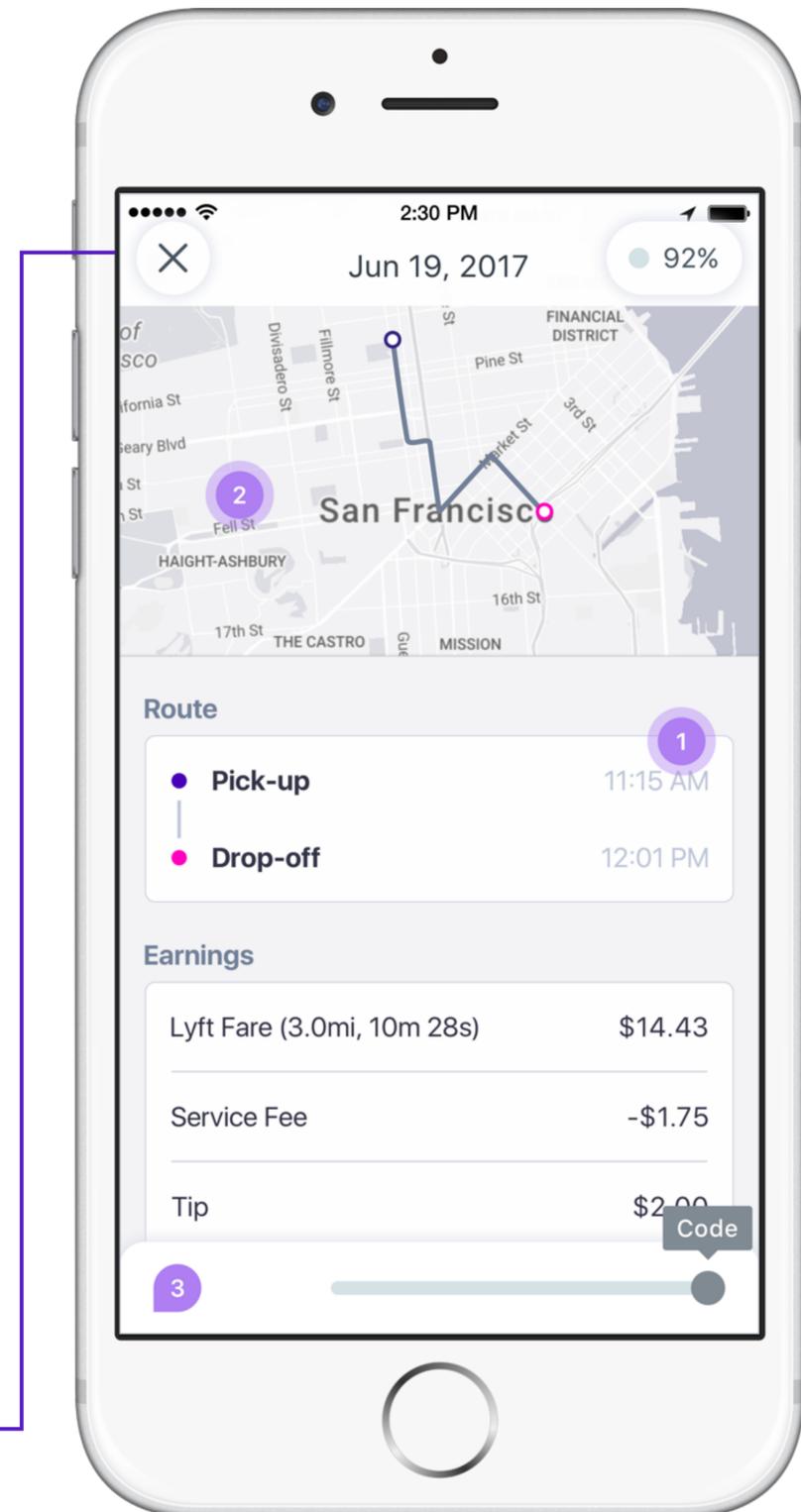
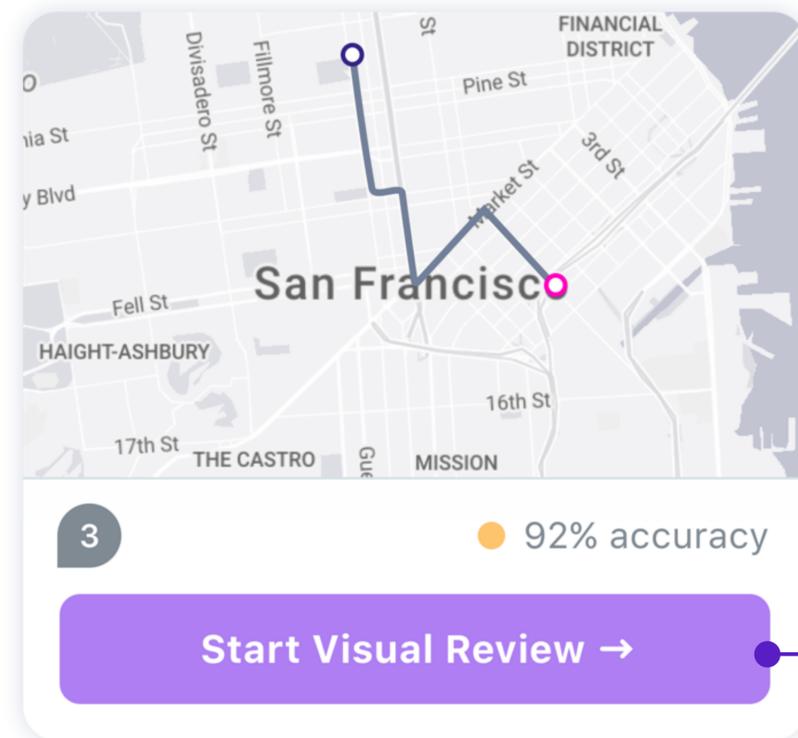
Visual Review Request
Assign designers to review visuals

