

ShopCrawl: Make Shopping a Party



ShopCrawl

Meet our team

Anthony R.



Co-term in Computer Science on the HCI track. Did my undergraduate in Economics. Born and raised in New York City.

Luke M.



Undergrad studying Computer Science on the HCI track. Member of the varsity Men's Volleyball team.

Isaac P.



Co-term in Computer Science. Senior in Economics. Grew up in New Jersey.

The Problem:

1. Chore shopping is boring and monotonous.
2. It's hard for people to meet new people.
3. Time is valuable.

The Solution:

Make shopping fun through social experiences.

↓ Our Value: Exploration

We aim to assist in helping younger adults begin to explore more. Given the nature of our product, we decided this should be our focus.

At a glance

- Explore new shop and products
- Make new shopping buddies by meeting like minded individuals in the area
- Rather than treating shopping as an isolated or purely transactional task, our design encourages users to discover, plan, and shop together

What features will express this value

We centered on three methods:

1. Community events
2. Crawl customization
3. Social features

Through these, we will can achieve our focus of exploration. In each, exploration exists: explore community events, explore new ways to crawl, and explore friends

Value Tensions

1. “In person” social app

Since we are an online app that centers on bringing people together offline, this very dynamic is a tension. We can aim to solve this through ensuring that our app is not a replacement for true social encounters, but the method in which these events occur.

2. Inclusivity

People may bring their already existing friends, and be less willing to meet new members from the community. This has the potential to breed exclusivity. We aim to minimize this by encouraging people to meet new other community members by setting their Crawls to public, where anyone can join.

3. Buy In

Exploration occurs only if members want to meet new people and want to try new places. We believe that by motivating people to try new places (through app-exclusive discounts for example) and meet new people is enough to overcome this tension.

Our Tasks

Simple: Join a Crawl.

The main use case of ShopCrawl is to join Crawls where you can go on social shopping experiences. Joining a Crawl will be the task most users do when they open the app.

Moderate: Create Crawls.

Once users are comfortable with participating in already created Crawls, they have the option to create their own Crawls. This feature is used by intermediate users who want to further contribute to the community.

Complex: Connect with people from past Crawls.

Most public crawls will include Crawlers that users have not met. Users who are ready to make a new friend and take their social experience outside of ShopCrawl can connect with Crawlers.

Usability Goals and Key Measurements

Goal 1: Engagement and Eagerness

Why: As a social-based app, users should feel motivated to participate in or return to future crawls after using the prototype.

Key Measurements: Observations of enthusiasm and engagement.
Ask users: “1-5, how interested are you in using ShopCrawl again?” Aim for > 4 average.

Goal 2: Efficiency and Ease of Use

Why: It's important that our app is easy to understand and accessible for different users to enable people from different backgrounds to participate

Key Measurements: Time spent on the last task from “Past Crawls”.
Ask users: “1-5, how easy was it to complete the last task?” Aim for > 4 average.

How our product is hitting these goals

Our product effectively meets our usability goals by keeping navigation intuitive and visually consistent, reducing friction for users as they move through each task. The social and interactive elements, like viewing friends and adding crawlers, encourage engagement and make the experience feel active and community-driven. It also hits the efficiency and ease-of-use goal by keeping navigation simple and consistent enabling users to move between screens with minimal steps, clear feedback, and intuitive button placement that reduces decision time. Overall, the design balances clarity and enjoyment, promoting both ease of use and eagerness to return.

Next Changes to hit goals

To better hit these goals we plan to implement the following:

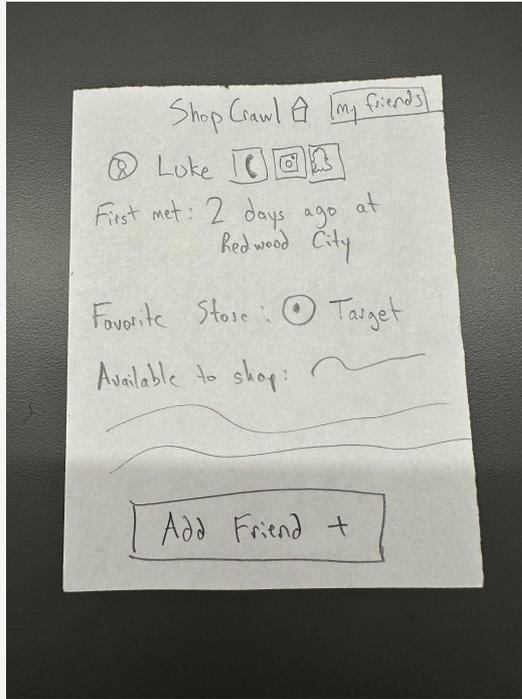
1. **Expand the metrics we collect beyond final task time.** By tracking misclicks, backtracks, and hesitation points, we can better identify friction in the flow and quantify improvements in efficiency over time.
2. **Systematically address high-severity incidents.** View systemic reports that users are sending to be able to identify problems before they arise
3. **Improve visual hierarchy and consistency.** By constantly refining component spacing, color contrast, and typography will help users navigate faster and with fewer errors, directly raising “Ease of Use” ratings above our >4 goal.

