# armoir

find that missing piece

Alex Weitzman, Ellen Roper, Janna Huang, Starr Jiang

CS 194H: User Interface Design Project
Winter 2020

## I. Problem Description

People of all ages and backgrounds often have difficulty finding a fashion style that perfectly portrays them. For reasons such as financial constraints in exploring new styles, shortage of time to shop for new clothing, or lack of inspiration, people fall back into their routine styles or grow dissatisfied with the image their fashion choices portrays. Furthermore, the culture of “fast fashion” and throwing out clothes has a huge negative impact on the environment. The clothing and shipping industries are highly pollutive and the consequences of manufacturing and shipping new clothes are dire for the quality of our air, oceans, and ecosystems.

## II. Solution Overview

Armoir is a clothes-lending platform for people to discover their personal style in a sustainable manner by leveraging the power of their local community. See Figure 1. With Armoir, users can get inspired by others’ styles by browsing closets, borrowing their friends’ clothes, and lending their clothes for free or a small rate/day. Users can find that missing piece in their wardrobe that helps them complete, discover, or enhance their look, and the design for local communities mitigates the need for expensive and environmentally-harmful shipping for clothing handoff.

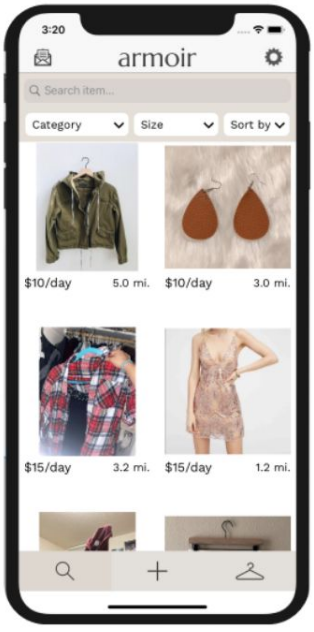



**Figure 1**. Armoir search screen

## III. Tasks

Because the Armoir platform relies on exchanges between two people, we divided our tasks between **lender** and **borrower**.

**Lender Tasks:**

***1. Upload clothing to your account***

In order to curate a collection of items, we need people to actually contribute to the marketplace. Thus, this is one of our most pivotal tasks. As such, we aimed to make it intuitive and user-friendly. When a user uploads clothing to their account, it appears in their virtual closet for easy access and editing. Users can upload clothing from the + button in the middle of the navigation bar, or when clicking 'upload item' in their closet.

***2. Mark clothing as exchanged***

In order to adequately keep track of confirmed exchanges, the lender must mark the clothing as exchanged before it shows up in the borrower's virtual closet. Lenders can do this by clicking 'mark as exchanged' button directly from the chat.

***3. Send a reminder to get clothing back***

Lenders can remind borrowers to return their clothing by simply clicking the 'send reminder' button in the 'lending' tab in their closet, which sends an automatic message to the borrower with a reminder of how many days are left in their agreement. This facilitates keeping track of their clothing without feeling pressure or the discomfort of confrontation with the borrower.

***4. Mark clothing as returned***

Just as with exchanging, the lender must mark the clothing as returned so that there is no ambiguity as to when the transaction has ended. It appears in the same place in the inbox/chat as the exchanged button once the item has been exchanged.

**Borrower Tasks**

***1. Explore and request to borrow clothing***

One of the core tasks for our app -- this is the heart of the clothing exploration and exchange process. The user can search by clothing name, type, or size, and they can order the search results by location or price for maximum flexibility.

### *2. Check when borrowed clothing is due*

In order for borrowers to seamlessly return clothing, they need to be able to track when it is due to avoid conflicts. To do so, borrowers can check the 'borrowing' tab of their virtual closet to see when items are due without even clicking a button.

### *3. Coordinate clothing return*

Also part of the return process, borrowers must coordinate with lenders on the when/where/how of the clothing return transaction. They can do this from the inbox feature of the app, where all communications between lenders and borrowers happens in one place.