Automated Visual Comparison
Future facing idea

Final Design / Visual Review Entry Point

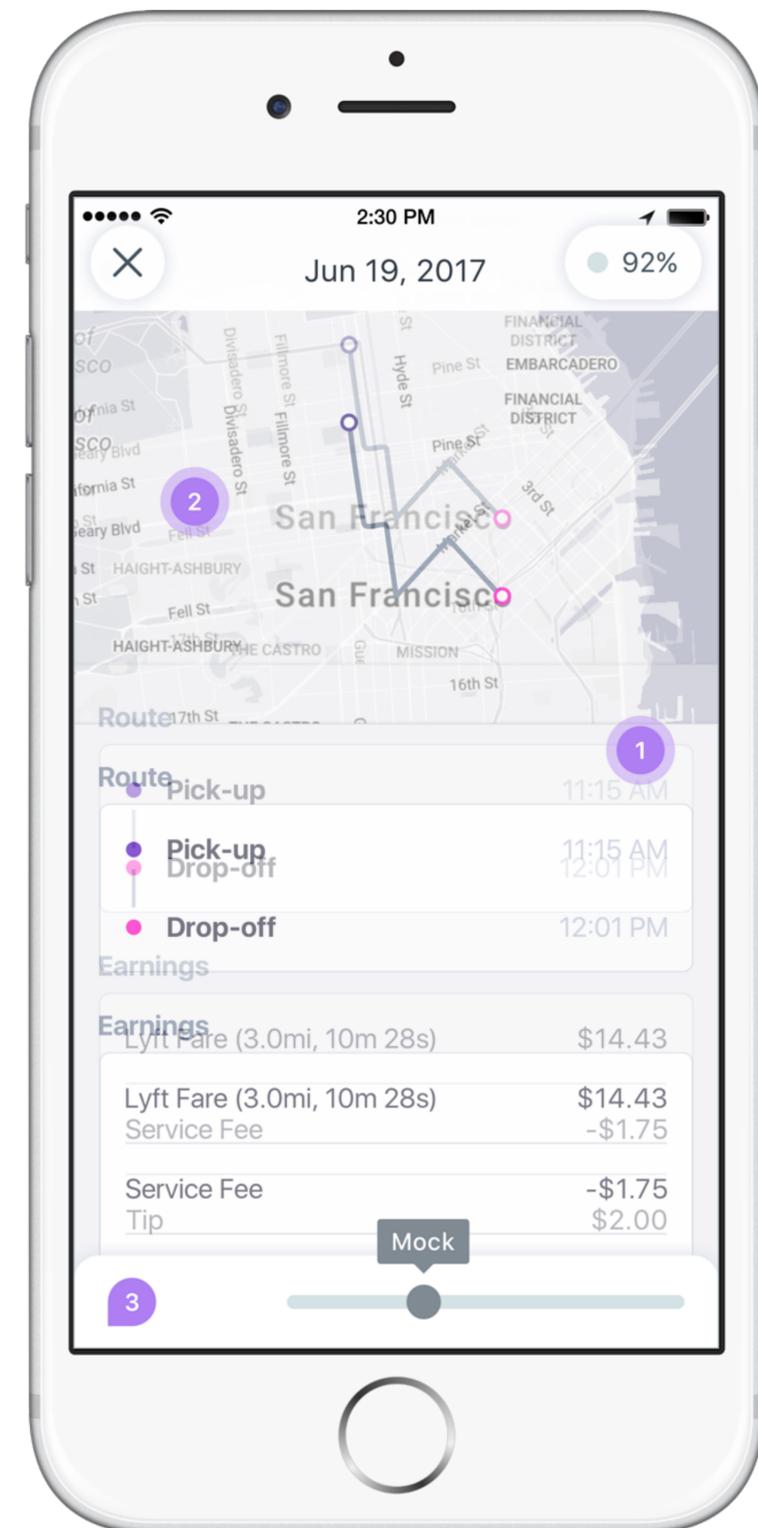
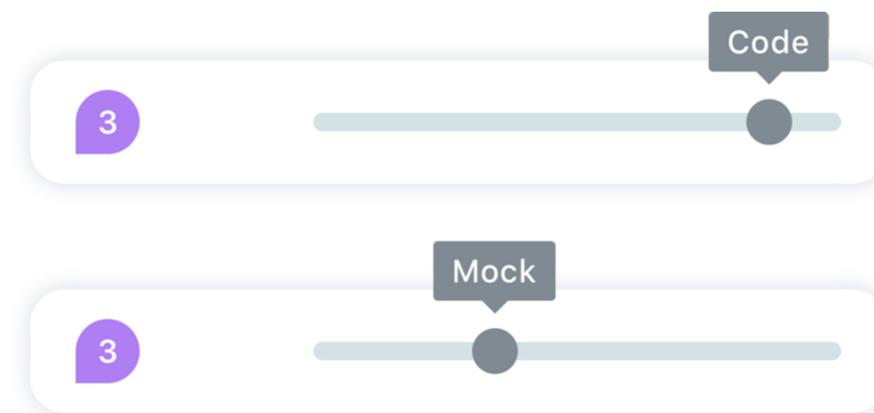
The designer can start a visual review by tapping the module on the PR screen. This launches the Visual Review feature.

