



Reclaim

Final Report

CS147 Winter 2021
Sustainability

Miles M., Igor B., Melissa W., Disney V.

Table of Contents

Project Name and Value Proposition	3
Team Member Names	3
Problem and Solution Overview	3
Introduction	3-4
Needfinding Interviews	4-5
POVs & Experience Prototypes	6-9
Design Evolution	9-25
Final Prototype Implementation	25-29
Summary & Next Steps	29-31
Appendix	32-38



Project Name and Value Proposition

Project Name

Reclaim

Value Proposition

Everything is for sale.

Team Reclaim

- Miles M - Bit Manipulator
- Igor B - Head of Semicolons
- Melissa W - Digital Dynamo
- Disney V - Chief Inspiration Officer

Problem and Solution Overview

Problem

You want to buy an item used, but finding someone nearby who's selling what you need is hard—even if plenty of people would be willing to sell if asked.

Solution

Reclaim is an app-based reverse marketplace where buyers post what they need and anyone nearby can fulfill their request. Reclaim unlocks hidden supply for buyers and reduces the burden on sellers to manually list everything with which they're willing to part. Our mission is to make buying used easier than buying new.

Introduction

We are a group of college students passionate about environmental entrepreneurship. Ten weeks ago, we started the quarter eager to design a sustainability-related app and embarked on a quarter-long design journey in CS147 with this goal in mind. We began with needfinding interviews to better understand our users and their needs within the sustainability space. We brainstormed potential solutions to meet these needs and then created experience prototypes for our top three solutions and tested them with real people. Based on their feedback, we selected a concept for our application and iterated through application designs using a low-fi prototype, medium-fi prototype, and high-fi prototype. During this process, we also tested our application on users and received a thorough critique from some of our expert classmates regarding the usability of our application.

We invite you to walk through some of this journey with us today. Join us to learn what inspired our concept, how we reached our final design, and the testing we performed to achieve an optimal user experience. We now go back 10 weeks to begin our design journey with you.

Needfinding Interviews

Methodology

Our design process started with 15 sustainability-related needfinding interviews to help us to better understand the needs of our users and how we might be able to help them. We selected a diverse range of participants, being careful to include individuals of different ages and professions. We also selected individuals with varying levels of interest in sustainability as we wanted our final application to be helpful for sustainability activists and novices alike. Some of our interviews were with “extreme users” such as a dumpster diver, an eco-activist, a molecular coffee maker, and a waste engineer. We recruited our participants via Reddit and Twitter, as well as through friends. To protect the safety of ourselves and our interviewees, we conducted our interviews remotely via Zoom. To protect our participants’ rights and anonymity, each participant signed a consent form prior to the start of the interview.

We asked our participants some basic biographical questions followed by a range of questions about sustainability. Though we had a script, we let the interviews flow naturally: we were particularly interested in interviewees’ personal stories, emotions, and thoughts regarding sustainability. A small sampling of prompts from our interview guide is included below:

- Tell me about a time when you felt wasteful. How could that have been prevented?
- Tell me about a time that you felt ‘inspired’ about sustainability.
- What makes you personally invested in sustainability?
- What kinds of disposable products do you use?
- Tell me about a part of your job/daily routine that you think is problematic from a sustainability perspective.

Insights

We started to digest the content of our interviews by making 3 empathy maps (see Appendix 1 for an example empathy map, pg. 32). These maps helped us step into our participant’s shoes and better understand their thoughts, actions, and emotions regarding sustainability. While we were creating our empathy maps and reviewing the notes from our interviews, four key insights emerged.

First, we noticed that our participants purchase and use sustainable goods when it is convenient for them. One of our participants, a college student named Nadiia, said that “I don’t want to be [an environmental] martyr... I’m not going to make my life unnecessarily difficult.” We realized that in order to

help users like Nadiia live sustainably, we need to make it easier for our consumers to buy used than it is to buy new; we could not sacrifice convenience.

Second, we learned that we should not expect consumers to pay more for sustainable goods when cheaper alternatives are available. Andy, the sustainable molecular coffee maker, shared that “I don’t think Atomo! (his company) would have succeeded if the price had been higher than the price of normal coffee.” This common theme made us realize that our consumers need competitive, cost-effective, sustainable alternatives.

Third, we discovered that our participants had difficulty finding the used goods they were looking for locally, and that difficulty locating items sometimes led to our participants asking friends or family members for some types of goods. For example, a college-aged participant Arzy explained that “I don’t give up when searching for certain items... I [sometimes] ask to borrow [them] from friends.” We realized that Arzy—along with many of our other participants—needed help locating the used products they are looking for locally.

Finally, Dan the dumpster diver explained that many items that are considered “waste”—food and goods alike—are actually perfectly fine to use. Dan shared that grocery stores often throw out items long before their expiration date to make room on the shelves for newer goods when they arrive. Dan, who is a nurse and has never gotten foodborne illness after a lifetime of dumpster diving, shared that “Most items haven’t reached their expiration date, and we as humans can identify what’s good and bad.” Dan made us realize that people need better visibility into items’ durability and shelf life. All four of these findings inspired and informed our thinking in the next steps of the design process.



Figure 1: Photos of one of Dan's dumpster dives.

POVs & Experience Prototypes

In the next stage of our design process, we digested our three most insightful interviews and brainstormed possible solutions. For each individual, we started by generating a Point of View statement (POV). POV statements are a concise reframing of a problem that is grounded in user needs and insights. POVs involve three parts: who we met, something new that we were amazed to realize, and a synthesized insight that we might be able to leverage in designing a solution (what we call a game-changer).

We then used each of our “game-changing” statements to brainstorm 10-15 How Might We (HMW) statements. How Might We statements are short questions that aid in the brainstorming process and often answer at least 3 of the 5 Ws (who, what, when, where and why). They helped us dig deeper into different parts of our POV game-changing statements and slowly work towards a solution. A sampling of these statements is included below for each of our three participants; we selected our most promising HMW statement for each of our participants and used it to generate a potential solution. We then tested an assumption for each of our solutions using an experience prototype. Below you can see this brainstorming process, our potential solutions, and the corresponding experience prototype for each of our three users.

Point of View for Michelle

- **We met Michelle**, a Hong Kong student who goes to college in the United States.
- **We were amazed to realize** that Michelle thinks shipping used items is “weird,” both at home and abroad: Hong Kong is so small that local pickups are the norm.
- **It would be game-changing** to bring Hong Kong’s hyper-local reselling model to other parts of the world.

How Might We Statements Inspired by Michelle

- HMW encourage local pickups and exchanges?
- HMW make face-to-face exchanges more convenient?
- HMW make local pickups faster and less time consuming than shipping?
- HMW build well-stocked local trading economies?
- HMW adjust expectations so that local pickups are the default?
- HMW make buying items from far away unnecessary?

Solution and Experience Prototype Inspired by Michelle

Michelle and our desire to make buying items from far away unnecessary inspired our first potential solution. We came up with the idea to have a reverse marketplace for college campuses where buyers post what they need and potential sellers respond. We then created an experience prototype to explore this potential solution further. The assumption we wanted to evaluate was whether people would be willing to sell certain items if someone wanted to buy them, even if they weren’t originally looking to

sell those items.

To test this assumption, we set up a group text between a seller and a fake buyer. Prior to the experience prototype simulation, Miles—who played the role of buyer—posted that he needed a copy of a textbook for one of his classes, and Grace responded because she had a copy. During the simulation, Miles and Grace were connected via text message to negotiate. Grace ended up selling her copy of the textbook to Miles.

Reflecting on the prototype, Grace shared that “I think it’s a mutually beneficial situation — potentially more so than in a seller-centered marketplace. Knowing that I’m selling to someone who needs it and will make good use of it — there’s something satisfactory about it.” We found that this prototype generally worked quite well and that our assumption was validated: Grace had items she was willing to sell, even though she wasn’t actively looking for a buyer. The main issue we identified while running this experience prototype was that some items have sentimental value, and users may not want to sell these items even if they are not actively using them.

Point of View for Olivia

- **We met** Olivia, a Chinese-American college student who lives in Chicago and is passionate about fashion.
- **We were amazed to realize** that despite her interest in fashion and buying second-hand, she avoids thrifting because her parents disapprove of it—they view it as unnecessary and dirty, especially considering that they can buy her new clothes.
- **It would be game-changing to** destigmatize thrifting in the eyes of her parents.

How Might We Statements Inspired by Olivia

- HMW make thrifting cleaner?
- HMW get parents involved in thrifting themselves?
- HMW make thrifting more trendy?
- HMW make thrifting feel like an ecological necessity?
- HMW make other people less judgemental about thrifting?
- HMW repackaging thrifting so her parents find it acceptable?

Solution and Experience Prototype Inspired by Olivia

Olivia and the repackaging POV inspired our third solution: a subscription service where the buyer receives used items in a personalized “mystery box.” Items for this “mystery box” are selected for the user based on their responses to a short survey. The assumption we wanted to evaluate was that the fun of a curated gift box exceeds the fun of just going to the thrift store yourself.

We set up a real-world experience prototype in which a test user received a mystery box of used items. We first collected some basic information from our user, including her clothing size and her favorite color. Based on her responses, Miles—a team member—visited his local thrift store and purchased items for her. Selecting items that matched her preferences went smoothly. Unfortunately, she wasn’t

interested in keeping most of the items; she explained that opening her mystery box “felt like a disappointing birthday as a kid.”



Figure 2: Photos from our “Mystery Box” experience prototype.

Point of View for Dan

- **We met** Dan, a semi-professional dumpster diver who hasn’t bought groceries since October, and bought a deep freezer to store goods through next Winter.
- **We were amazed to realize** that most of what Dan finds in dumpsters is practically new, and is only thrown away to make room for new items on the shelves.
- **It would be game-changing** if grocery stores had alternative ways to dispose of items that didn’t involve the dumpster.

How Might We Statements Inspired by Dan

- HMW redistribute grocery store waste that is still usable?
- HMW repurpose excess items into other new products?
- HMW encourage consumers to buy items about to be thrown out?
- HMW eliminate surplus inventory?
- HMW make waste more useful?
- HMW make sure everything gets bought before it's thrown away?

Solution and Experience Prototype Inspired by Dan

Dan and the final “how might we” inspired our third potential solution. To ensure that everything gets purchased before it is thrown away, we came up with another potential solution: an application that enables consumers to buy grocery stores’ items that will expire soon at a discounted price or for free. With this concept in mind, we started to create an experience prototype. The assumption we wanted to

test for this solution was that consumers are willing to buy older items at a discount.

To test this assumption, we created a Google Form survey that assessed consumers' comfort levels and preferences; 28 potential users responded. We found that our respondents overwhelmingly chose the older, cheaper item over fresher, more expensive items. Interestingly, this decision was not universal: few respondents chose to buy older produce, meats, and dairy products. Overall, we thought this test validated our assumption, as most respondents were willing to buy discounted older items.