Core Change 1: Report Button

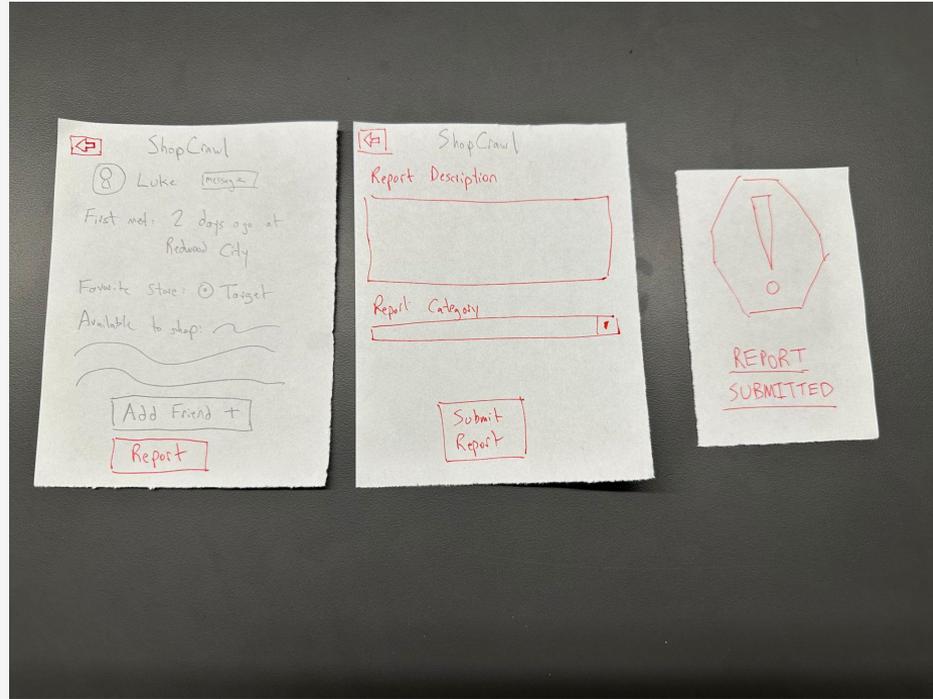
Core 1st change: Users flagged safety concerns. wanting a way to report or block others before connecting. Added a visible “Report User” option on profiles and after crawls, and added post-crawl rating/review screens to build trust.

How to affects our goal and measurement: This boosts willingness to use again by removing the biggest bottleneck. This helps to increase overall engagement and eagerness

Core Change 1: Report Button



Original Crawler Profile



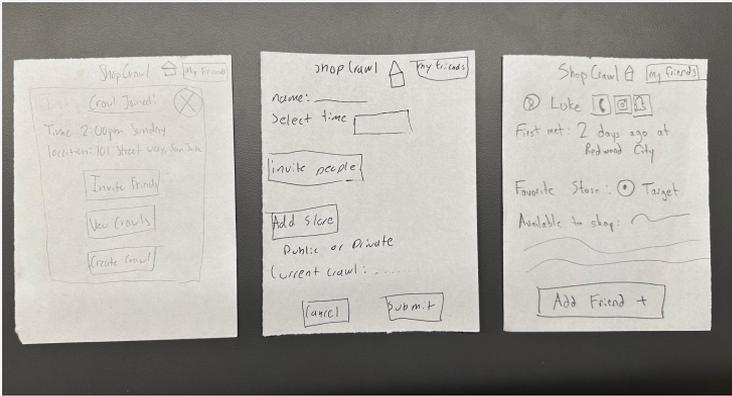
After Change: Added report button, report description page, and report submission confirmation.

Core change 2: User Confirmation

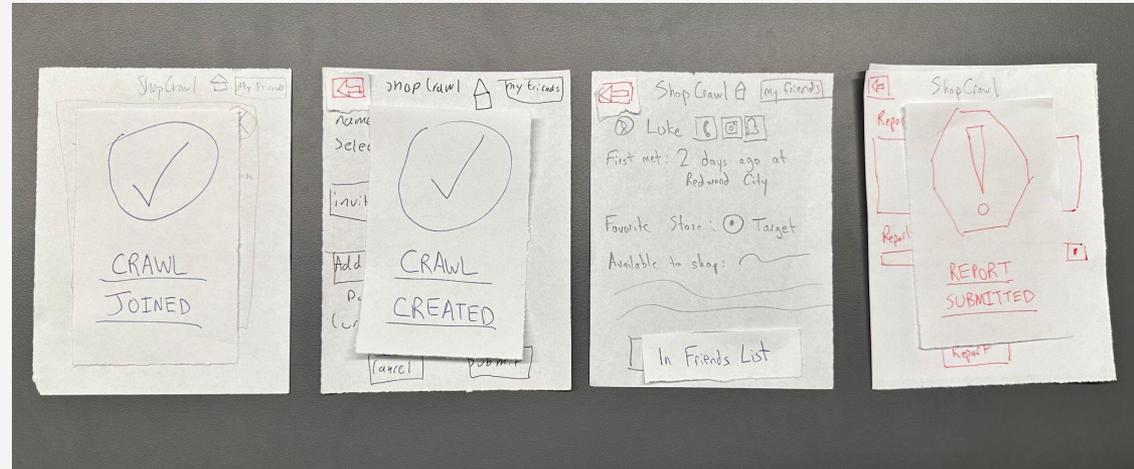
Core 2st change: Feedback from testing revealed that users were often unsure whether their actions, such as reporting a user, or confirming a crawl, had gone through successfully. To address this, we added confirmation screens and status alerts that clearly show when an action is completed (e.g., “Report Submitted”). These small but visible cues build user confidence and make the flow feel more responsive and intentional.

How to affects our goal and measurement: Providing clear confirmations directly improves Efficiency and Ease of Use by reducing user hesitation and preventing repeated actions. It also enhances Engagement, since users now receive immediate visual feedback for every completed step.

Core change 2: User Confirmation



Original Pages with Action Items



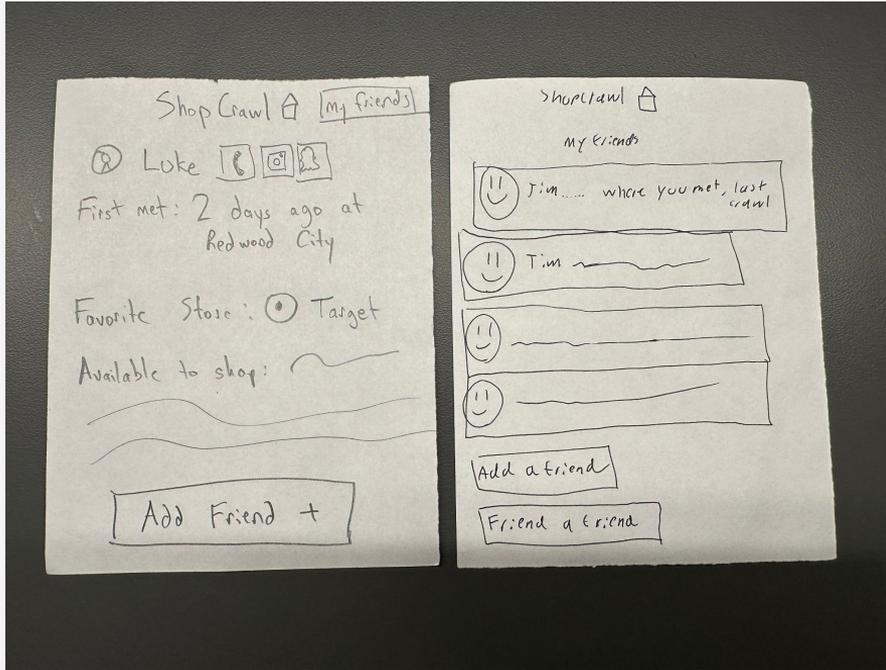
After Change: Added confirmation popups for Crawl Joined, Crawl Created, and Report Submitted. Added a visual change in the profile after you add a friend, letting you know that they are now in your friends list.