## IV. Task Flows

**Upload clothing to your account**

To upload clothing, the user has two options: they can upload from the navigation bar at the bottom of the screen or they can upload directly from their virtual closet page. Both of these options follow the same flow, directing a user to the camera where they can take a photo of their item, then asking them to fill out details about the item such as size, asking price/day, and description. See Figure 2.


**Figure 2**. Upload clothing to your account

## Mark clothing as exchanged

To mark clothing as exchanged, the lender navigates to the inbox and subsequently the chat that concerns the exchanged item. They then tap the 'mark as exchanged' button to complete this task. See Figure 3.

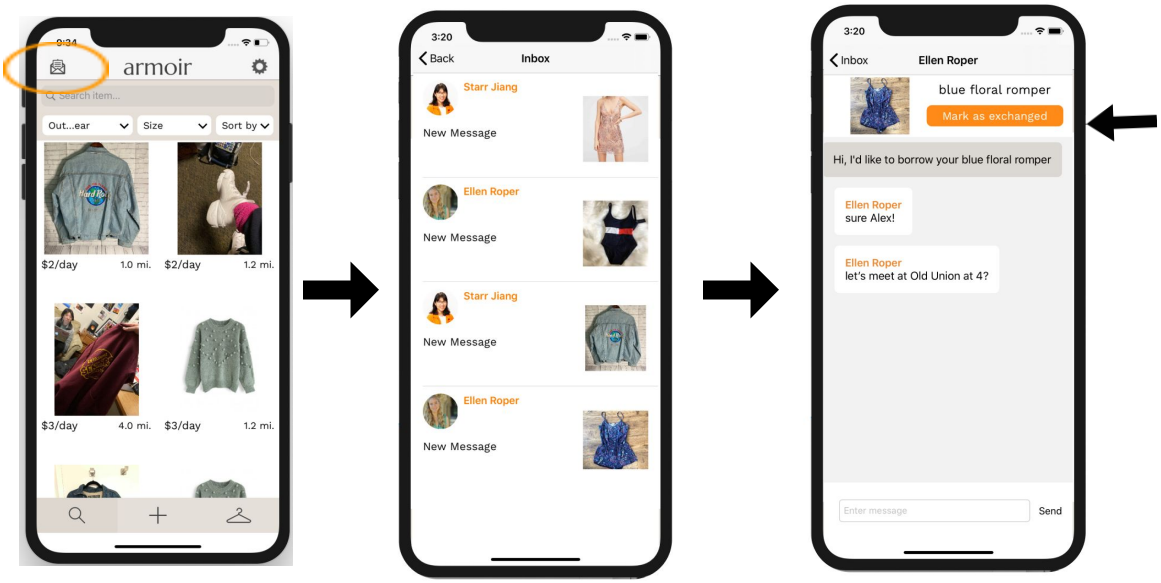



**Figure 3**. Mark clothing as exchanged

## Send a reminder to get clothing back

The lender navigates to their Closet page and then to their Lending tab, clicks the item, and taps the send reminder button. This sends an automatic message to the borrower as a reminder. See Figure 4.