Design Evolution

Selecting Our Solution

We next reviewed our three potential solutions to determine which was most promising for us to move forward with in our design process. To recap, the three solutions that we tested via experience prototypes were:

1. A reverse marketplace where buyers post what they need and potential sellers respond.
2. A subscription service where consumers receive "mystery boxes" of thrifted items based on their answers to a short survey.
3. An application that enables consumers to buy (discounted or free) items that will soon expire at grocery stores.

We decided not to move forward with the second and third solutions because we identified significant problems and barriers to implementation. The third solution—which allows consumers to buy expiring groceries at discounted prices—would require working with grocery stores, which is suboptimal during the current pandemic. Our third solution—the "mystery box" of thrifted items—clearly had little promise: our user told us that receiving the mystery box felt like a "disappointing birthday."

We felt our first idea, the reverse marketplace, was by far our most promising solution. It solves several needs: for example, by virtue of being hyper-local and relying solely on local pickups, this marketplace eliminates the need for shipping—the very need we were trying to meet in our original HMW. The reverse marketplace also unlocks hidden supply for buyers, eliminates sellers' need to manually list items they would be willing to sell, and supports buyers who have vague requests like "dorm room decorations." Second, the reverse marketplace idea is novel. We have never seen this type of marketplace before and feel that it could be revolutionary for our consumers. Its final merit was that it is interface-driven and could be successfully implemented via a web platform or mobile application.

Task Brainstorming

After deciding on the reverse marketplace, we brainstormed different tasks a user might want to accomplish with our application. We tried to select a mixture of simple tasks (common or introductory), moderate tasks, and complex tasks (which are often infrequent or for "power users").

Task Selection

After brainstorming, we selected what we thought were the four most important tasks. They were:

1. Our first task is simple: **a buyer requesting an item** using our platform. We selected this task because it is one of the core functions that our application needs to support; requests are crucial on our site and there would be no exchanges of goods without them.
2. Our second task is medium difficulty: **a seller making an offer** on a buyer's request. Again, this functionality is crucial to our application as no deal can be made on our platform unless an offer is made on a buyer's request.
3. Our third task is also medium difficulty: **a buyer accepting a seller's offer**. Again, this task is essential as our users must be able to exchange their goods with others.
4. Our final task is complex: **a seller setting up automatic alerts so that they will be notified when someone requests a particular item**. We thought supporting this task would help attract and support sellers in what is an otherwise buyer-oriented marketplace.

We feel that these four tasks cover the key functionality of Reclaim. We then moved on to creating our first prototype.

Low-Fi Prototype: Brainstorming

We began the brainstorming process for our low-fi prototype by drawing a series of concept sketches and task storyboards; these sketches helped us investigate different interface options and quickly iterate and improve our designs. These sketches helped determine the key structure and flow of our application. Here is a small sample of the sketches and storyboards we created before developing our low-fi prototype:

Reclaim

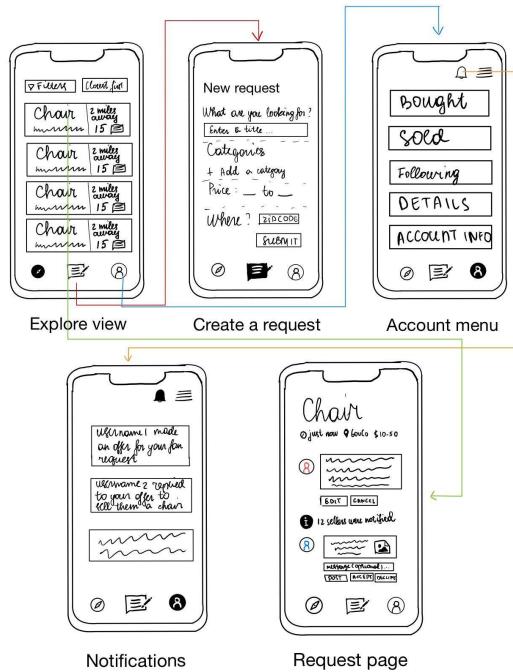


Figure 3: Linear mobile-oriented interface storyboard

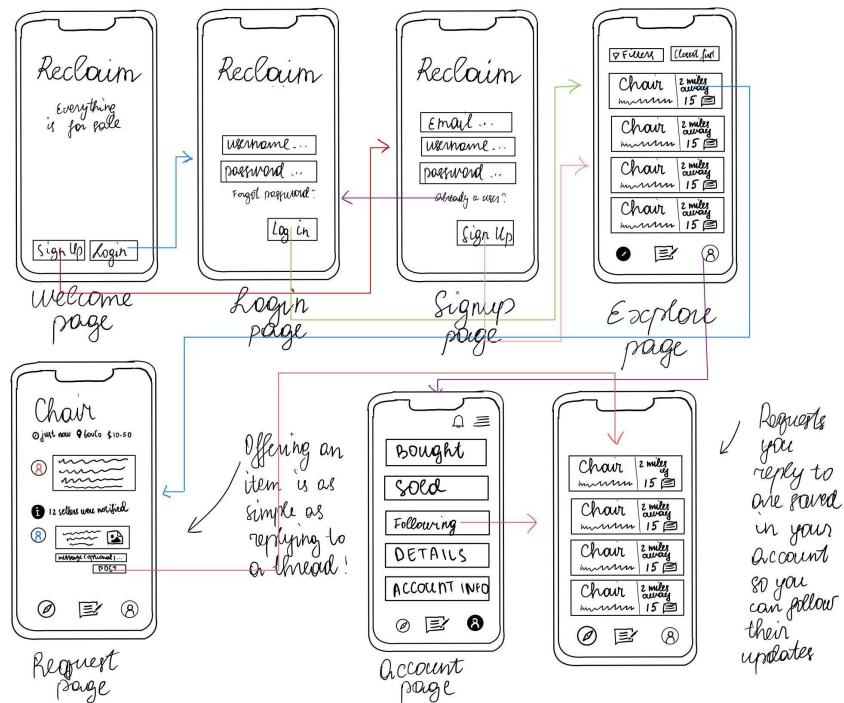


Figure 4: Storyboard for posting an offer for a request (medium task)



Low-Fi Prototype: Creation

We decided to build our Low-Fi Prototype using Balsamiq, a web-based mockup tool. This prototype supported our four key tasks but excluded support for many tangential functions, such as creating an account or logging in. Because showing users a design that appears highly polished can reduce the quality of user testing feedback for low-fi prototypes, we were careful to make our prototype as basic as possible. We used all of Balsamiq's sketch-like defaults, including their default font, scribbled lines in place of non-critical text, used no color, and excluded non-essential interface elements. Additionally, all of the prototype data is hard-coded, and all of the input field information is autofilled.

Low-Fi Prototype: Key Screens



Figure 5: Key screens from our low-fi prototype; task flows can be found in the appendix pg. 33-34.

Low-Fi Prototype Evaluation: User Testing

Methodology

The next stop along our design journey was testing our low-fi prototype. We performed four user tests via Zoom to evaluate the prototype. Our participant group was diverse and included a high school student, a college student, and two adults. All of our participants lived in urban or semi-urban areas where it is common to buy and sell used goods. Additionally, to reduce bias, none of our team members had ever met any of the participants; we typically interviewed friends of friends to avoid potential bias. Prior to the test, all participants signed consent forms.

During the test, we asked our users to perform our four tasks. We also asked our users to practice the “think out loud” method where they speak the actions they are performing to help us better understand their thought process and how they interacted with our prototype. We also had them hold their phone up to their camera so that we were able to see both their face and their phone screen during the user test. This approach enabled us to obtain more detailed information about their emotions and

expressions, helping us to further identify points of confusion and satisfaction.

Findings: Execution of Tasks

All of our participants were able to successfully complete our four key tasks. While the users were able to complete the tasks, we also noticed some significant points of confusion. They were able to perform task 1 (posting a request) and task 2 (posting an offer) quickly and smoothly; they did not end up on unintended screens or have to pause to think about how to proceed. However, our participants encountered some trouble while performing task 3 (accepting someone's offer) and task 4 (setting up automatic alerts). For task 3, participants had difficulty figuring out how to find the offer; all of them incorrectly clicked the account tab while we had intended them to find the offer through the notification bell icon. For task 4, participants again clicked the account bottom button instead of clicking on the notification bell icon. We next proceeded to look over our notes from our user tests to further look into common trends.

Findings: Key Strengths and Weaknesses

We were encouraged to see that our users were generally happy with our interface and were excited about the concept of a reverse marketplace. While running the test, our participant Alyssa commented that requesting an item is "pretty intuitive... easiest path I've ever gone through to get an item." While making an offer, our participant Isabel said that "it made sense how to post the offer... I like how simple it is." We also repeatedly heard that our users liked the unified view for requests as it shows all offers (and if the buyer has declined them) publicly. Users felt like this would help them better figure out what the buyer was looking for and would help them not make an offer similar to the one that was declined. As exciting as it was for us to hear that they enjoyed using our application and appreciated the concept, our participants also helped us realize that some parts of our application were confusing.

We noticed that our participants had three main areas of confusion. First, they were confused about our automatic alerts feature, which alerts a seller when a buyer requests a certain item. They had difficulty finding where to create automatic alerts and thought that clicking the notification bell icon to create an automatic alert did not make sense; they thought of notifications as a read-only part of the application. They also were confused how "automatic alerts" differ from general notifications. Second, they found the "+ Request" button on the bottom menu misleading. Some of our users thought they could click this button to access their past requests, however this button only helps users create ***new*** requests. Third, our users had difficulty finding many of the features that we had hidden on a menu on the account screen. For example, we had users' "Requests" and "Offers" only accessible through the account screen, and our users helped us realize that this was quite inconvenient. Additionally, users were unsure of the difference between "Your Details" and "Manage" on the account screen. With these problems in mind, we redesigned our application to make it easier and more intuitive to use.

Reclaim

Medium-Fi Prototype: Creation

Response to Low-Fi Findings: Design Changes

We made three major design changes to help our future users avoid the same confusions that our low-fi participants experienced. First, we renamed “Automatic Alerts” to “Watchlist” and added it as a button on the bottom menu to make it easier for users to locate. Second, we changed the “+Request” bottom button to “Your Posts”, a screen that includes options for the user to see posts where they made the request and posts where they have made an offer. It also includes a button for the user to create a new request. We hoped that this change would make it easier for our users to view their posts while continuing to make the “New Request” button easy to access. Third, we merged the “Your Details” and “Manage” buttons on the account screen to make it more straightforward to use. As we already moved the “Requests” and “Offers” buttons to the bottom menu to make them easier to find, we could also remove them from the account view. We felt hopeful that these changes would really help our users have a better experience using our application and were excited to incorporate them into the creation of our medium-fi prototype. Below you can see our plans for our design changes to incorporate into our medium-fi prototype.