Core change 3: Standardized Layout & Hierarchy

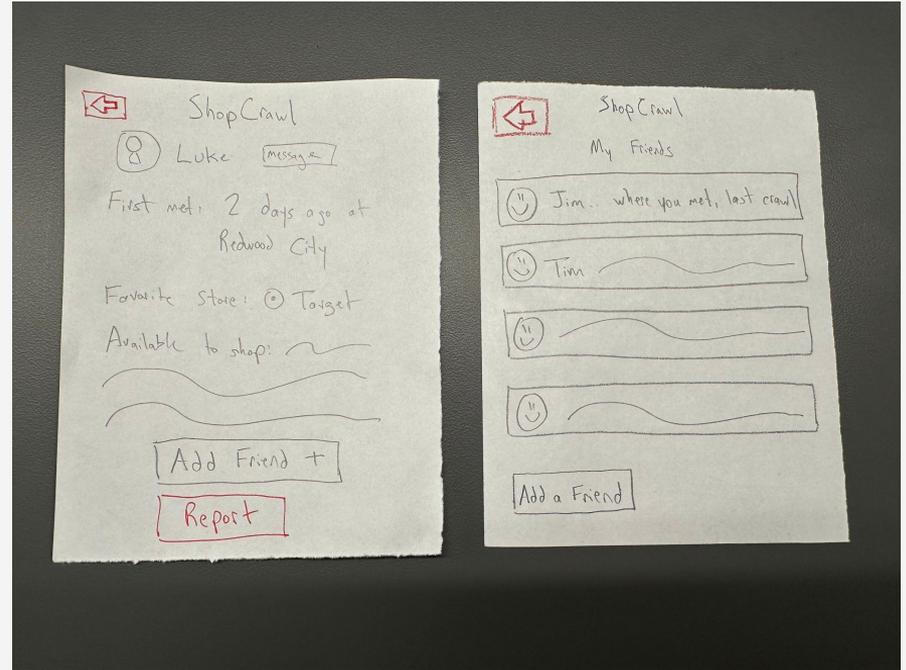
Core 3rd change: Feedback from testing and the TA noted inconsistencies in spacing, sizing, and alignment across screens, which affected the app's clarity. In our revised sketch, we standardized the layout and hierarchy which included aligning components, unifying text sizes, and maintaining consistent button placement across pages. These refinements improve scannability and make the overall interface feel more cohesive and predictable.

How to affects our goal and measurement: By improving visual structure, users can now process information faster and move through tasks with fewer misclicks or hesitations. This directly strengthens our Efficiency and Ease of Use goal, helping us achieve shorter task times and higher "ease of use" ratings (aiming for >4 average).

Core change 3: Standardized Layout & Hierarchy



Original Pages with inconsistent layout



After Changes: consistent placement of back button, aligned spacing and sizing of UI elements.

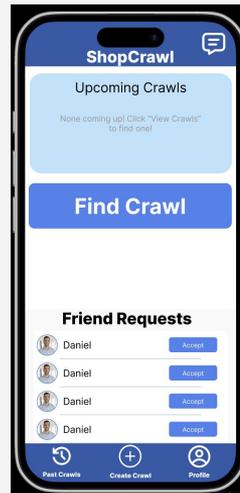
Simple Task flow:

Our simple task is centered around how to join a crawl. Effectively we want to enable users to view available crawls, see where each crawl is going / people on it / stores on it, and then have the ability to join the crawl

Thus our task works as follows

1. Load into the home screen
2. Hit the find crawl button to open available crawls
3. Filter button for the crawls
4. Click on a crawl to see more information
 - a. Click on the people and stores
5. Then click on the join crawl once you find one

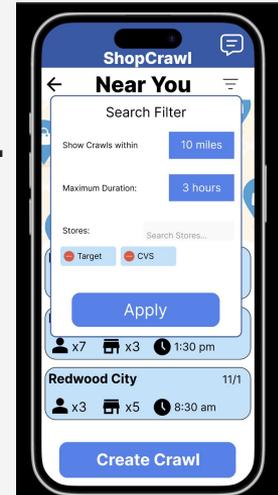
1.



2.



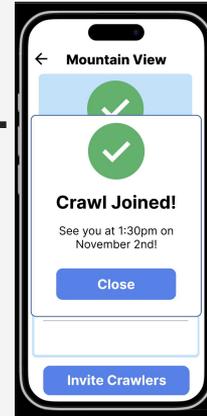
3.