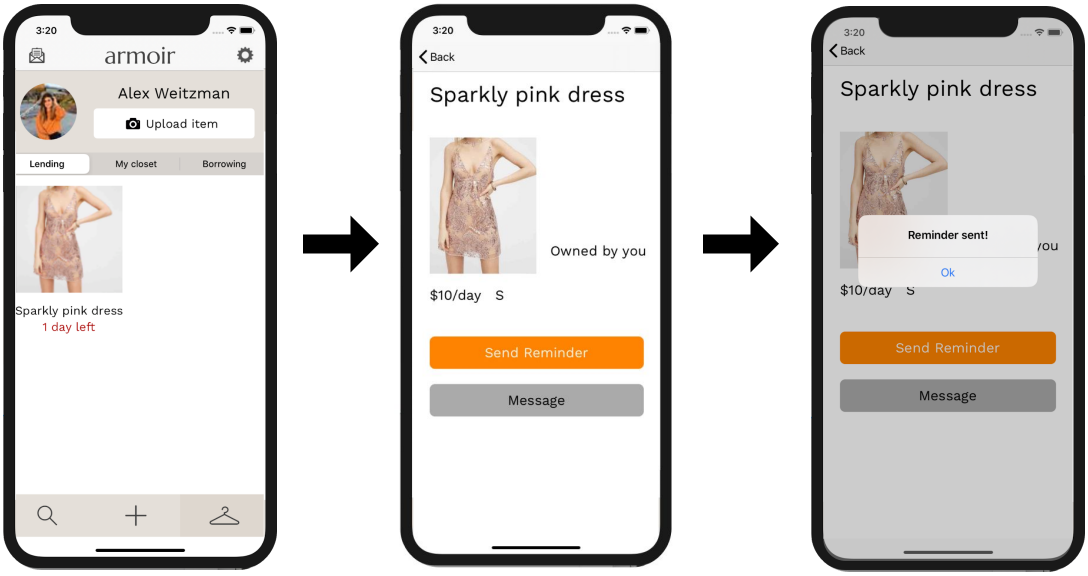



**Figure 4**. Send a reminder to get clothing back

### Mark clothing as returned

The lender navigates to the inbox, then to the chat for the item in question, and taps the “mark as returned” button. See Figure 5.

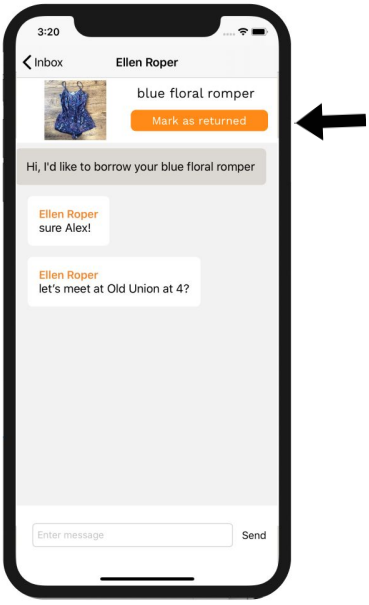



**Figure 5**. Mark clothing as returned.

### Explore and request to borrow clothing

The user can search for a specific item of clothing by name, size, or type, or they can simply browse by scrolling through the items on screen. They select the item they want and tap “request to borrow”, which automatically creates a chat with the lender for the item. See Figure 6.