Figure 6: Alerts redesign.

Reclaim

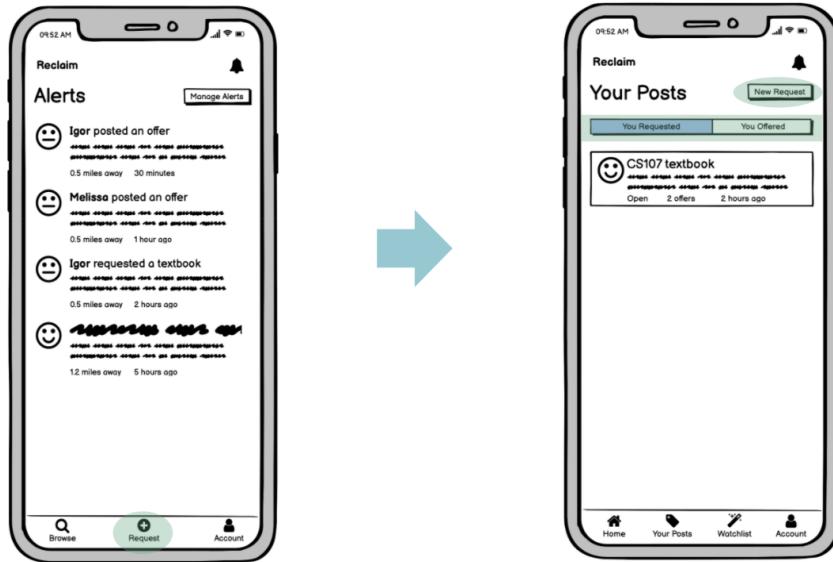


Figure 7: Bottom menu redesign.

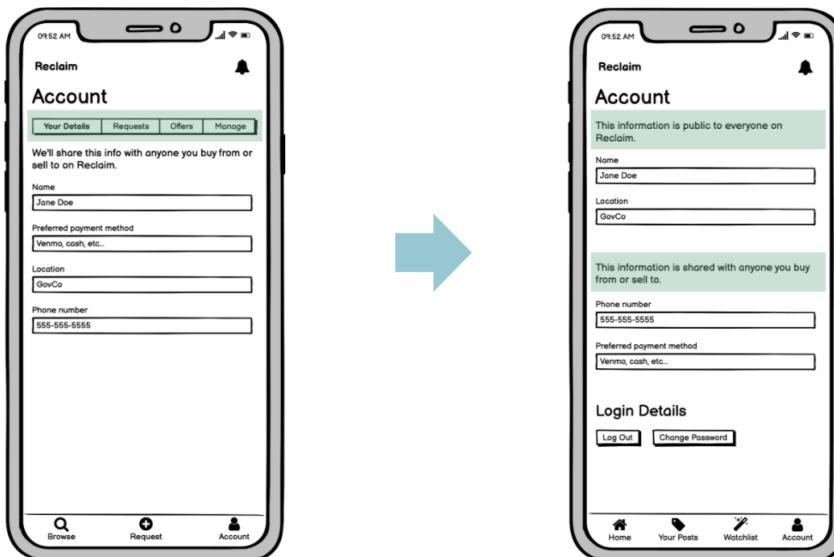


Figure 8: Account redesign.

Creation of Medium-Fi Prototype in Figma

We decided to create our Medium-Fi Prototype in Figma, a collaborative interface design tool often used for prototyping. We took our greyscale low-fi prototype and tried to bring it to life. Unlike our lo-fi prototype, in our medium-fi prototype we used color and made intentional choices about font and layout; our goal was for it to resemble a real-world app. We also incorporated the major design changes previously mentioned into this prototype.



Medium-Fi Prototype: Key Screens

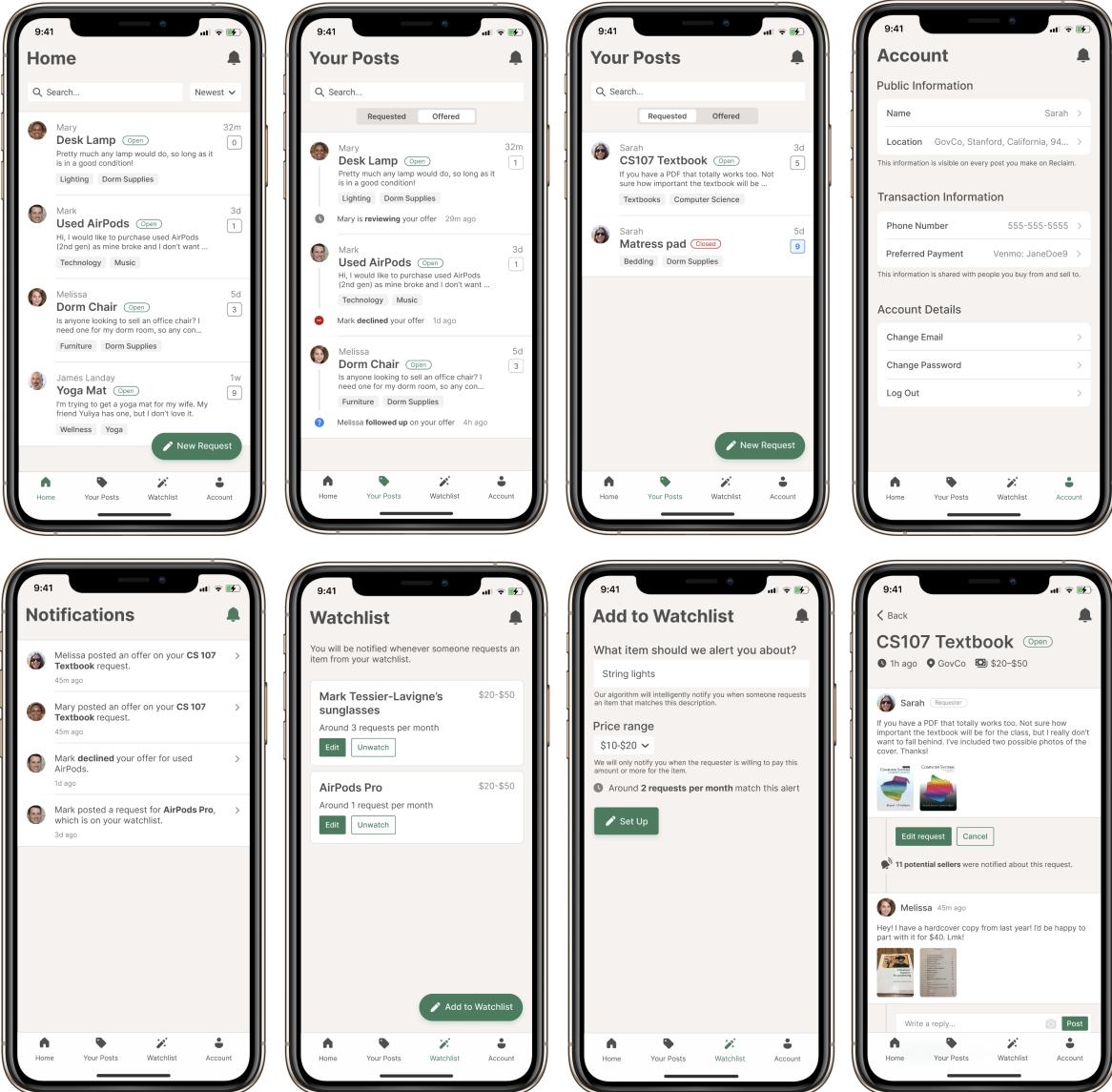


Figure 9: Key screens from our medium-fi prototype; task flows can be found in the appendix pg. 35-37.

Medium-Fi Prototype Evaluation: Heuristic Evaluation

Evaluation Technique

After we built our medium-fi prototype, a group of 3 “expert” classmates helped us find usability problems in our interface as part of a heuristic evaluation. They independently evaluated our interface using ten heuristics and sent us a report of their findings — including 54 different violations, each rated

Reclaim

with a severity of 1 to 4. If you'd like to learn more about how our application was evaluated at this phase, you can see the list of the ten usability heuristics in the appendix page 37-38.

Results and Revisions

After receiving our heuristic evaluation, we decided to prioritize the severity 3-4 violations and to only fix the simple severity 1-2 issues. Of the 54 violations, the three most common types were: 1) H4 Consistency and Standards (26%), 2) H3 User Control and Freedom (15%), and 3) H3 User Control and Freedom (15%). Below you will find a list of all of our severity 3-4 violations along with a few lesser severity violations that we thought were important or easy to fix. The severity is enclosed in parentheses next to the type of violation. There were also several recommendations that our evaluators gave us at the end of the report, some of which we will address. Below you can find a list of the violations and our responses organized by screen.

Violations: Account

1. H2. Match between system and the real world (4)

- **Problem:** There is no way to add a profile picture.
- **Solution:** We added the ability for a user to set their own profile picture.

2. H5. Error prevention (3)

- **Problem:** You can change your name frequently, potentially confusing others on the platform.
- **Response:** While we understand this might be confusing on a real-world platform, we think this issue is out of the scope of the prototype.

3. H6. Recognition rather than recall (3)

- **Problem:** You can change your name after a deal is finalized, potentially confusing others.
- **Response:** While we understand this might be confusing on a real-world platform, we think this issue is out of the scope of the prototype.

4. Recommendation (not directly related to a specific violation)

- **Problem:** Editing each account field takes the user to a new screen.
- **Solution:** We added an overlay pop-up for editing account fields to reduce the number of screens.

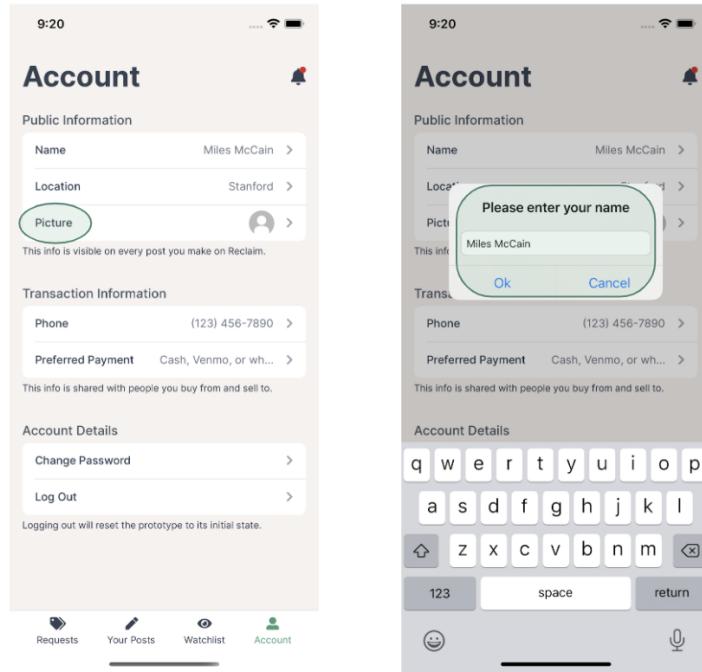


Figure 10: Key changes to the account screen are highlighted in green.