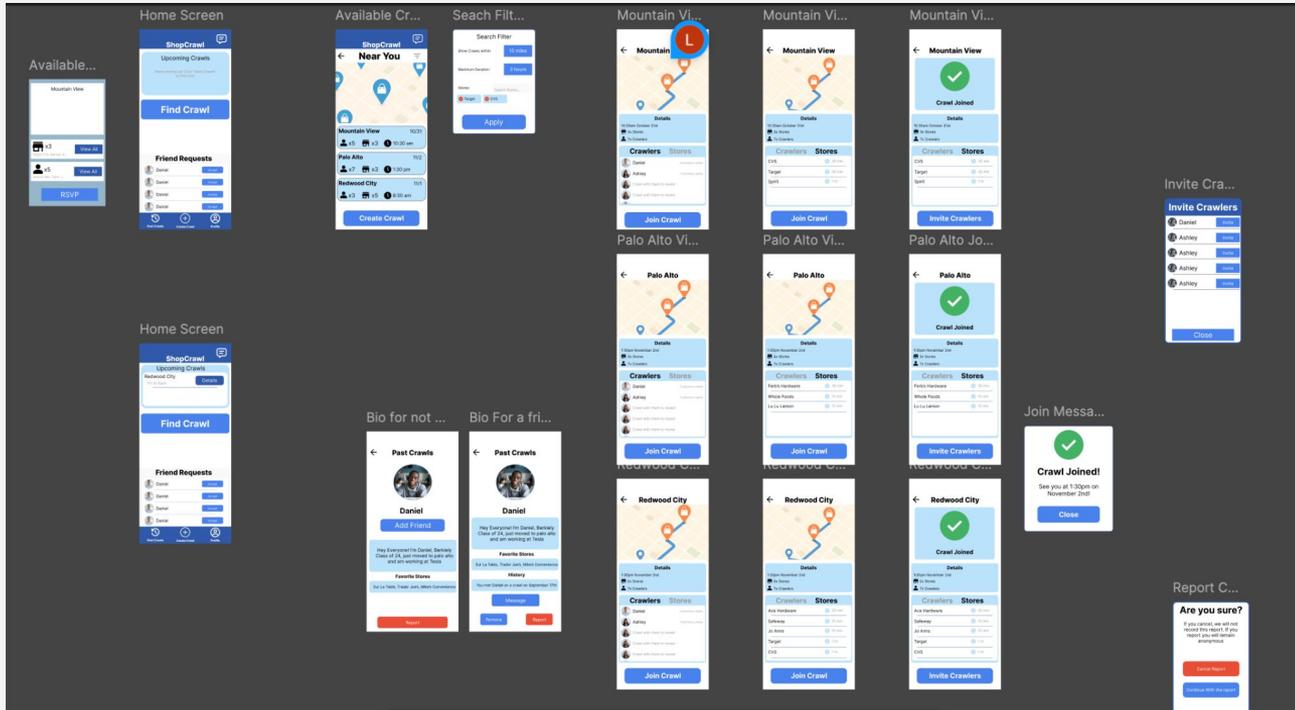
4.



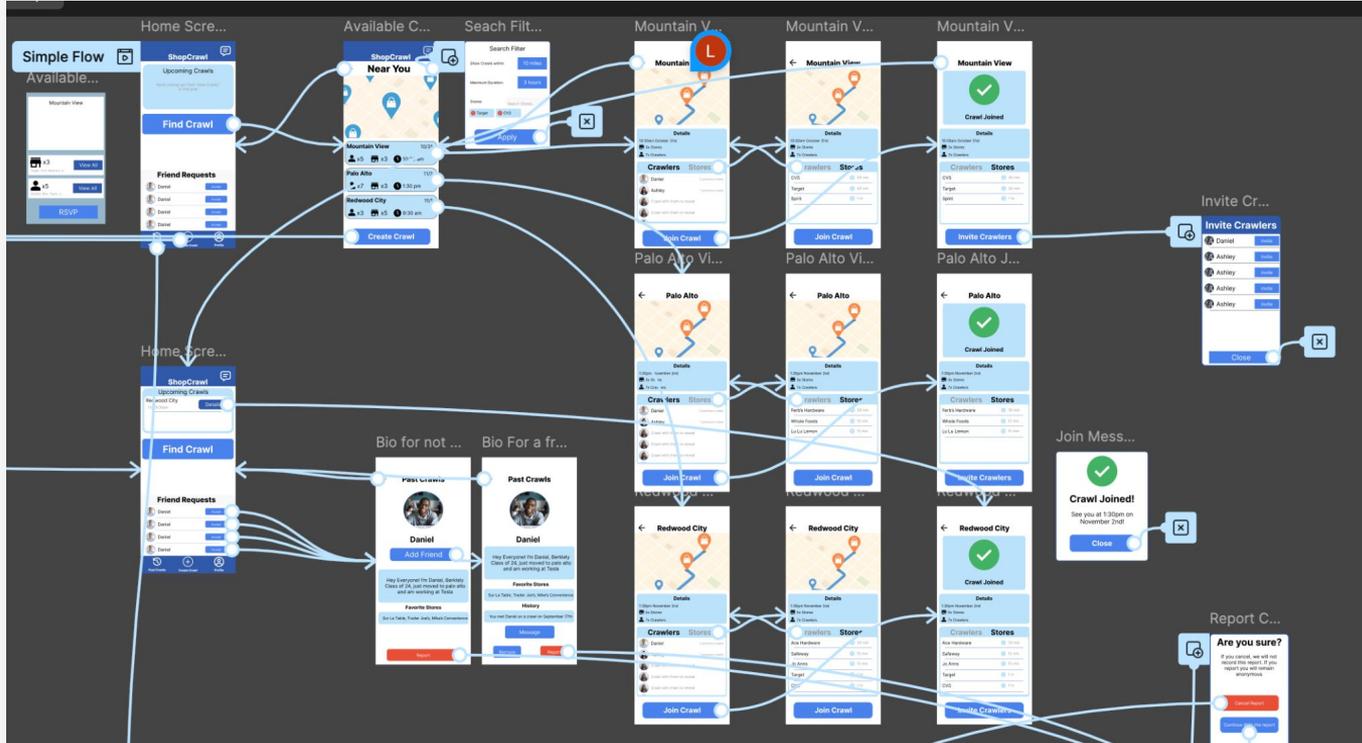
5.



Simple Task All Screens Medium Fi:



Simple Task flow all screens with Flow Direction



Medium Task flow part 1:

Our medium task is centered around how to create a crawl. We want users to seamlessly create a crawl by selecting a time, adding stores, inviting friends, and confirming the event.

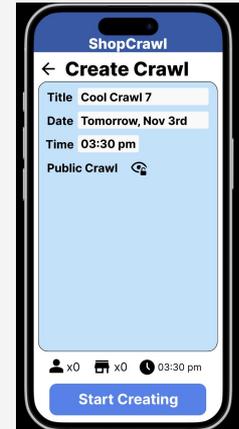
Thus our task works as follows

1. Load into the home screen
2. Fill in crawl information
3. Click to toggle between private and public crawls

1.



2.



3.