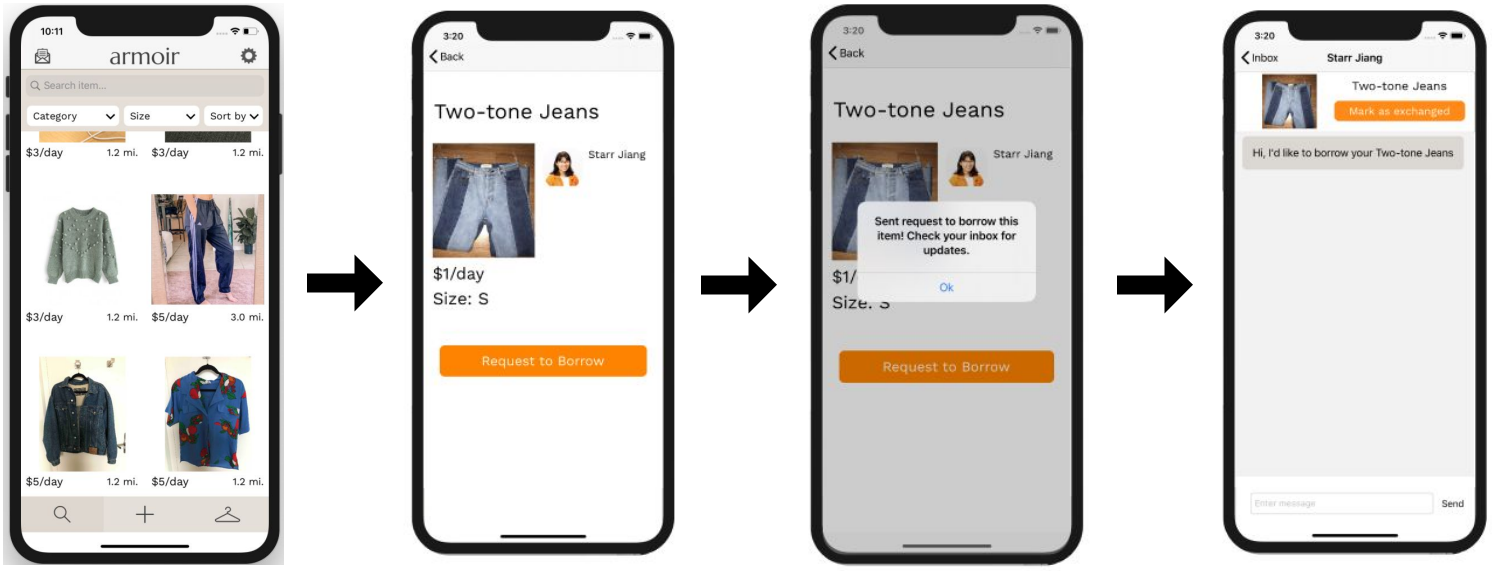



**Figure 6**. Explore and request to borrow clothing

**Check when borrowed clothing is due**

The borrower navigates to their closet, then to to the borrowing tab to see which items they are borrowing. Every item has its due date listed below the photo. See Figure 7.