Violations: New Request

5. H2. Match between system and the real world (3)

- **Problem:** There is no way to select a price range lower than \$10 when creating a new request.
- **Solution:** We changed the price field to have the user select the higher bound of what they are willing to pay, and included a “Nothing (free)” option.

6. H5. Error prevention (3)

- **Problem:** There is no way to upload a photo to a request.
- **Solution:** We included the ability to upload photos to requests.

7. H3. User control and freedom (3)

- **Problem:** There is no emergency exit when creating requests.
- **Response:** We added a back button to the new request screen.

Reclaim

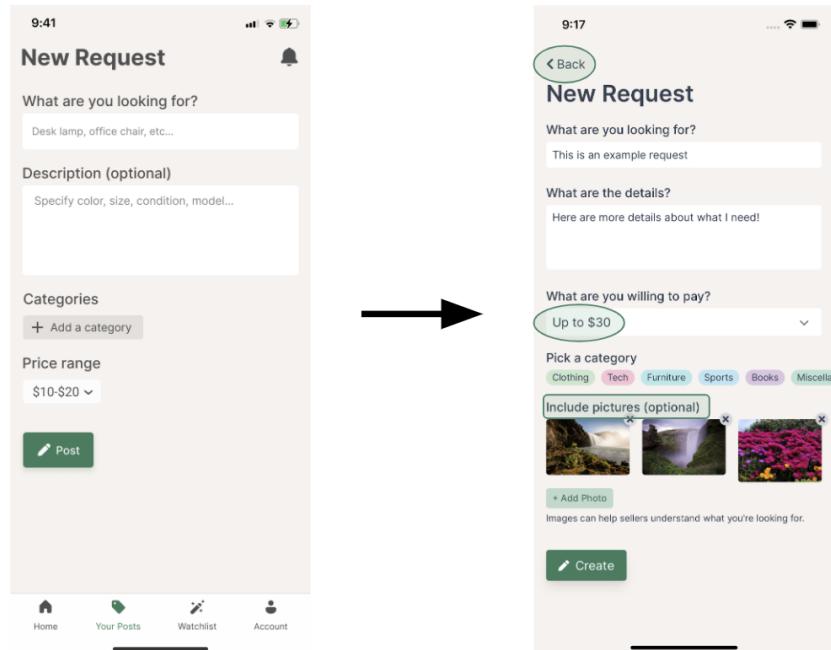


Figure 11: New request screen before and after the changes inspired by the heuristic evaluation.

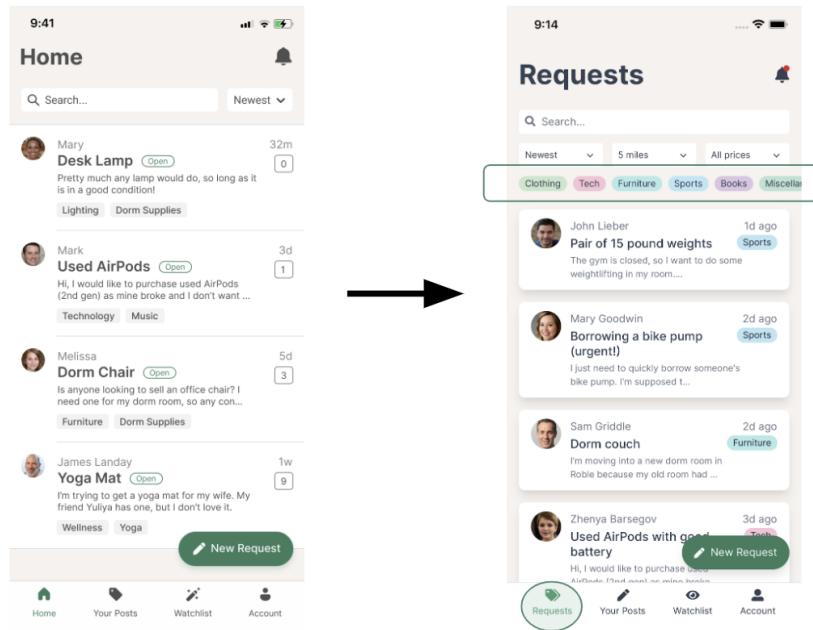


Figure 12: Home [renamed to Requests] screen before and after the changes inspired by the heuristic evaluation.

Reclaim

Violations: Requests

8. H1. Visibility of system status (2)

- **Problem:** Home label on the bottom toolbar is unclear.
- **Solution:** We renamed this tab “Requests” as it is a more descriptive name.

9. H2. Match between system and the real world (2)

- **Problem:** The green “open” tag on offers is unclear.
- **Solution:** We never had an “open” tag on offers, but we removed it from **requests** on the home screen because they are implicitly open.

10. Recommendation (not directly related to a specific violation)

- **Problem:** Not enough use of color.
- **Response:** We colored our tags and used elevation to denote separation to introduce more color and shadow into our application.

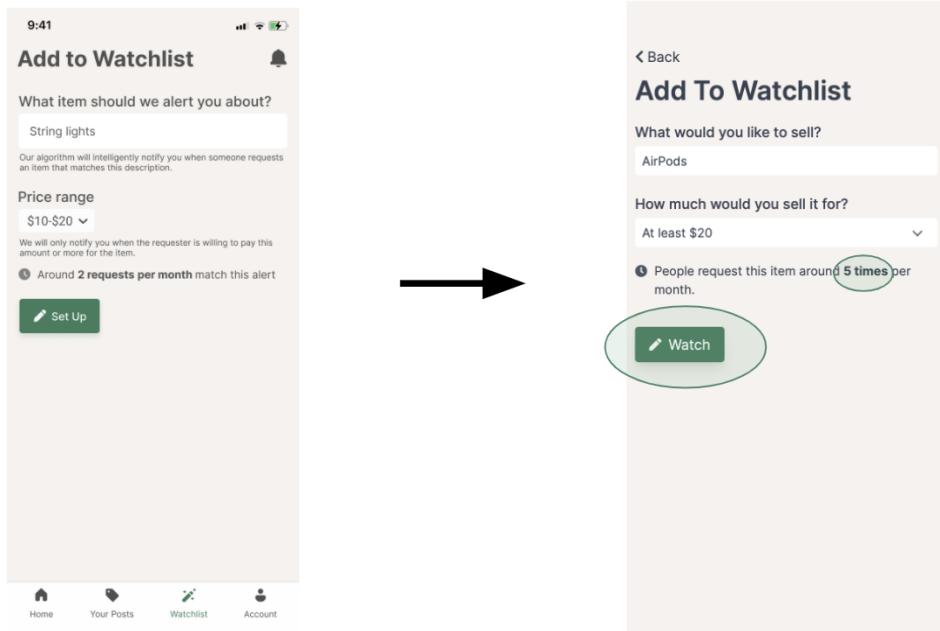


Figure 13: Add to Watchlist screen before and after the changes inspired by the heuristic evaluation.

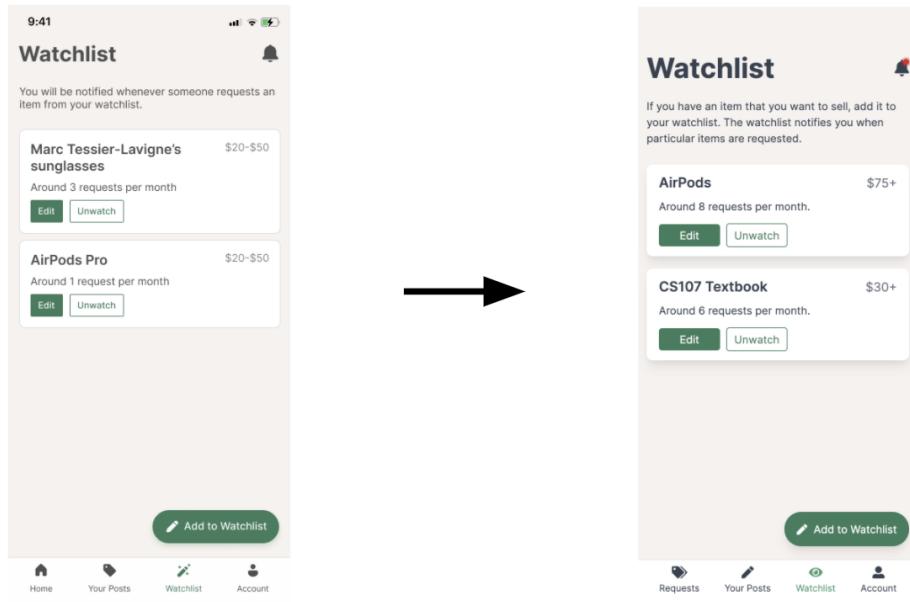


Figure 14: Watchlist screen before and after the changes inspired by the heuristic evaluation.

Violations: Watchlist

11. H3. User control and freedom (2)

- **Problem:** No emergency exit when adding to or editing the Watchlist.
- **Solution:** We added a back button to these screens.

12. H5. Error prevention (3)

- **Problem:** There is no confirmation message before you remove an item from the Watchlist.
- **Response:** We added a confirmation message before an item is removed from the Watchlist.

13. H5. Error prevention (3)

- **Problem:** The purpose of the Watchlist is unclear.
- **Solution:** We changed the description on the Watchlist tab to explain this feature in more detail.

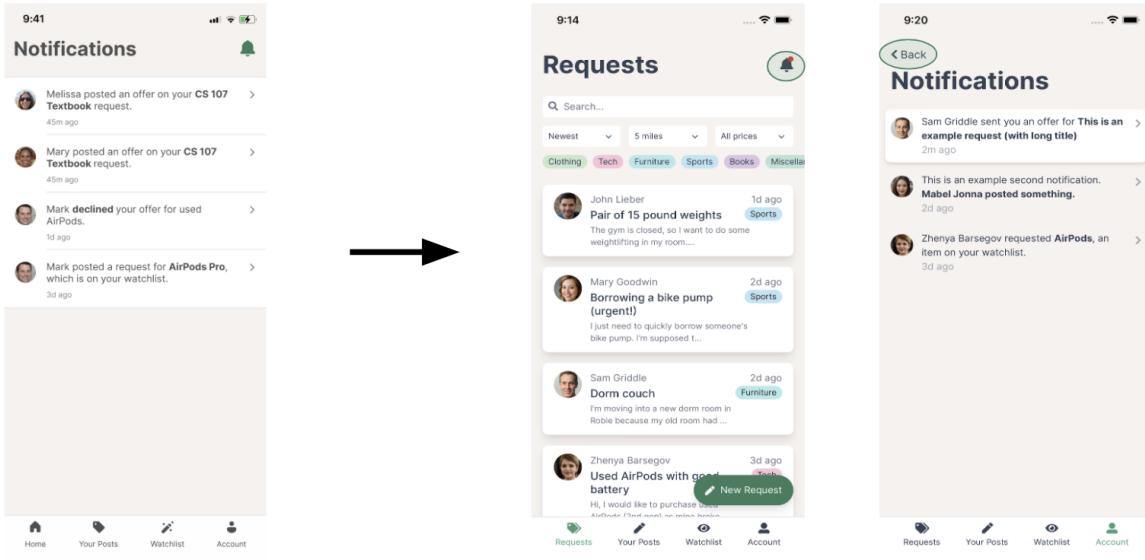


Figure 15: Notifications screen before and after the changes inspired by the heuristic evaluation.