Medium Task flow part 2:

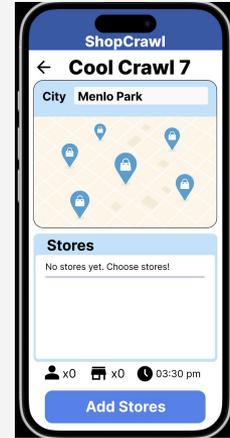
This part is centered around how to add a store

1. Load into the add store main screen
2. Search for the city you want to crawl
3. Click on the map for more details
4. Add a time
5. Add the store
6. Click to add more stores each and then once finished click the button on the bottom to move onto the add crawlers section

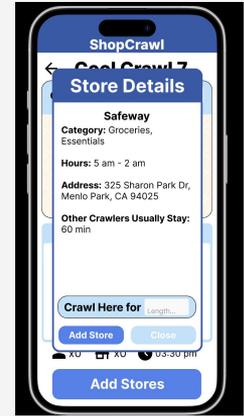
1.



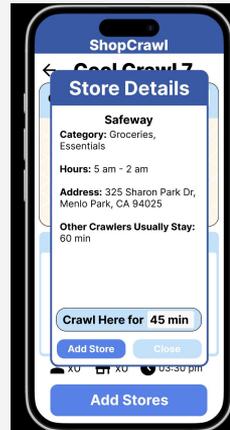
2.



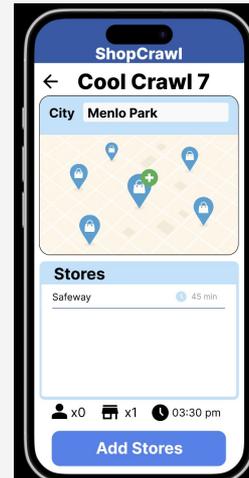
3.



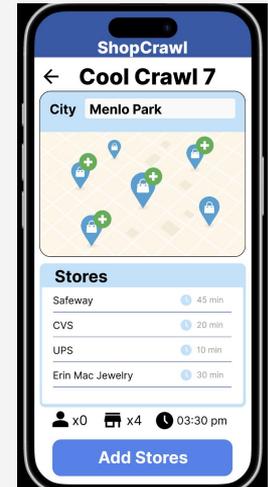
4.



5.



6.

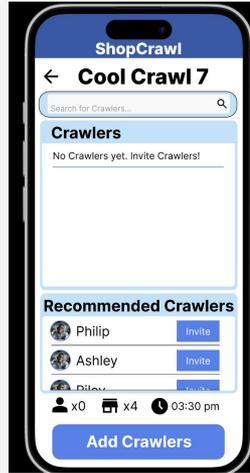


Medium Task flow part 3:

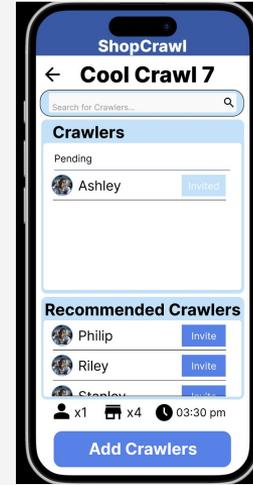
This part is centered around how to add a crawler

1. Load into the add crawlers main screen
2. Invite a recommended crawler
3. Search for crawlers
 - a. Add the person you searched for
4. Once finished, click the add crawlers to get to the final screen
5. Click to confirm the crawl, opens the crawl confirmed page

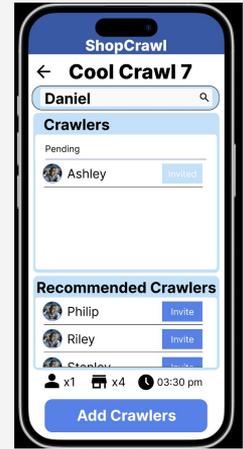
1.



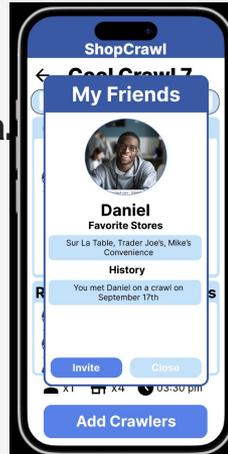
2.



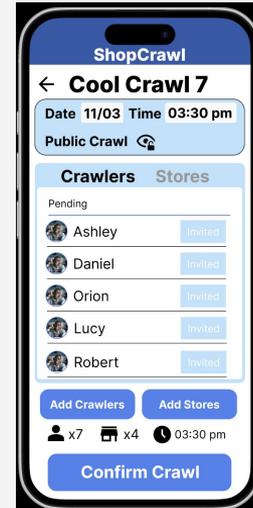
3.



3a.



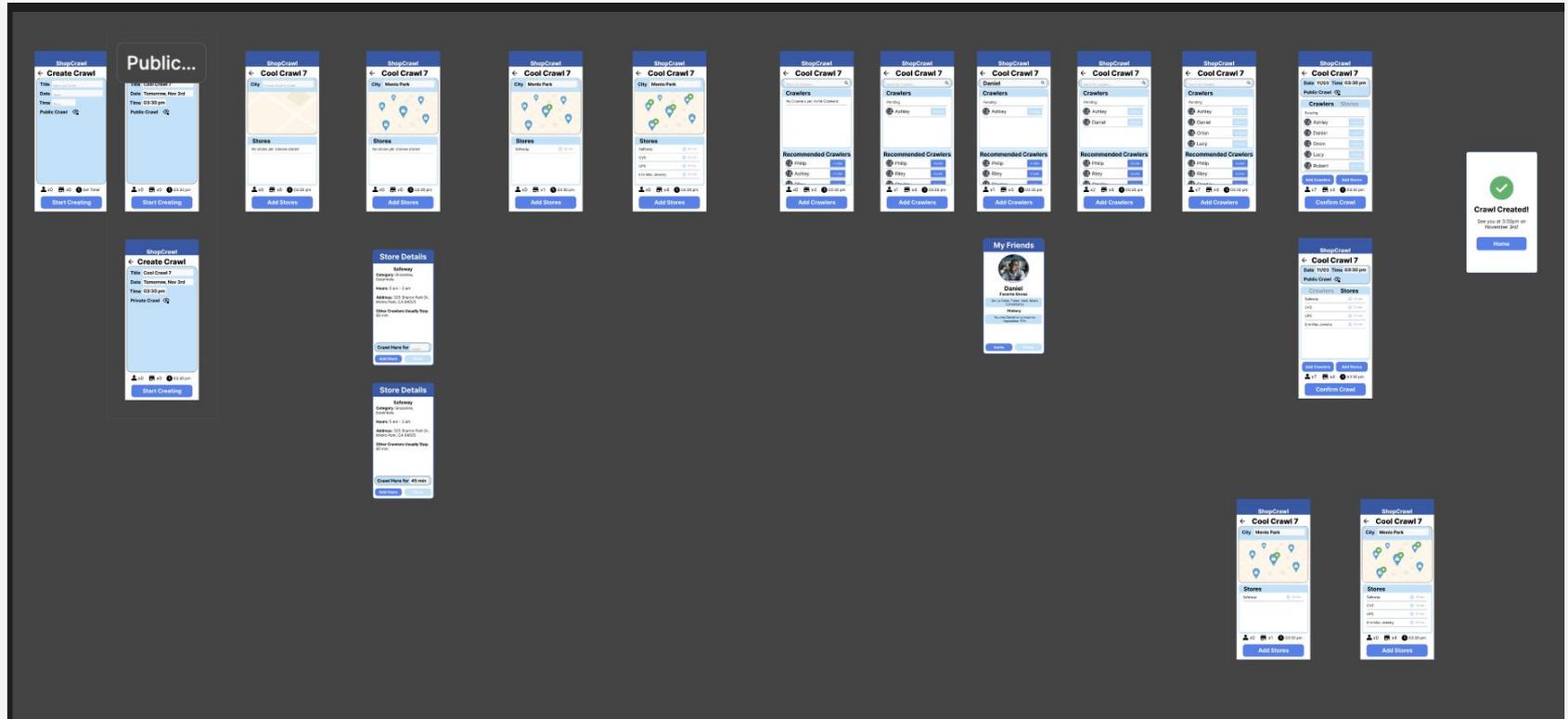
4.