[View a detailed Visual Review anatomy here →](#)

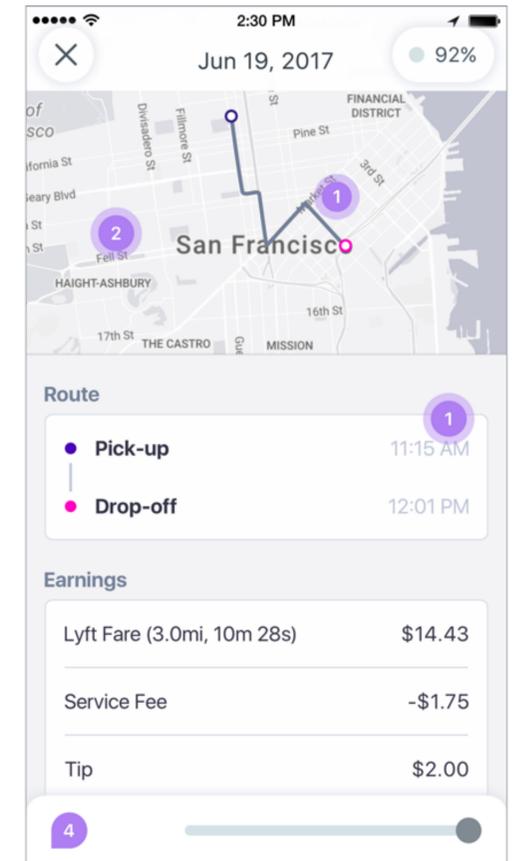
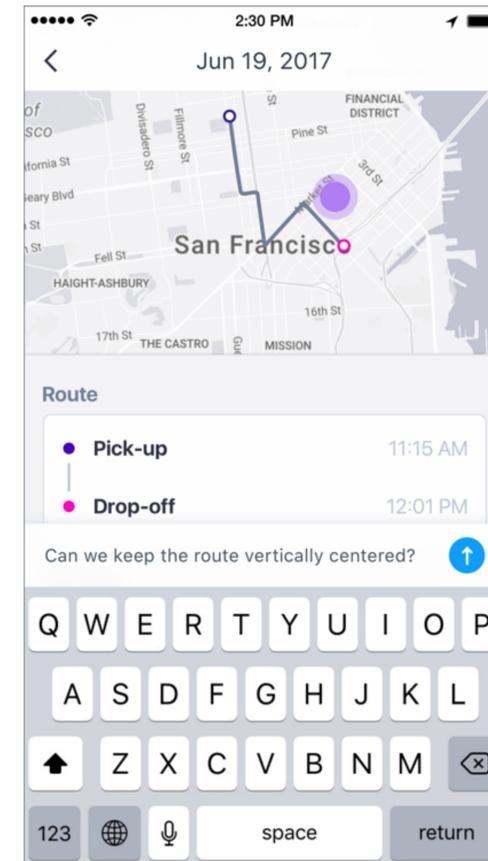
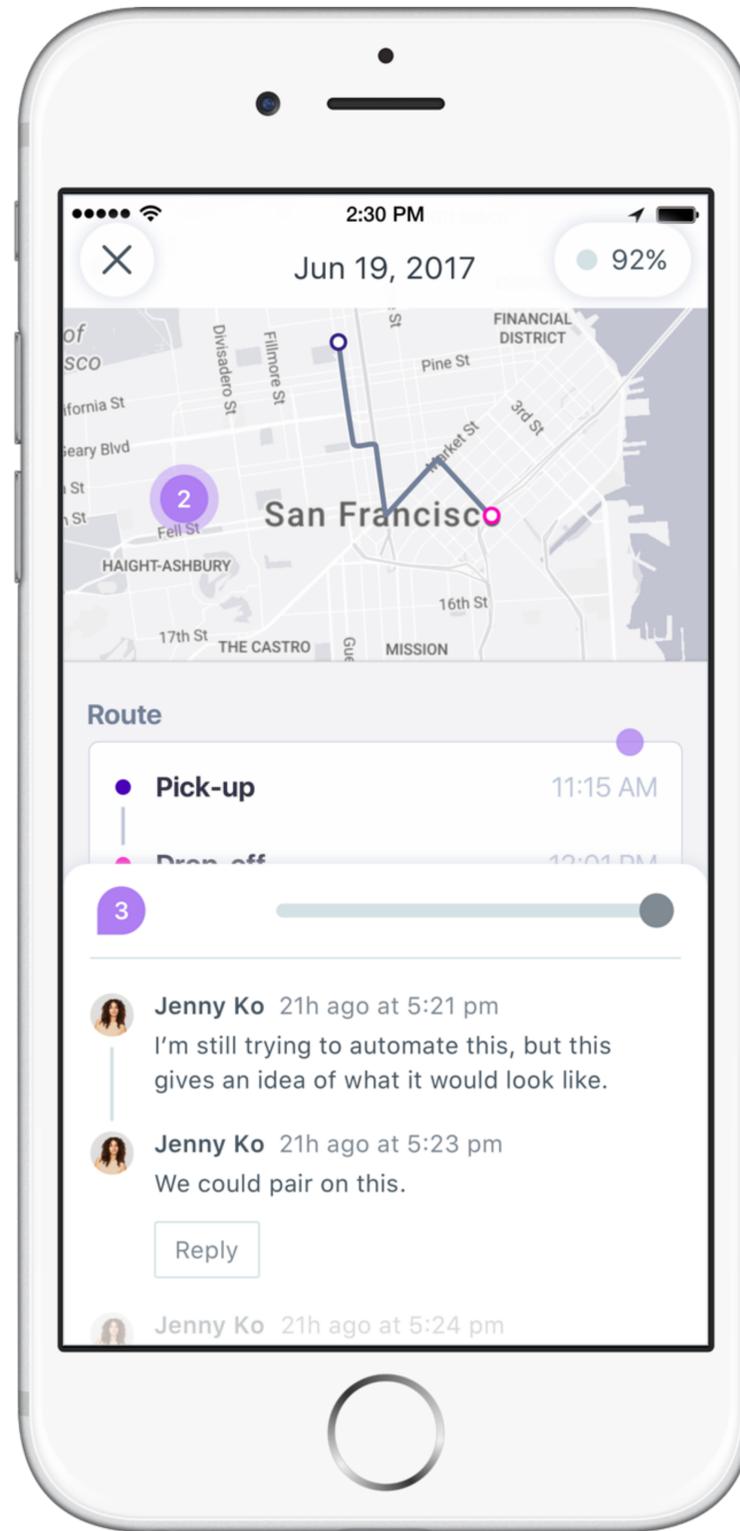
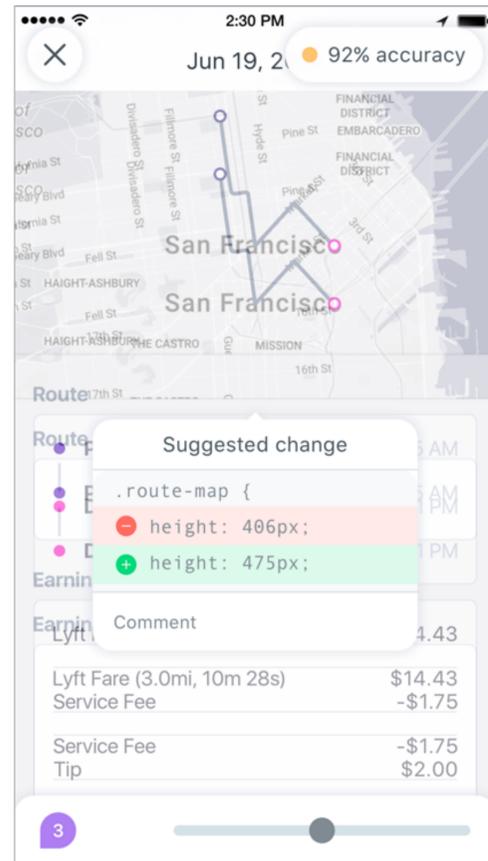
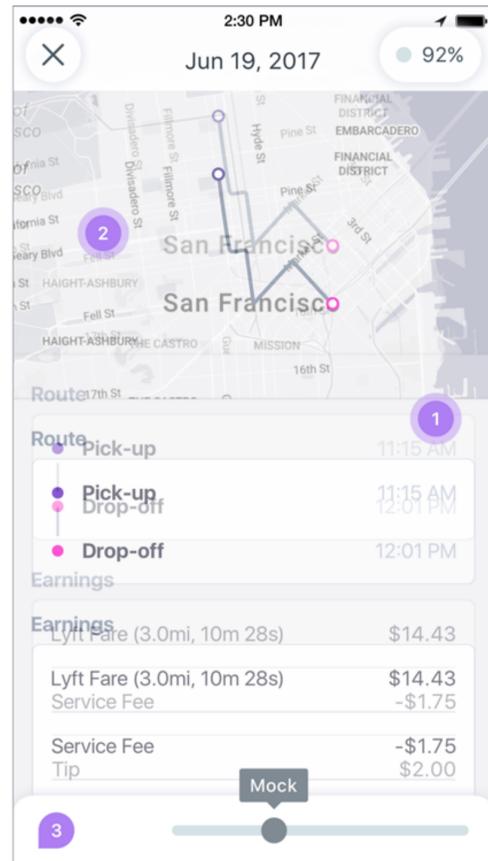


The Visual Review feature includes:

- An “onion skin” comparison between the screenshot and reference mocks.
- Commenting in context.
- The ability to swipe to see more screens.
- A future-facing idea for automatically catching design bugs and suggested CSS updates. This would be out-of-scope for the near term.