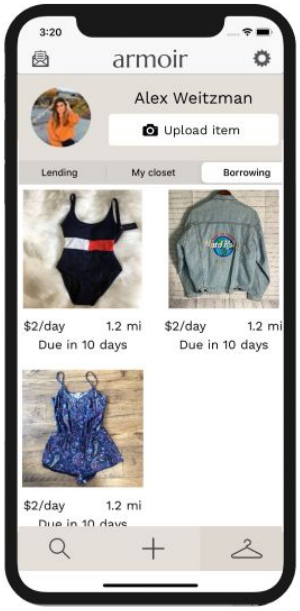



**Figure 7**. Check when borrowed clothing is due

## V. Design Evolution

### Initial Sketches

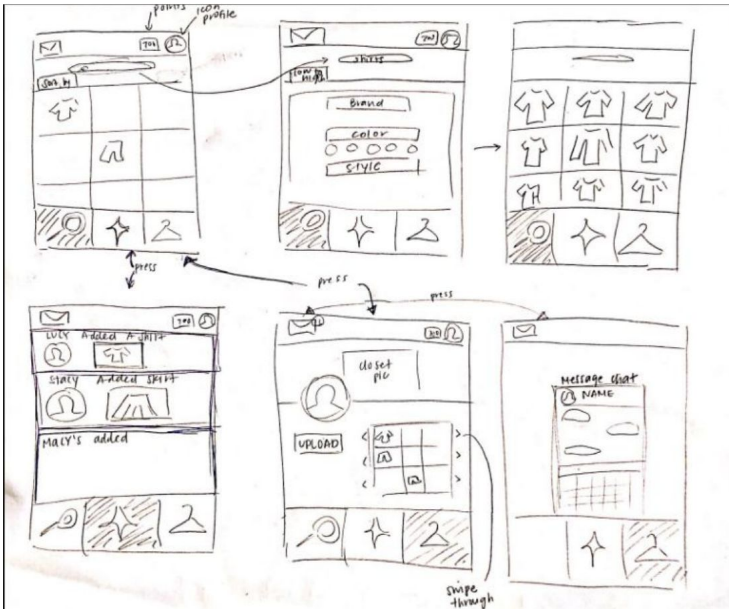



**Figure 8**. (Left) Tab-Based Interface

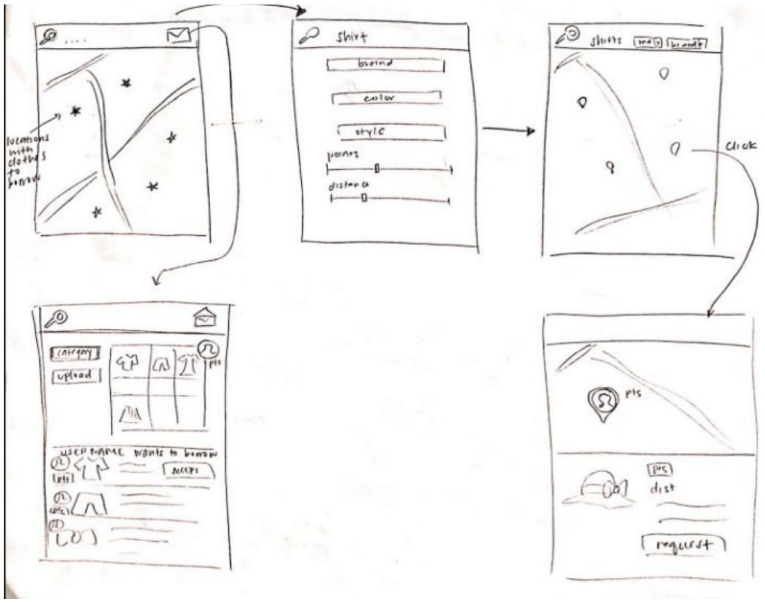



**Figure 9**. (Right) Map-Based Interface

The two storyboard interfaces that we came up with were a tab-based (Figure 8) and map-based (Figure 9) interface. We thought the map interface would be a novel way for people to browse and explore while also fostering a community-like environment by using location in a similar fashion that the Snapchat map feature does. Another approach was a tab-based interface which was more direct but also browsing heavy to foster more of a discovery environment. We chose the **tab-based design** because this interface put more focus on items than location.

### Lo-Fi Prototype

We started with a paper prototype (Figures 10, 11, 12) and fleshed out our three initial tasks, which were borrow clothes, lend clothes, and get recommendations. We then tested it in our first round of usability testing.