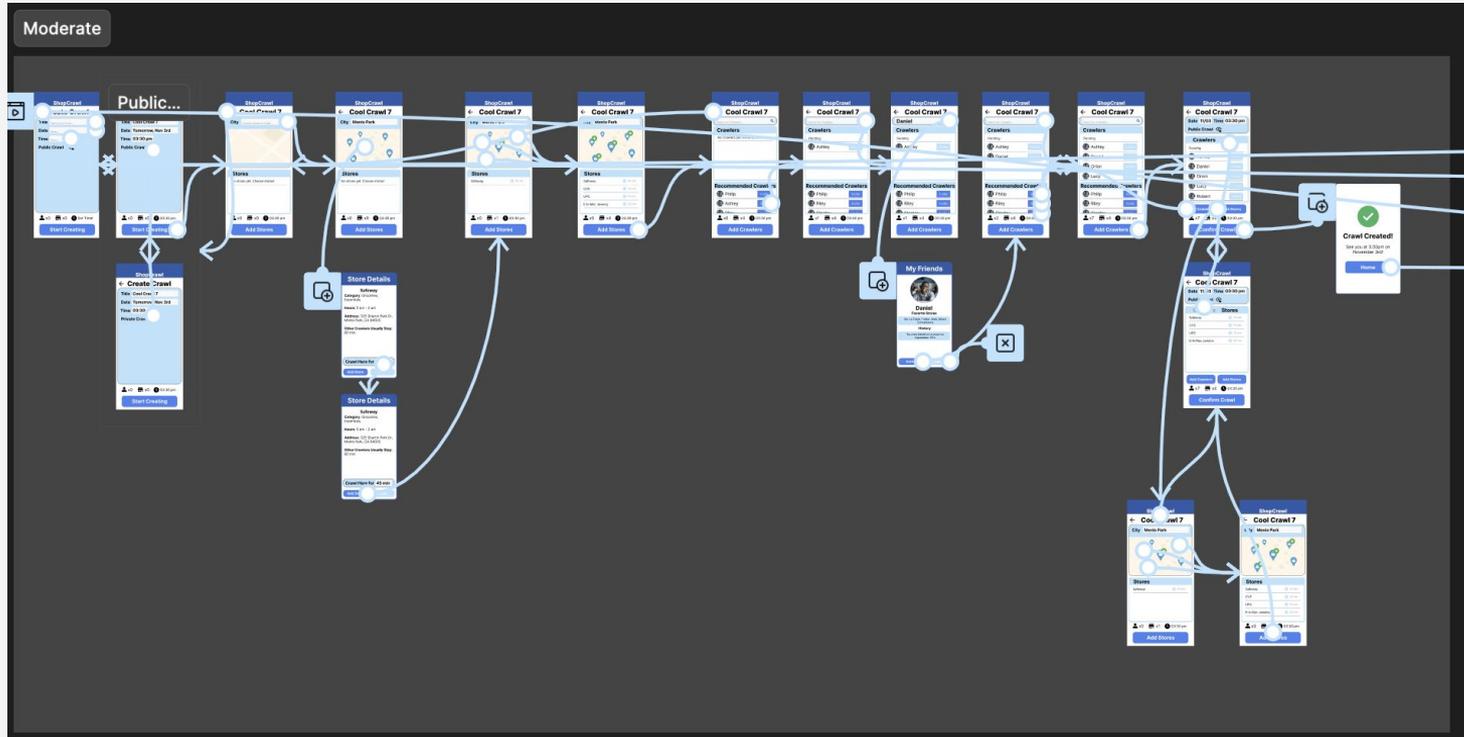
5.



Moderate Task All screens



Moderate Task All Screens with Flow Direction



Complex Task flow part 1:

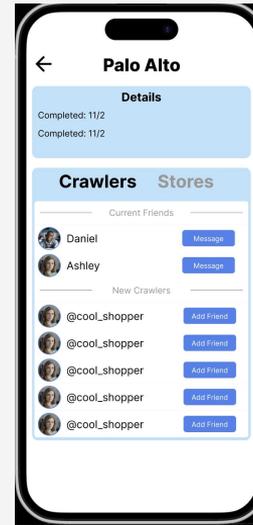
Our complex task is centered around connecting with people from past crawls. Thus we focus on how we can make it easy for you to find previous crawlers, message and stay in contact.