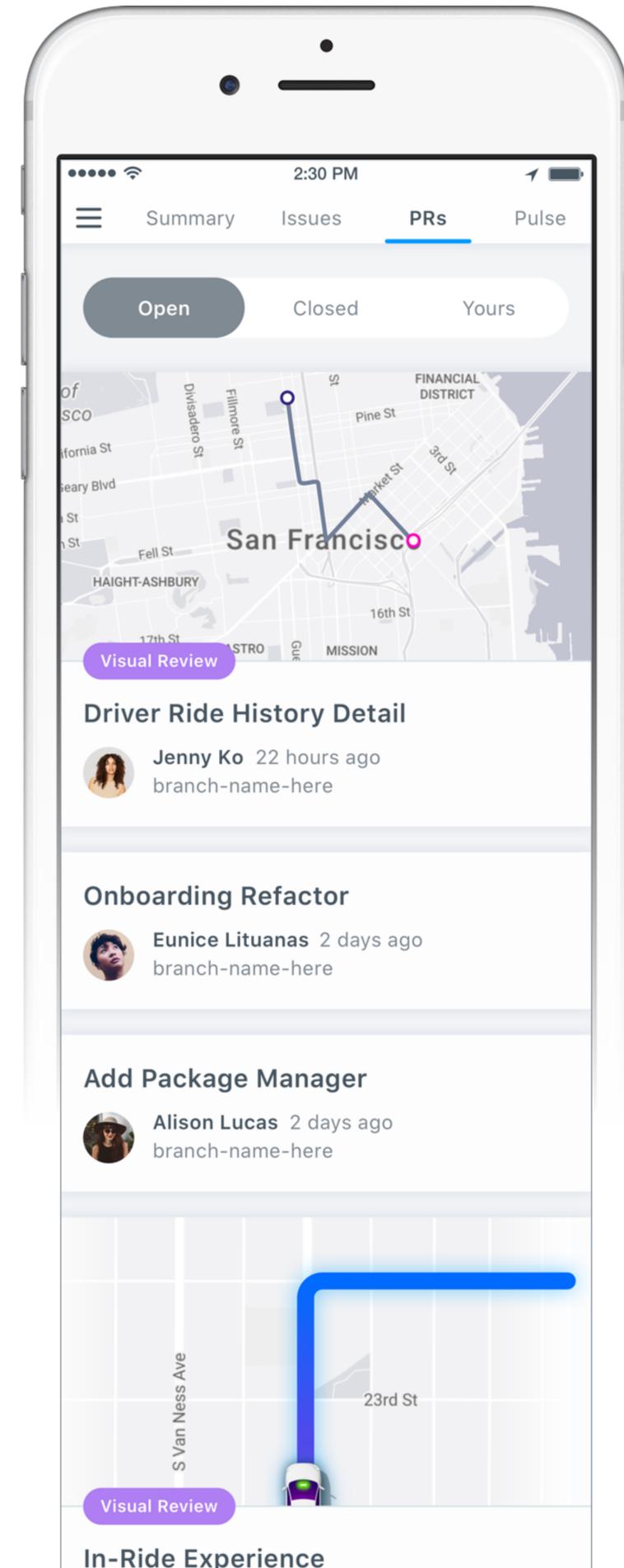
Final Design / Visual Review Interactions



[View prototyped interactions here →](#)

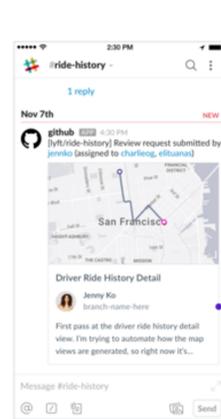
Final Design / Repository List of PRs

The user can also go up a level in the navigation to see the full list of PRs in the current repository.