**Figure 10**. Borrow clothes (red shirt)

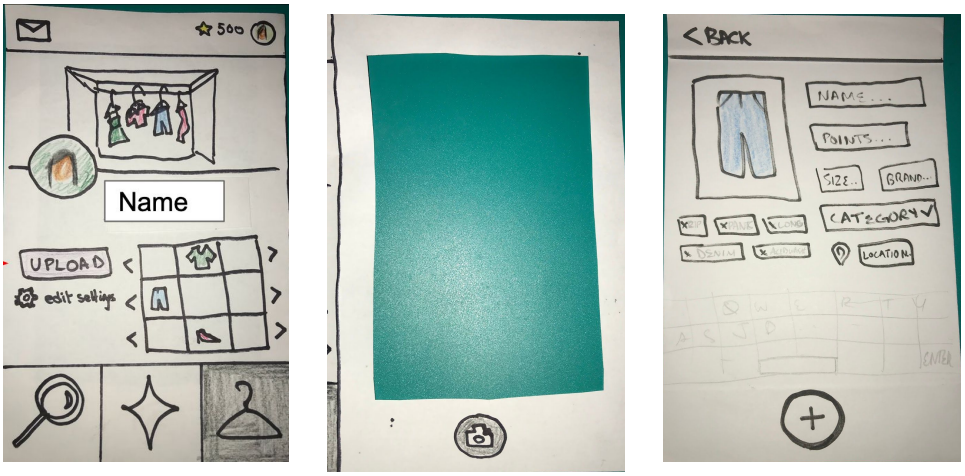



**Figure 11**. Lend clothes (add pants)

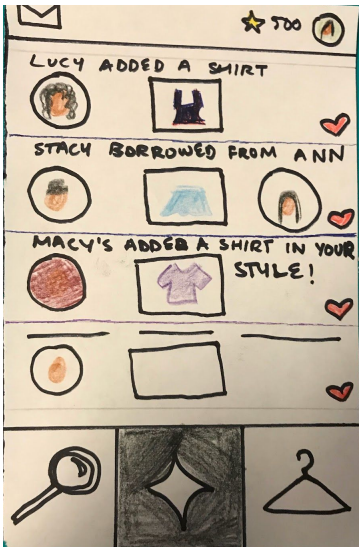


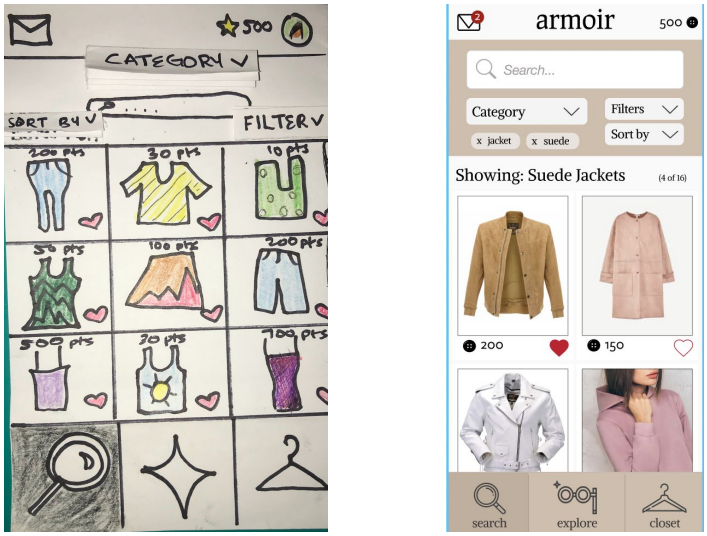

**Figure 12**. Get recommendations of what to borrow

## Med-Fi Prototype

From our prototype testing, we found that our low-fi prototype successfully created a space for "exploring". However, confusing layout of search bar and filter features made it difficult for users to search for clothes to borrow. We also found that users often had difficulty deciphering the icons of the tab. So, here were some of the changes we made.

### *Search*

We made the clothes bigger, emphasized the top search bar and filter drop downs, added labels to the navigation bar buttons, and designed the screen to make scrolling seem more obvious (i.e., cutting off the bottom images to suggest that there's more to see if the user scrolls down). See Figure 13.