1. Past crawls home page. See all the past crawls
2. Click on a past crawl to get to the information. See the people in the crawl, who of them you are friends with currently
3. Click on a person to bring up their bio. Can message, report, or add / remove friend
4. Message opens up messaging app
5. Back on messages opens up history of messages

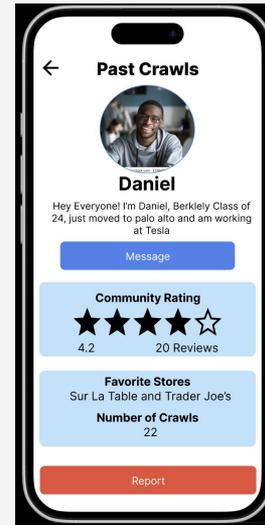
1.



2.



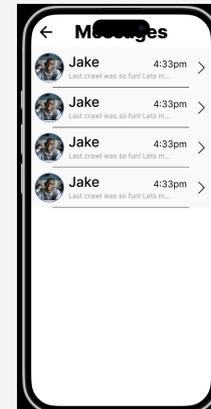
3.



4.



5.

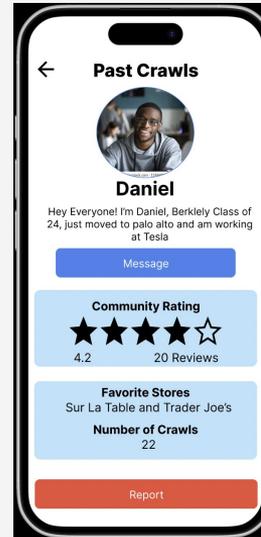


Complex Task flow part 2

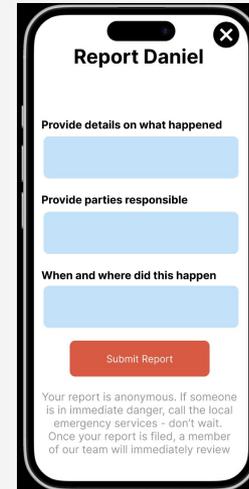
A core goal of ours with this new prototype was to increase the safety, thus adding reporting fields was important and we implemented as follows

1. Click on the report button on a person's bio
2. A report button appears. This asks the user to fill the details. It also emphasizes that the report is anonymous and that if there is an emergency to call local services.
3. If you want to cancel the report, it asks if you are sure to make sure that someone is actively trying to not report the person rather than accidentally cancelling

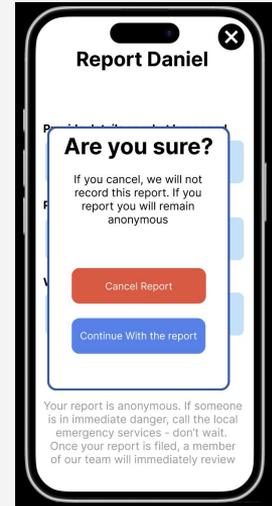
1.



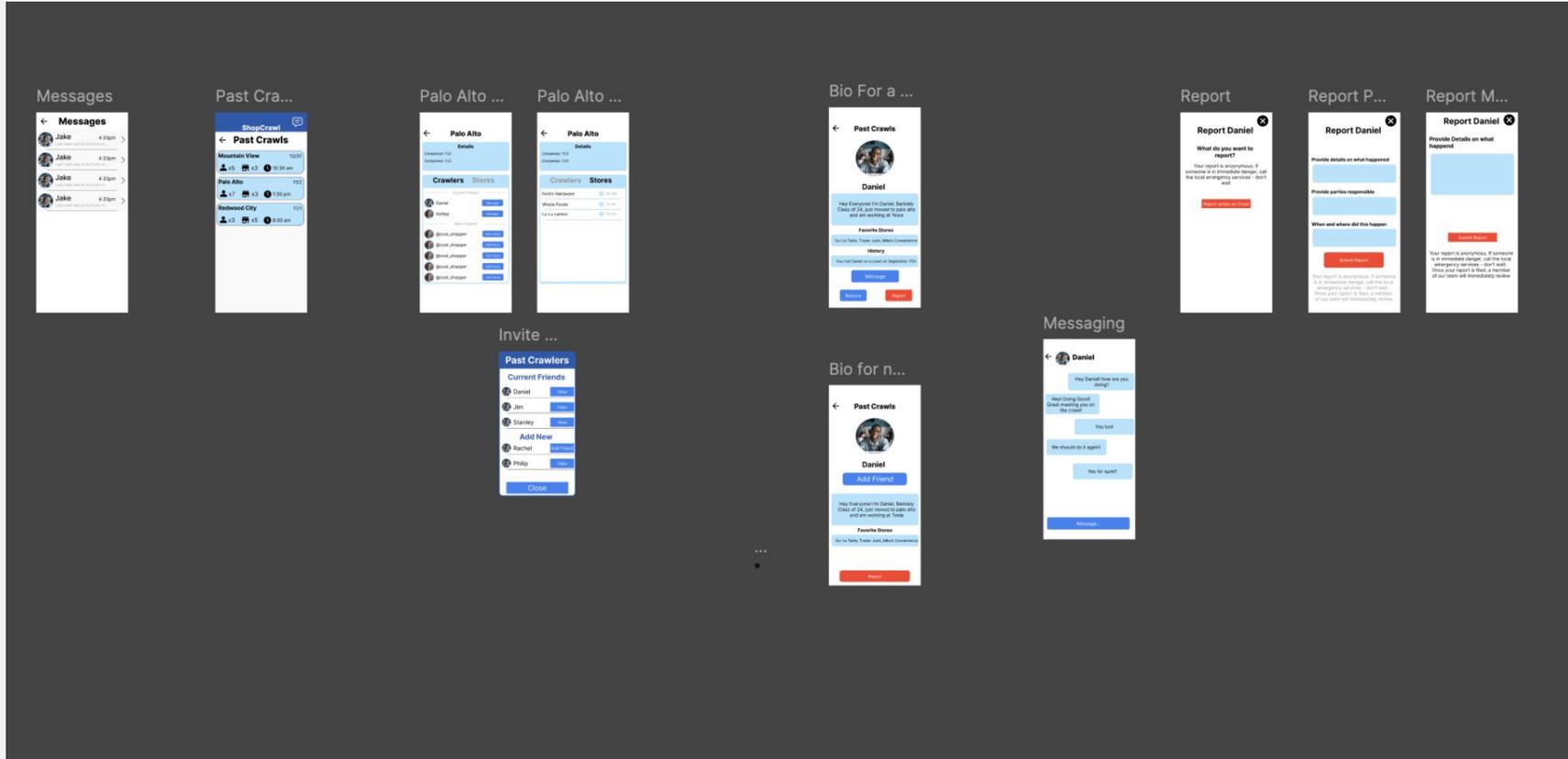
2.