Violations: Notifications

14. H1. Visibility of system status (3)

- **Problem:** There is no indication of unread notifications.
- **Solution:** We added a blinking red overlay to the bell icon and visually distinguished unread notifications on the notifications page itself.

15. H3. User control and freedom (3)

- **Problem:** There is no emergency exit on the notifications screen.
- **Solution:** We added a back button to the notifications screen.

16. H7. Flexibility and efficiency of use (3)

- **Problem:** There is an excessive number of screens for notifications.
- **Response:** There is only one view for notifications across the entire application; this is in line with other mobile applications.

17. H10. Help and documentation (3)

- **Problem:** There is no way to set notification preferences.
- **Response:** There are only two sources of notifications: your own activity and the Watchlist. We think the system-level notifications toggle is sufficient for the prototype, and implementing app-level notification settings is out of scope.

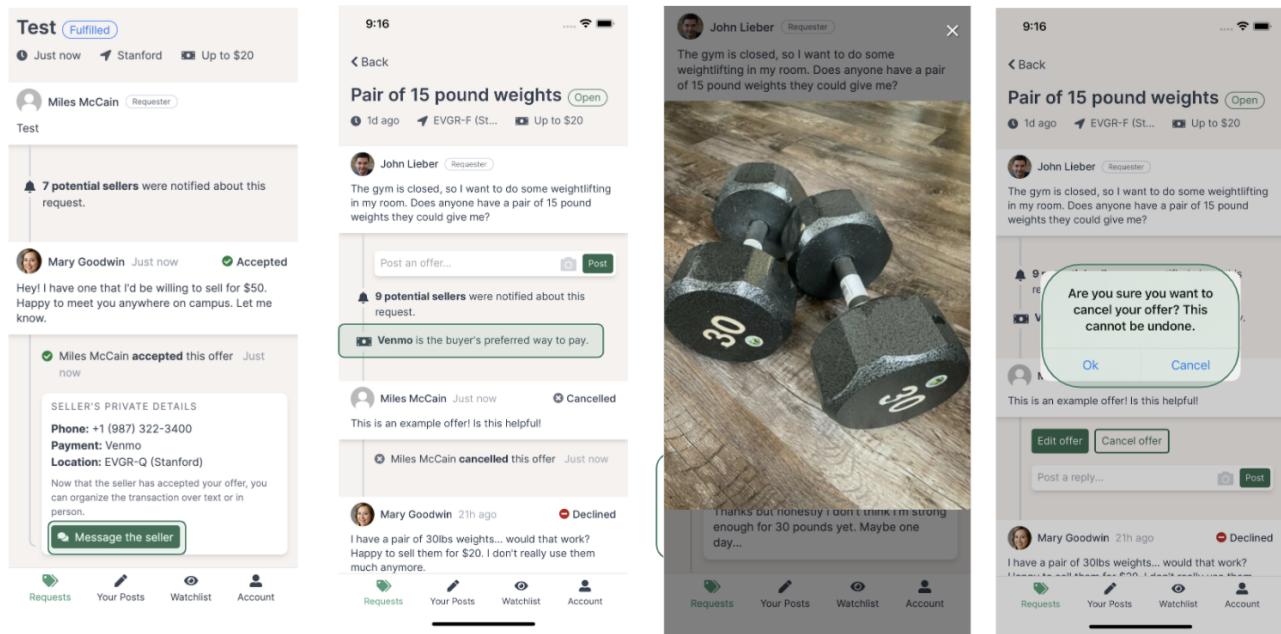


Figure 16: Offers screens after the changes inspired by the heuristic evaluation.

Violations: Offers

18. H10. Help and documentation (2)

- **Problem:** There is no guidance for communicating with the buyer after a deal.
- **Solution:** We added a button to allow the buyer and seller to message each other after transactions.

19. H3. User control and freedom (4)

- **Problem:** Accepted payment methods are not visible before the deal is finalized.
- **Solution:** We made it possible for sellers to view a buyers accepted payment methods before they decide to post an offer.

20. H8. Aesthetic and minimalist design (3)

- **Problem:** Description images are too small.
- **Solution:** We made the photo thumbnails enlarge when they are tapped.

21. H4. Consistency and standards (3)

- **Problem:** There are different button colors on the “accept offer” pop-up.
- **Solution:** We ultimately removed this particular pop-up, but we made the buttons the same color on pop-ups elsewhere in the app. (Note: the different button colors were intentional—we wanted

to denote a primary and secondary action—but the reviewers pointed out that this distinction may not have been important.)

22. Recommendation (not directly related to a specific violation)

- **Problem:** The function of the “Report a Problem” button is unclear.
- **Solution:** We removed this button as it confused our users and is outside the scope of the prototype.

23. H6. Recognition rather than recall (3)

- **Problem:** The conversation on the request page is difficult to follow.
- **Response:** We incorporated more whitespace and depth into our request view, and also introduced a smarter sorting algorithm for offers (declined and cancelled offers will always be pushed to the bottom to make way for newer open offers).

24. H1. Visibility of system status (3)

- **Problem:** There is no “unread” dot next to new offers.
- **Response:** New offers always appear at the top of the request view; we worry that adding additional indicators will make our interface cluttered for only marginal benefit.

25. H8. Aesthetic and minimalist design (3)

- **Problem:** There is too much text.
- **Response:** We reworked our interface to have more whitespace; we think that the remaining text is necessary and minimalist.

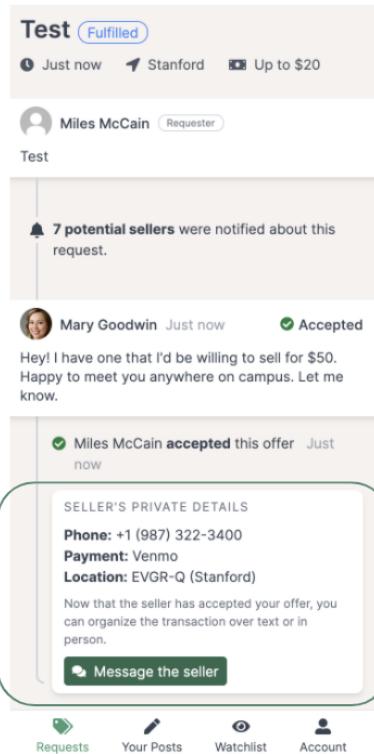


Figure 17: The seller's private details on your request after you accepted an offer.

Violations: Your Posts - Seller's Details

26. H6. Recognition rather than recall (3)

- **Problem:** The seller name and payment information is difficult to find.
- **Solution:** Clicking on an accepted offer from the 'Your Posts' page now brings you directly to the transaction details.

Final Prototype Implementation

High-Fi Prototype: Creation

We built our prototype primarily using Vue.js, Capacitor, and Xcode. Although our prototype is hypothetically cross-platform, we chose to target our implementation for the iPhone 11/12. By focusing on a single platform (iOS) and device size (iPhone 11/12), we could ensure our design patterns matched users' expectations while also optimizing for a single screen size. In our implementation, we used Vue.js to handle rendering, components, and state management, Capacitor to power the iPhone runtime, and Xcode for debugging and packaging. We also used Tailwind, a popular styling utility library, and a17t, a web component and customization toolkit. While we found our tools generally worked well, they fell short

in some areas—for example, they had little support for animations and transitions between screens, so we had to implement these ourselves with custom logic.

Hard-Coded Data & Limitations

We were able to fully implement all of our tasks in the high-fi prototype, though the app has limitations—for example, there is no central database, and some content is pre-written (sample posts, notifications, and offers) to support the task flows. However, the user may still dynamically create new posts, post replies, interact with simulated users, and update their account info (including the profile image). These updates will persist from session to session until the user logs out.

Other limitations include:

- The distance filter on the Requests page is non-functional; it exists simply to illustrate what the feature would look like if we incorporated proper geolocation.
- iMessage has a bug that causes conversations with phone numbers that do not exist to stall and to show up as “No Name”; because the phone numbers in our app are not real, the user may encounter this issue after pressing the “Message the seller”/“Message the buyer” buttons.
- Finally, because some of our ‘Wizard of Oz’ interactions are randomized, the user may see the same message repeated from different buyers or sellers (including within the same request).

“Wizard of Oz” Techniques

We also employed several “Wizard of Oz” techniques to support our tasks. When the user posts a request, they can press the ‘simulate an offer’ button to generate a random offer from a hard-coded user. Likewise, when the user posts an offer on a hard-coded user’s request, that user will respond two seconds later, either accepting the offer, declining it, or asking for more details. Finally, several other small interface components are randomized or hard-coded; these elements include the user’s location (which we autodetect to “Stanford,” though this can be overridden in the Account view), the number of users notified per request (a random integer), and the average number of requests per month on watchlist items (also a random integer).