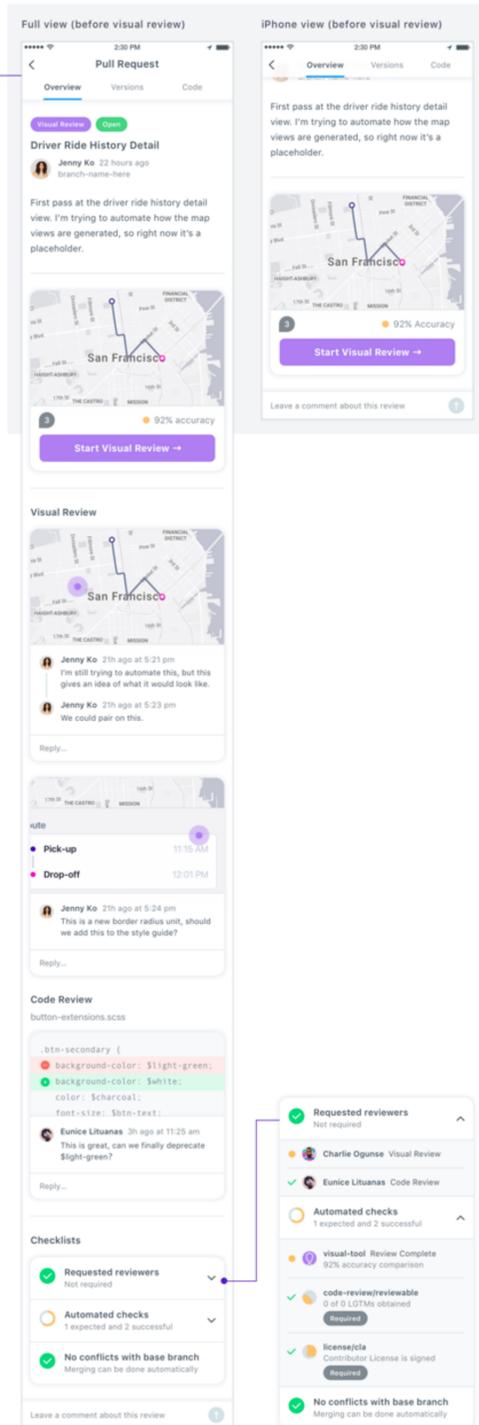


Final Design / Overall Experience

Slack Automatic Embed



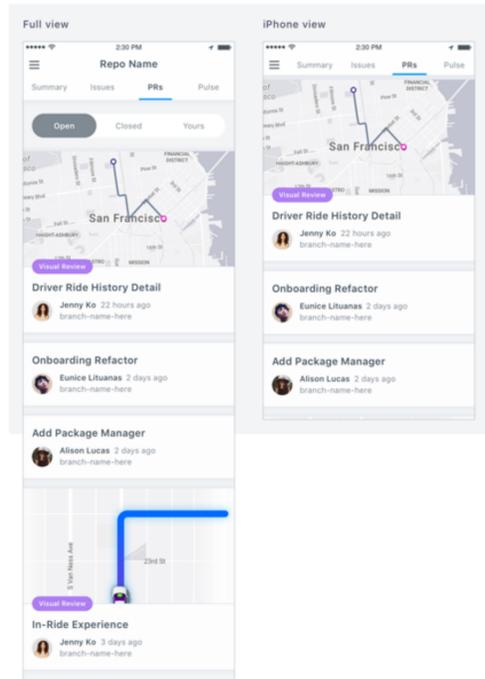
Pull Request



Visual Review



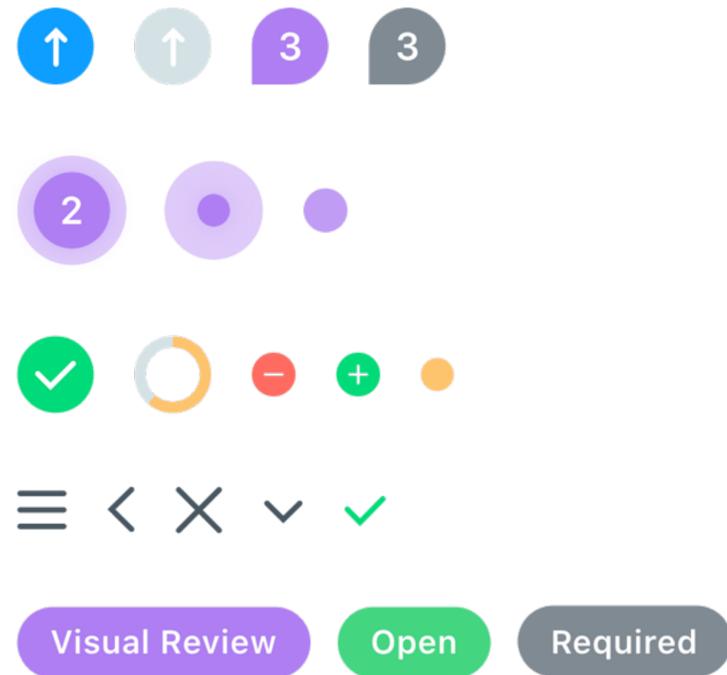
Repo Main View ("Reviews" Tab)



[View all mocks and interactions here →](#)

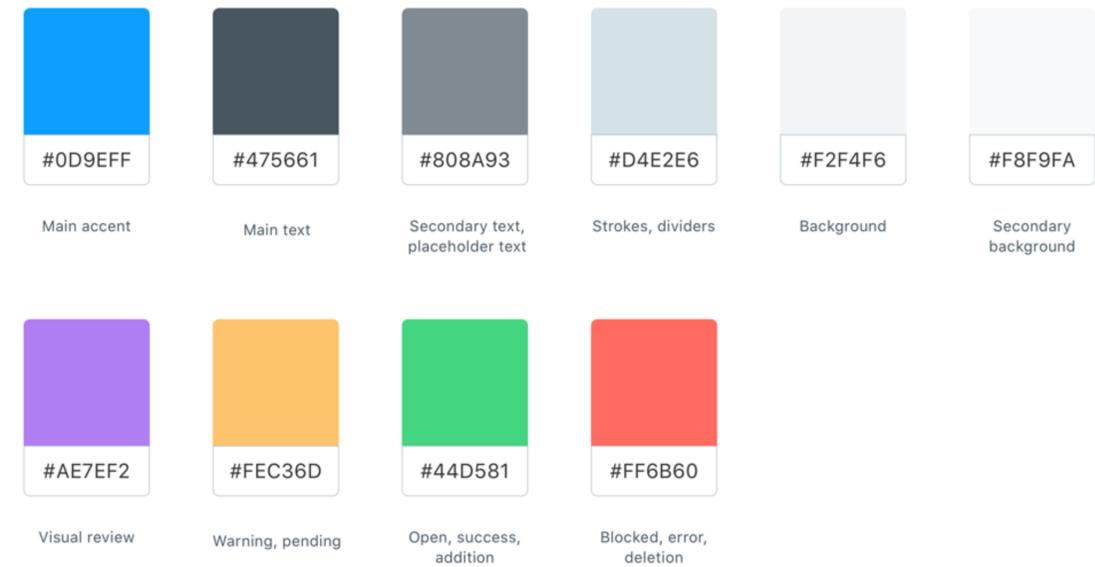
Visual Design / Style Guide

The final design includes an updated color palette, typography, and iconography.