3.

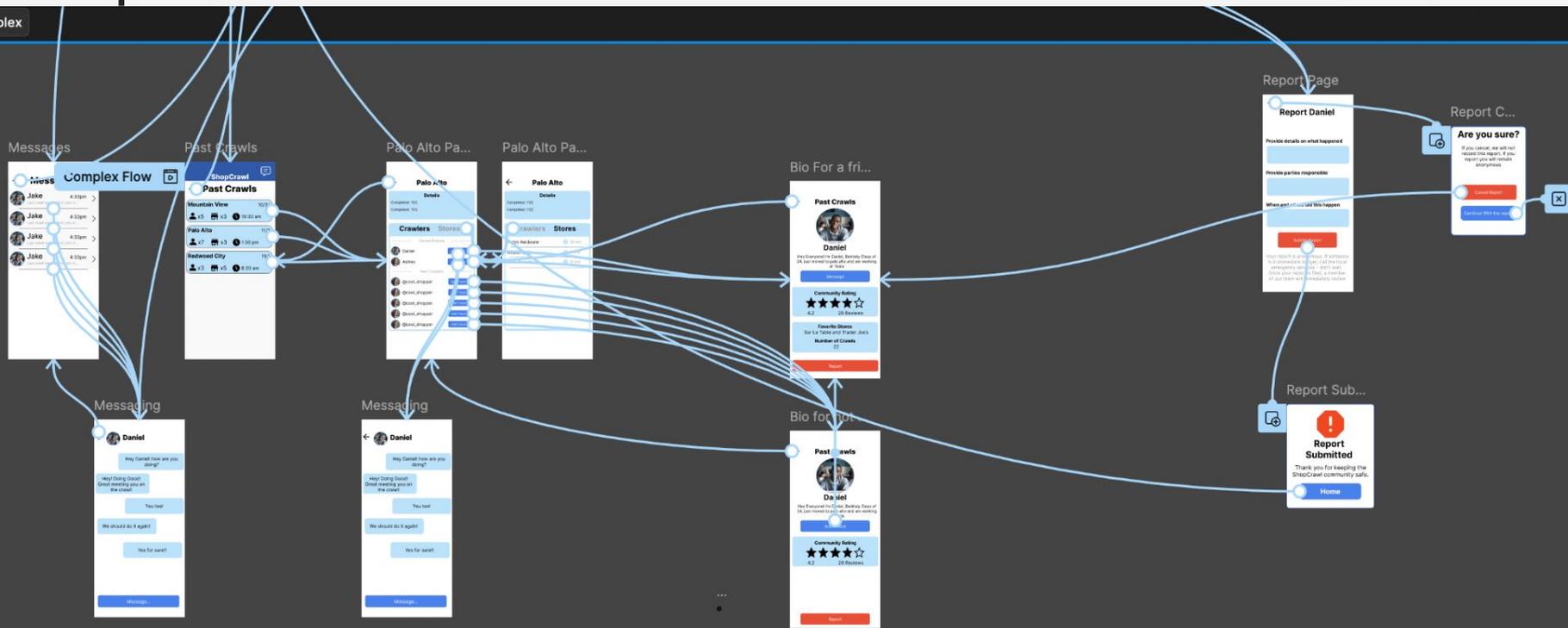


Complex Task All screens



Complex Task All screens with Flow Direction

Complex



Tools: Pros and Cons of Figma

We decided to use Figma after a careful debate. To start here are some of the pros and cons that we debated between:

Pros:

- Very quick to build and iterate on complex designs
- Allows for high-fidelity visuals that closely resemble a real product
- Easy to organize and test multiple flows in one shared workspace
- Enables collaborative editing and feedback in real time

Cons:

- Difficult to test interactions on a phone natively; most testing happens on desktop, which reduces realism in mobile experiences
- Limited ability to handle true user input (e.g., typing, dynamic text, conditional behavior)
- Simulated transitions can appear smooth, but they don't represent actual system performance or data flow
- Can become cluttered with many overlapping prototype connections, making edits slower as the design grows

From this, we decided that the pros of speed and flexibility to building these complex screens outweigh the limitations of fully interactive user input

Core Limitations

The following are the key limitations that we ran still have in our prototype:

- Limited user input flexibility: Most fields and buttons are preset or static and users can't truly customize inputs like adding specific stores or crawlers.
- Hard-coded flow: Several buttons and screens are purely illustrative; some clickable areas don't lead anywhere, serving only to show layout and intended navigation
- Low UI design: We aim to make the design more fun and active at a future time frame, right now is rather basic and not as enjoyable

Why these limitations? We wanted to focus on concept validation. We prioritized testing the overall flow and design direction before implementing detailed interactivity or data-driven features.

These limitations have an impact on having a true social experience for users of the prototype, since they cannot connect to real life Crawlers, or search for real nearby stores. The prototype still succeeds in allowing users to explore the core concept of ShopCrawl, and to learn the UI before we solve these limitations in later prototypes.

Hard-coded and Wizard-of-Oz features

Hard coded:

- Several buttons and text fields are non-functional placeholders used to demonstrate layout and navigation (the messages that exist in the messaging screen, can't send more in)
- Profiles, store listings, and crawler cards are pre-loaded visuals as they can't be edited, searched, or customized by the user.
- Navigation links (like back buttons or confirmation actions) simply move to preset screens rather than processing any real data.

Wizard of Oz:

- The “Create Crawl” page auto-fills fake text when clicked, imitating user input.
- The “Confirm Crawl” button brings up a pre-made success screen, mimicking a completed process.
- The “Report Submission” screen instantly displays a message, faking a server response.

Link to figma:https:

//www.figma.com/proto/oDg452S4xNPsIPNNnZN7NA/Shop-Crawl-Low-fi-Prototype?node-id=0-1&t=hXtQnIGvKkcDqFo5-1