Hi-Fi Prototype: Task Flows

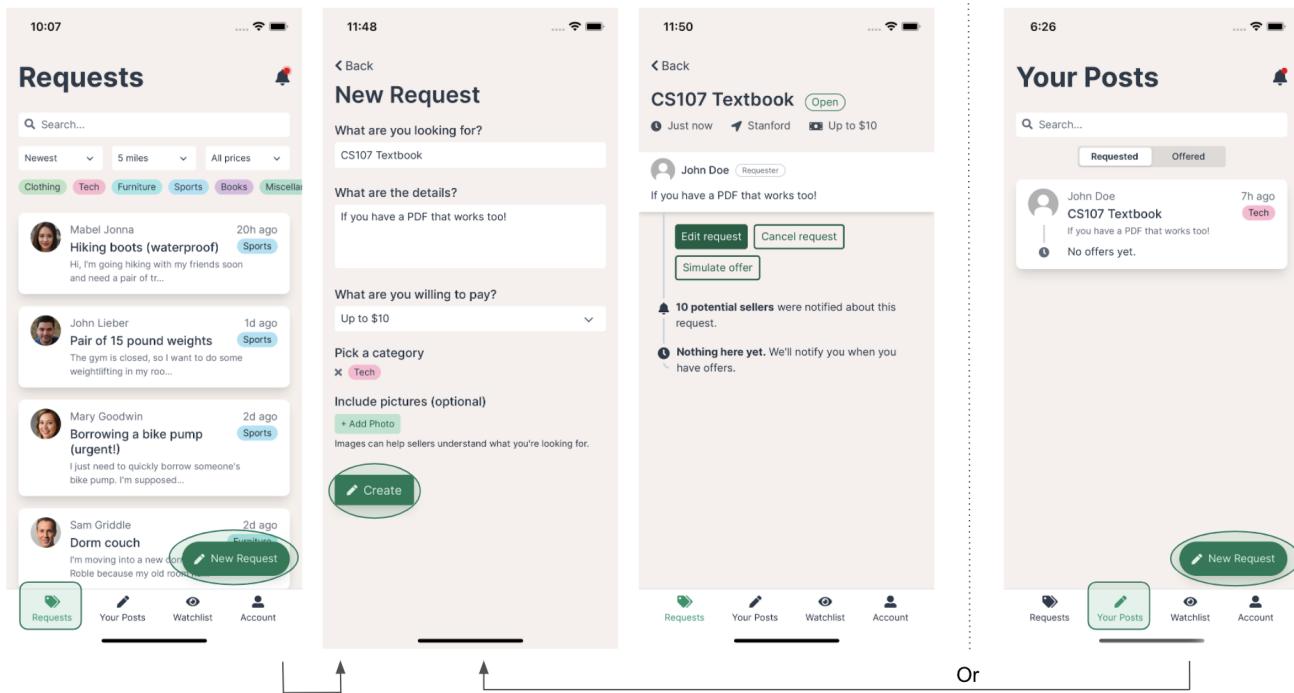


Figure 18: Flow for requesting an item (task 1).

Reclaim

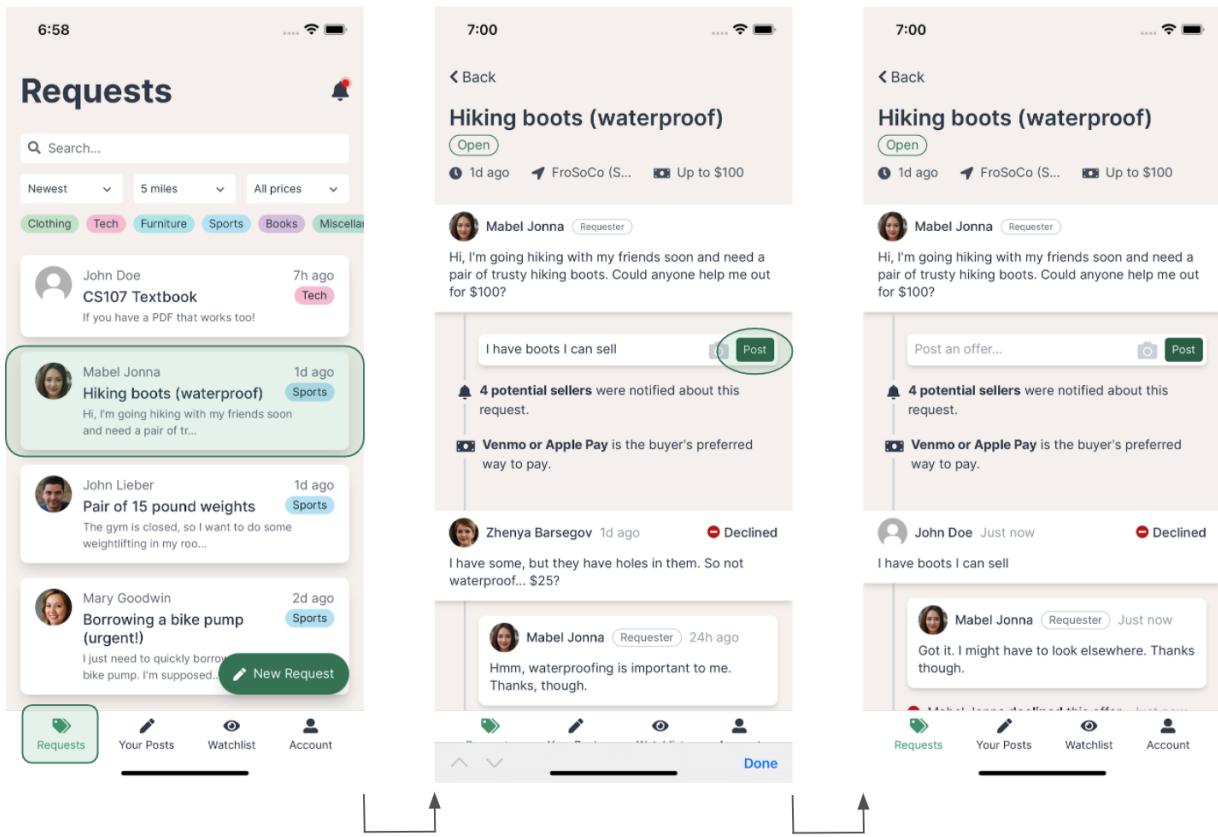


Figure 19: Flow for making an offer (task 2).

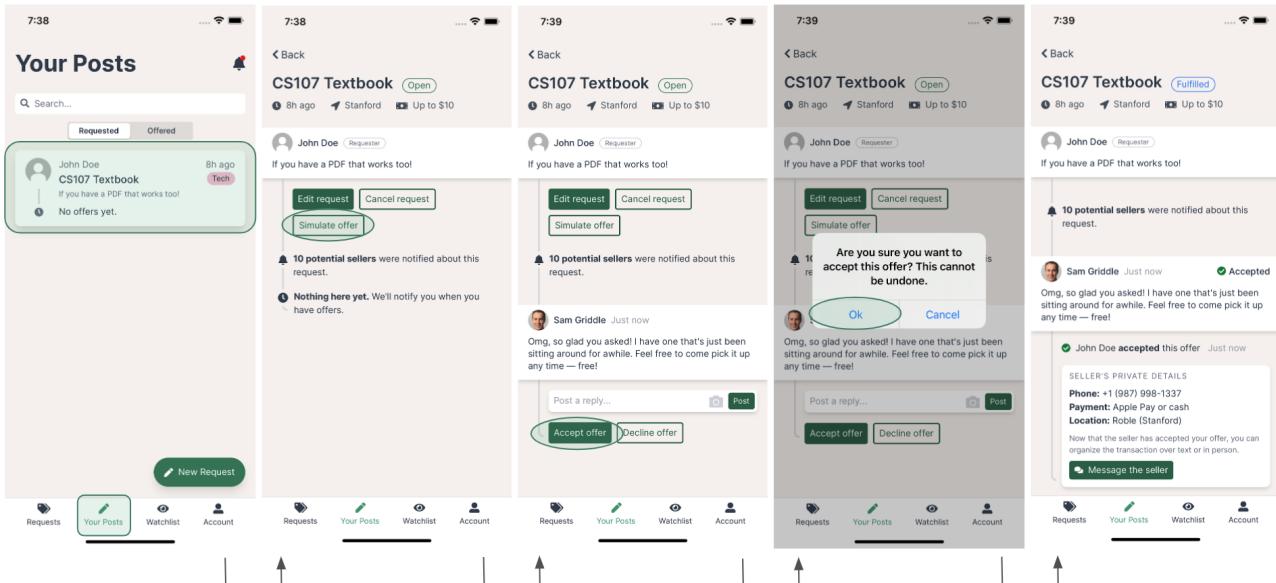


Figure 20: Flow for accepting an offer (task 3).

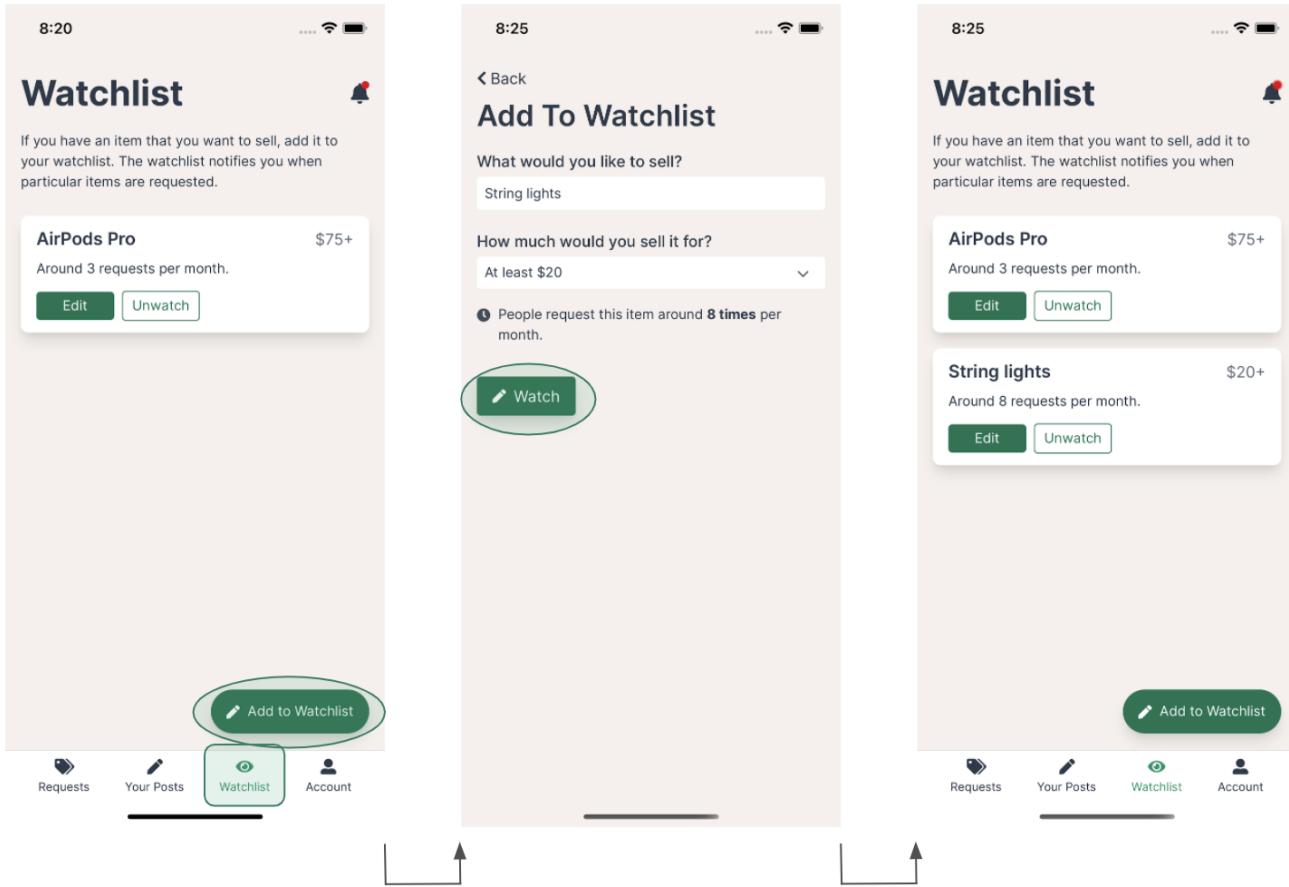


Figure 21: Flow for adding an item to the Watchlist (task 4).