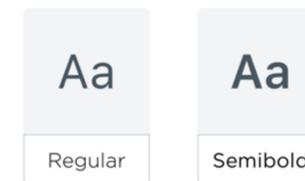
Style Guide

Color Palette



Typography

SF UI Text



Sizes @ 2x

Main headline

36pt (Semibold, 0.4)

Secondary headline

34pt (Semibold, 0.39)

Body copy body copy body
copy body copy body copy
body copy

32pt (Regular & Semibold, 0.38)

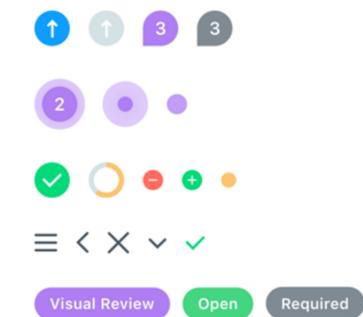
Subhead

28pt (Regular & Semibold, 0.36)

Minimum

24pt (Semibold, 0.36)

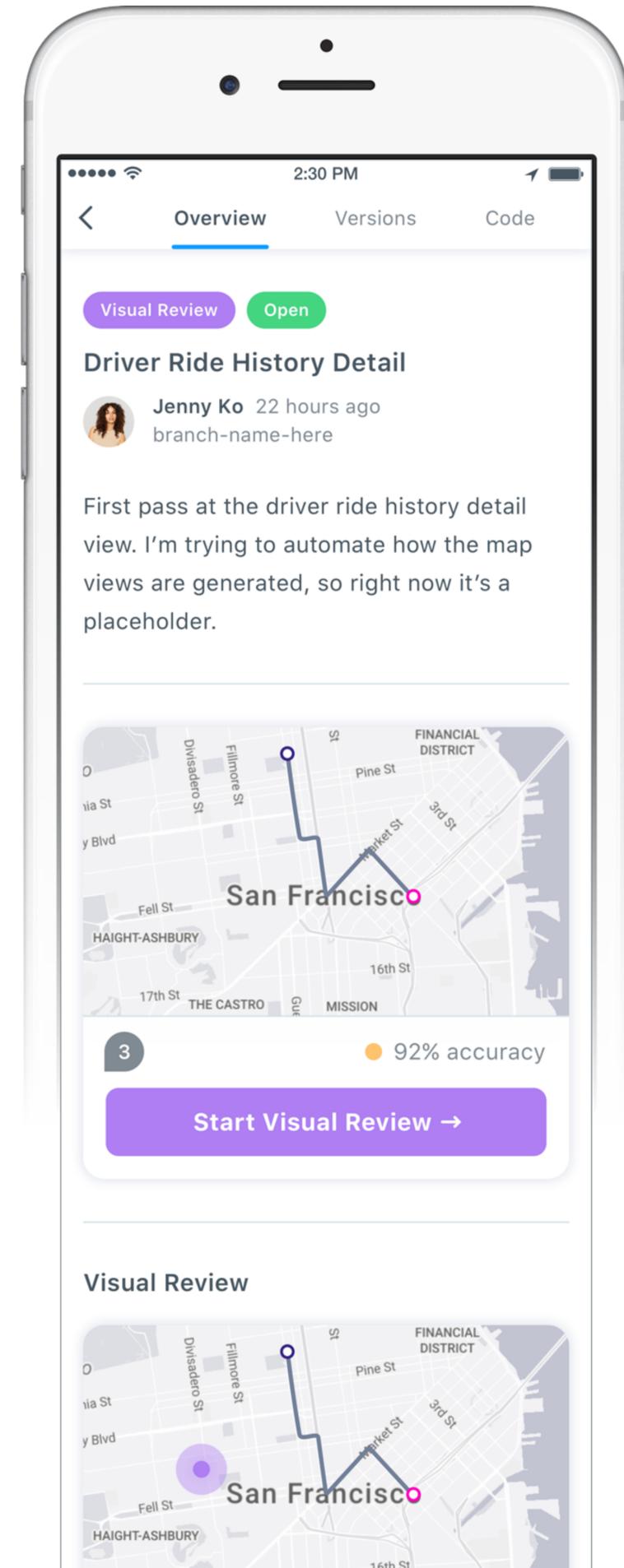
Iconography and Labels

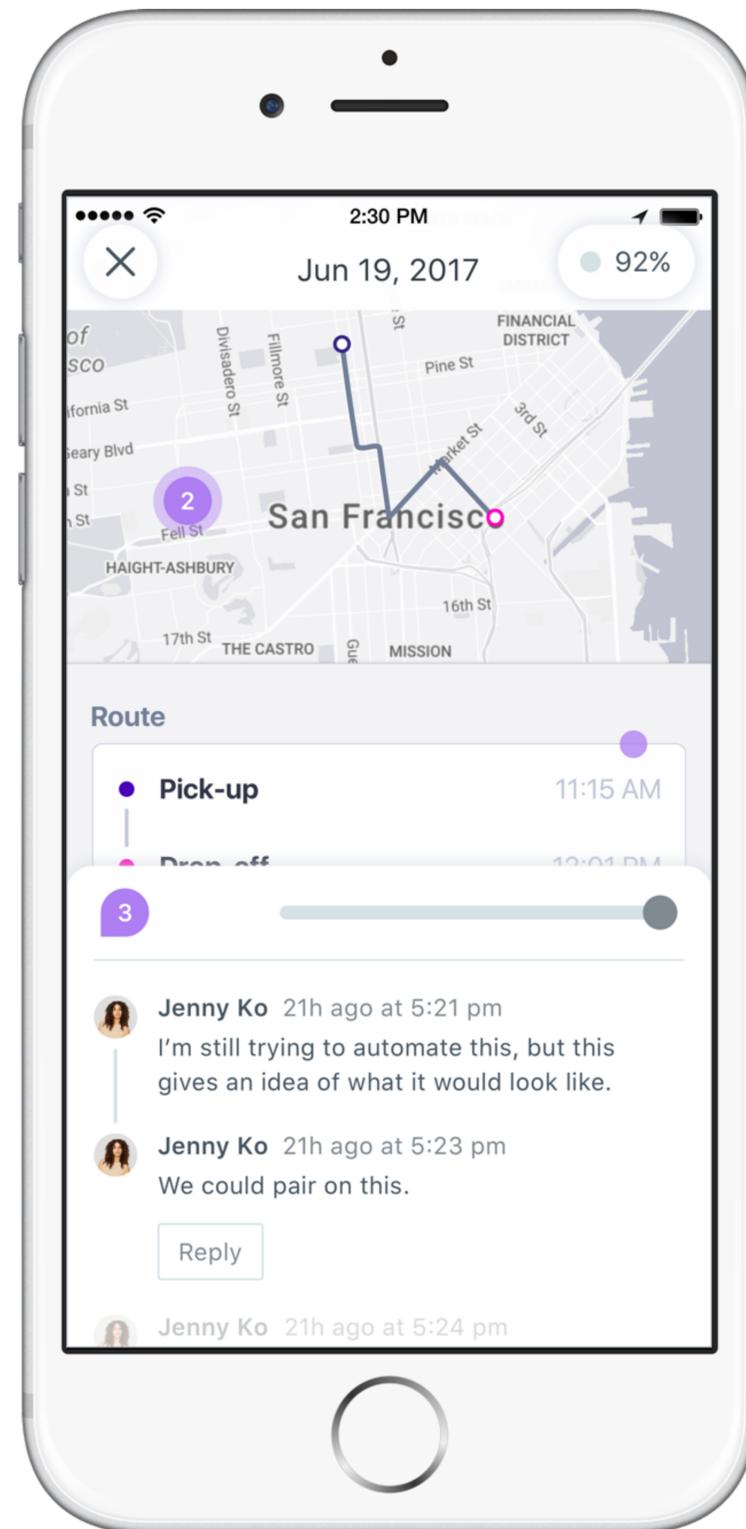


Final Design / Summary

The final Visual Review feature and updated PR flow improve product development work in the following ways:

- Normalizes design reviews later in project lifecycles by integrating design into the code review process.
- Provides better tools for communication between designers and developers.
- Makes it easy to preview mobile work on a device as well as provide feedback directly on the visuals.
- Increases the ability to catch design bugs by providing a visual comparison tool.
- Proposes a direction for future improvements that can automate the design QA process. Automated tools have the potential to reduce overhead for teams and increase the quality of design in final products.





All mocks and interactions →

User research overview →

Supplemental materials →

Eunice Lituanas Code Review
 Automated checks