Summary & Next Steps

Key Learnings About Sustainability

Waste

We were shocked to learn about the massive amount of goods and food that go to waste. There were two moments in particular that opened our eyes to the extent of waste in our world today. First, we were astonished to learn during needfinding that the average woman wears a piece of clothing only seven times before discarding it. Second, we were amazed that Dan—the nurse who dumpster dives—did not have to buy groceries for months at a time due to his success collecting perfectly edible “waste” from dumpsters. Both of these realizations fueled our desire to create Reclaim and help goods find new homes instead of ending up in the dumpster.

Intersectionality of Field

We learned that sustainability is a highly intersectional field. There are many different parties and types of individuals who work within the sustainability space, so practically everyone—nurses, students, engineers—has a role to play. We also learned that sustainability is something that many of our interviewees strive for, no matter their age, location, job, or education. However, many of them shared with us that they would only make changes towards sustainability if it was easy. Thus, we designed Reclaim to bridge this gap and help everyone be able to easily live more sustainability.

Key Learnings About Design Thinking

We learned so much about design thinking throughout our project, but three particular insights stand out. First, we saw firsthand the importance of iterative testing and responsive redesigns. For example, some parts of our low-fi prototype were completely redesigned in response to user feedback while others remain relatively unchanged. We realized that it was impossible for us to predict the results of user testing and grew to deeply appreciate the benefits of iterative design. Second, we thought at first that the different testing methods would uncover similar results. We were surprised to see that the low-fi user tests yielded vastly different types of information than the heuristic evaluation. This made us realize the importance of taking advantage of different types of testing as the types of issues uncovered can be different for these methods. Lastly, we learned about the value of teamwork and diversity of thought. As all of our team members come from different backgrounds and have unique strengths, we witnessed how useful diverse collaboration is in fueling design.

Future Work

Finalization of Application

Our application has several clear next steps. We would first link our prototype up to a real database so that data is accessible between users via the internet and the data persists even when users log out. We would also like to test our high-fi prototype with more users. As we have not conducted user testing since the low-fi prototype, we are eager to hear how users will respond to our design changes. We would also subject our application to one additional heuristic evaluation to ensure that we have identified any remaining key usability issues prior to deployment.

Deployment Strategy

After our application prototype is finalized, we would deploy it at our university in several stages. First, we would make it available to individuals within the same undergraduate dorm. We feel this is a good first step for deployment as there is inherent convenience and community for us to capitalize on during this stage of testing. To measure success at this stage, we would conduct several user interviews, paying attention to satisfaction in buying, satisfaction in selling, usability issues, and any potential new features our users think would be helpful.



Final Remarks

Thank you for joining us on our design journey. We enjoyed creating Reclaim and hope you enjoyed learning about the various stages of our design process—from needfinding and brainstorming to iterating through prototypes and testing. Please feel free to check out our website at <https://hci.stanford.edu/courses/cs147/2021/wi/projects/Sustainability/Reclaim> to see our high-fi prototype in action!

Appendix

Empathy Maps

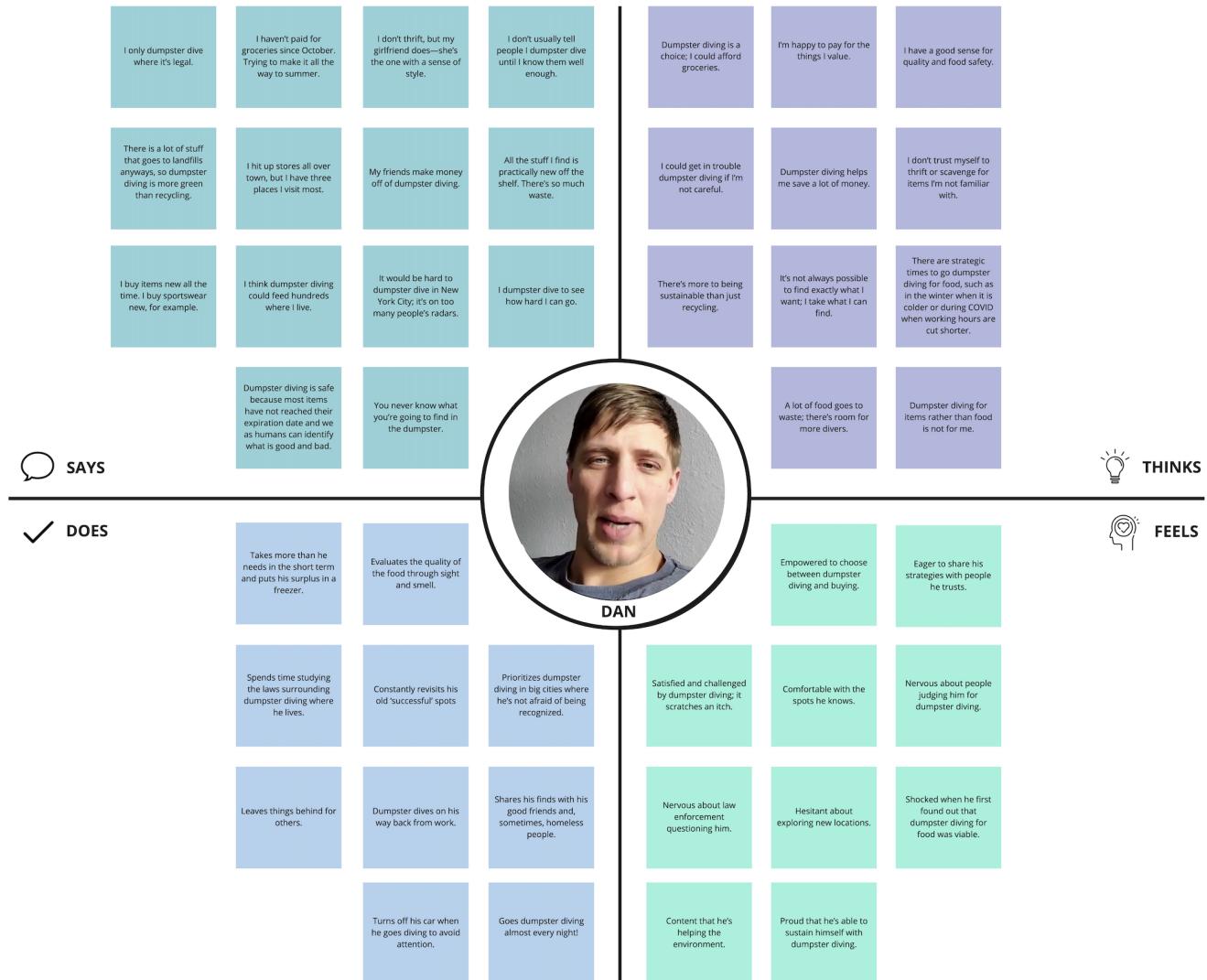


Figure 22: Empathy map for Dan.



Low-Fi Prototype: Task Flows



Figure 23: Flow for requesting an item (task 1).

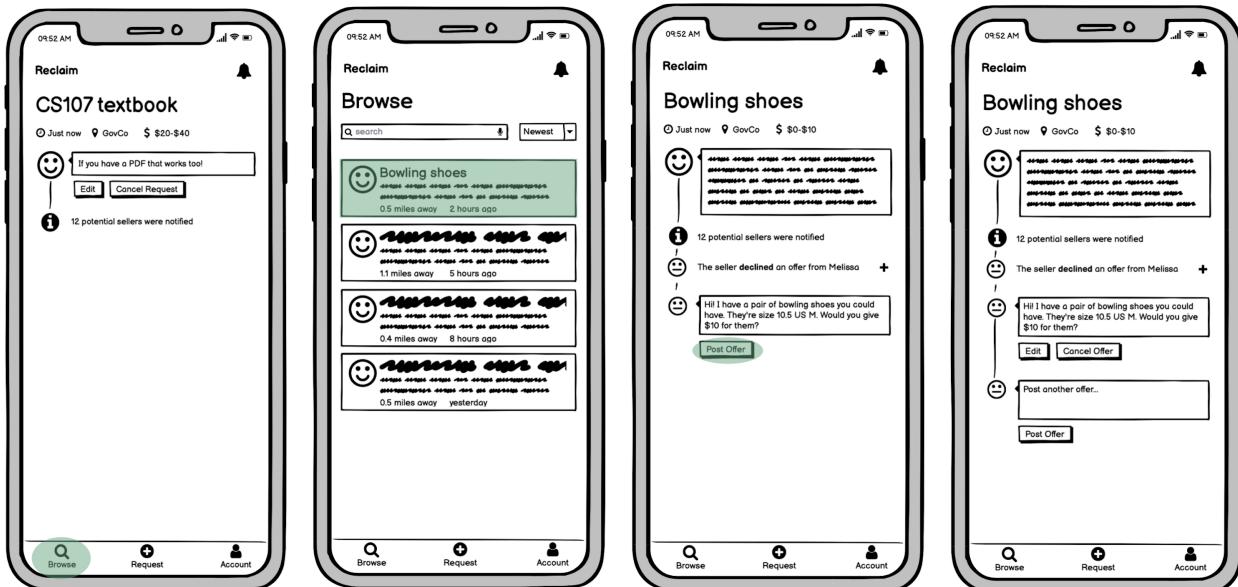


Figure 24: Flow for making an offer (task 2).

Reclaim



Figure 25: Flow for accepting an offer (task 3).



Figure 26: Flow for setting up automatic alerts (later renamed to Watchlist) (task 4).