 1 expected and 2 successful

visual-tool Review Complete
 92% accuracy comparison

code-review/reviewable
 0 of 0 LGTMs obtained
 Required

license/cia
 Contributor License is signed
 Required

No conflicts with base branch
 Merging can be done automatically

3 92% accuracy

[Start Visual Review →](#)

Jenny Ko 21h ago at 5:21 pm
 I'm still trying to automate this, but this gives an idea of what it would look like.

Jenny Ko 21h ago at 5:23 pm
 We could pair on this.

[Reply...](#)

Eunice Lituanas 3h ago at 11:25 am
 This is great, can we finally deprecate \$light-green?

[Reply...](#)

[Visual Review](#)

Driver Ride History Detail
 Jenny Ko 22 hours ago
 branch-name-here

Onboarding Refactor
 Eunice Lituanas 2 days ago
 branch-name-here

Requested reviewers
 Not required

Automated checks
 1 expected and 2 successful

No conflicts with base branch
 Merging can be done automatically

[Visual Review](#)

In-Ride Experience
 Jenny Ko 3 days ago
 branch-name-here

Suggested change

```

.route-map {
  - height: 406px;
  + height: 475px;
}
  
```

[Comment](#)

Code

Mock

Charlie Ogunse 1m ago at 2:30 pm
 Can we keep the route vertically centered?

[Reply...](#)

Route

- Pick-up 11:15 AM
- Drop-off 12:01 PM

Jenny Ko 21h ago at 5:24 pm
 This is a new border radius unit, should we add this to the style guide?

[Reply...](#)

Thank you