Medium-Fi Prototype: Task Flows



Figure 27: Flow for requesting an item

Reclaim

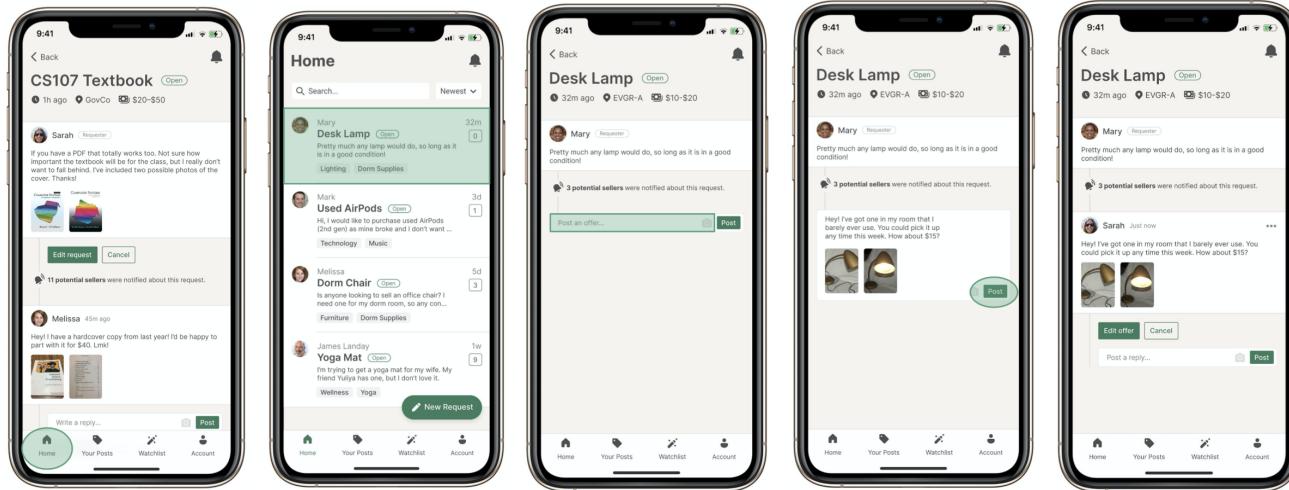


Figure 28: Flow for making an offer

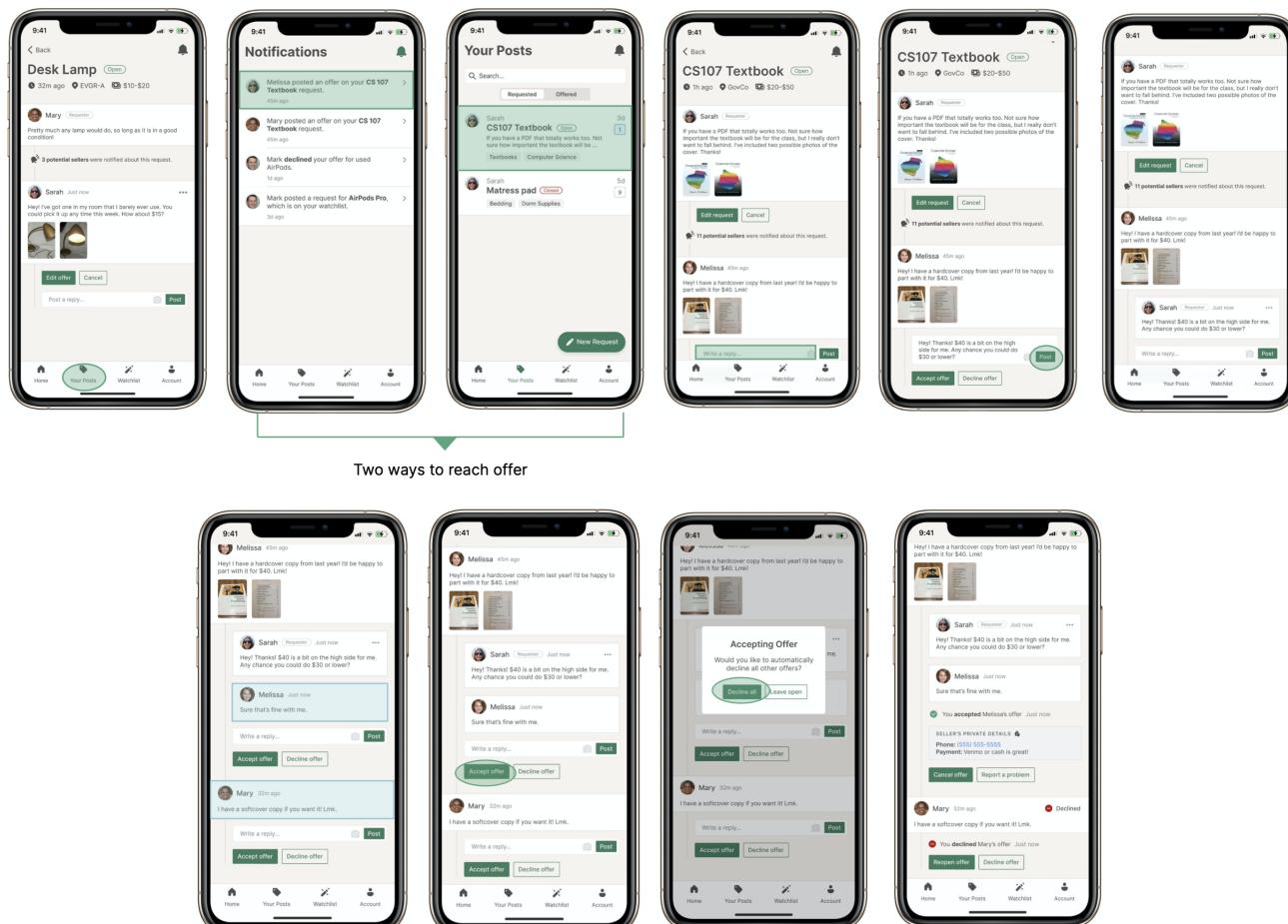


Figure 29: Flow for accepting an offer

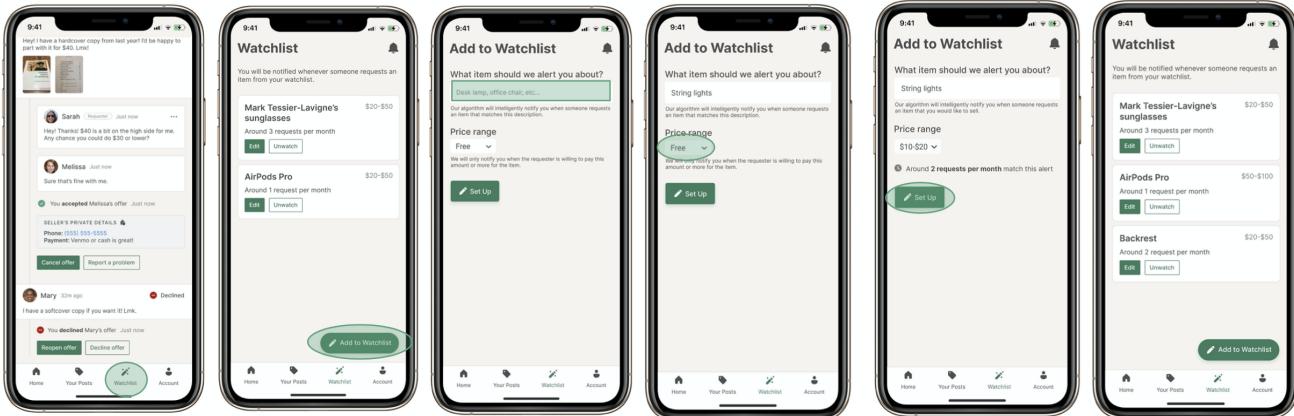


Figure 30: Flow for adding an item to the Watchlist

The Ten Usability Heuristics

- **H1. Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
- **H2. Match between system and the real world:** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
- **H3. User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
- **H4. Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
- **H5. Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.
- **H6. Recognition rather than recall:** Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable when appropriate.
- **H7. Flexibility and efficiency of use:** Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
- **H8. Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Reclaim

- ▶ **H9. Help users recognize, diagnose, and recover from errors:** Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
- ▶ **H10. Help and documentation:** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

High-Fi Prototype: Account Creation Flow

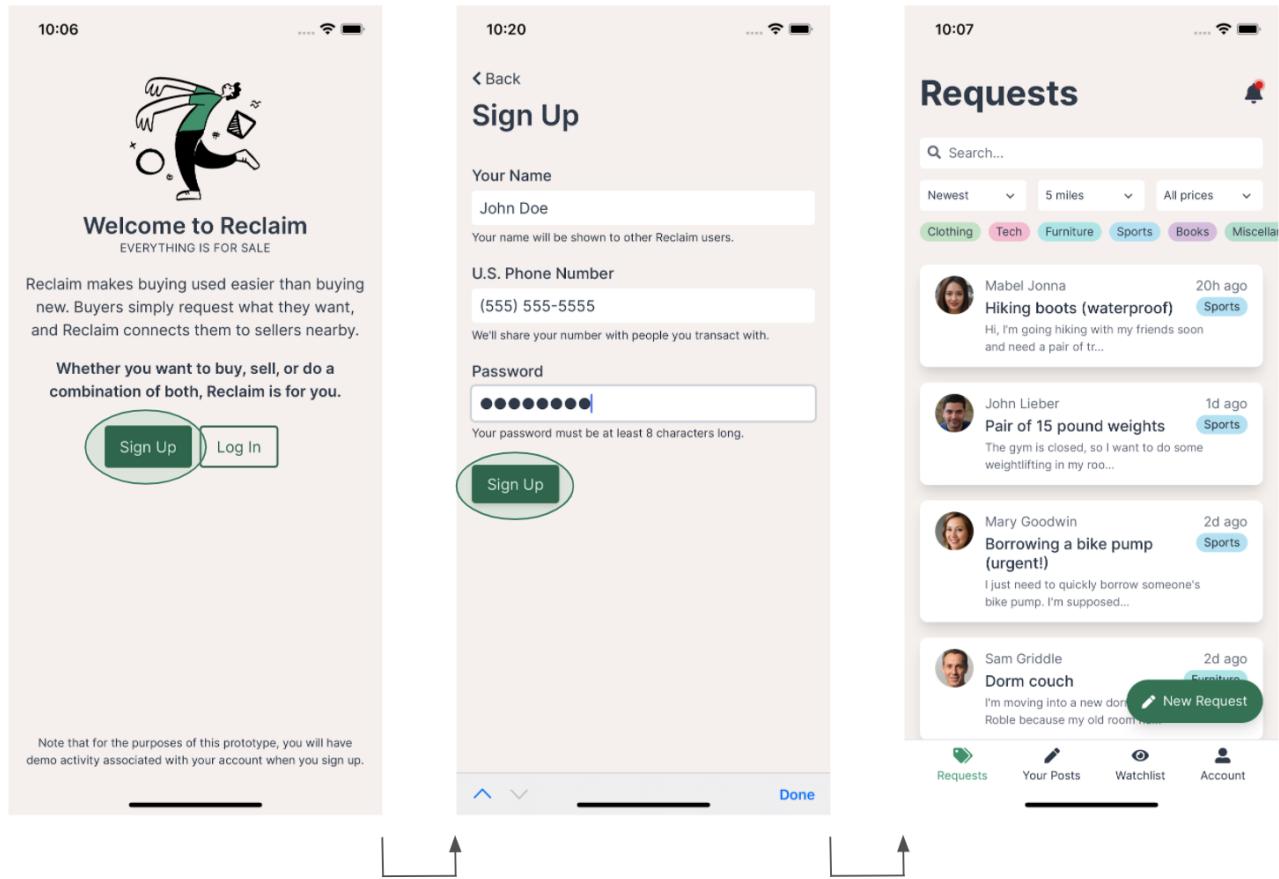


Figure 31: Flow for signing up for an account.