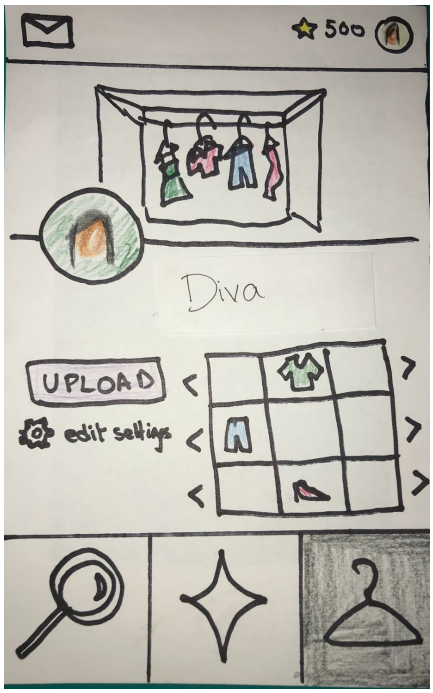



Lo-fi Med-fi

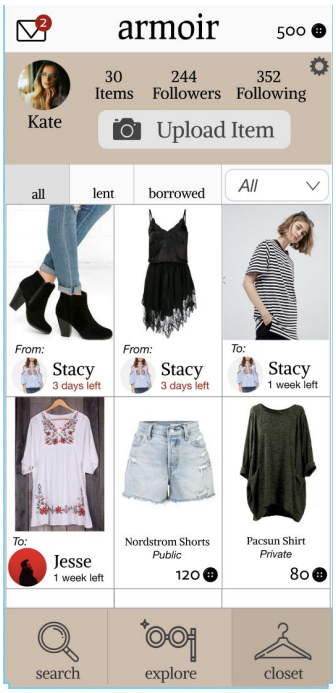
**Figure 13**. Change in search

### *Closet*

We learned that users found the 'cover photo' at the top confusing and made the closet tab feel too cluttered. We removed this, and added followers, tabs, and categories sections to easily sift through the closet, based on feedback that the closet could use more of the same grid approach as the browse feature. See Figure 14.




Lo-fi



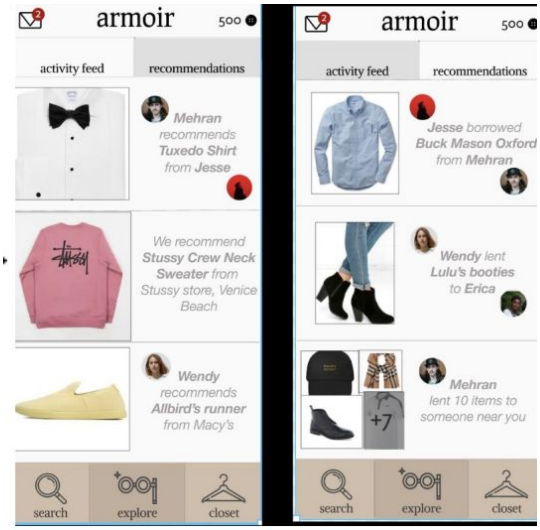

Med-fi

**Figure 14**. Change in closet

### *Explore*

People were confused by the mix of friends' activities and personal recommendations. We made two tabs to separate different information. See Figure 15.


Lo-fi



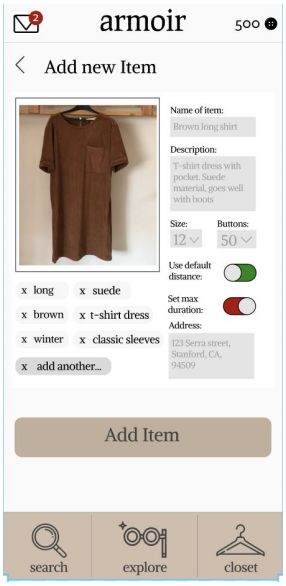

Med-fi

**Figure 15**. Change in explore

### *Upload*

We auto-filled some fields as a hint for users, made fields more descriptive and provided toggle options, and made tags more obvious to remove/add. See Figure 16.


Lo-fi




Med-fi

**Figure 16**. Change in upload

## Hi-Fi Prototype #1

One of the biggest takeaways when transitioning from the med-fi prototype was that the 'explore' tab was especially dense and confusing for users. The recommendations felt unintuitive to people and extraneous to the app's overall purpose. We included recommendations from friends, as well as recommendations from the app itself, which added too much complexity. Thus, our first major change was **removing the explore feature**. The explore tab in the navigation bar was removed and replaced with the closet tab to emphasize the "digital closet" aspect of Armoir. We added a "news" tab where the closet used to be, where users can keep track of their exchanges. See Figure 19 for comparison between med-fi and hi-fi #1 tabs. Because the explore feature was removed, we also took out the 'like' and 'follow' features.


**Figure 17**. Old vs new tabs

We also had recommendations to borrow vs. buy, but after much deliberation we felt that buying did not reflect the communal nature of the app that we wanted to create. Thus, our second major change was **focusing on borrowing/lending** (not buying).

We **refined the tasks**. Our old tasks were: 1) explore & borrow clothes 2) lend clothes 3) get recommendations to borrow/buy. Our new tasks became: 1) explore clothes 2) borrow clothes 3) lend clothes.

In our first hi-fi prototype, we added an inbox (non-functional yet) to the UI. We also added camera functionality (previously wizard of oz) and cosmetic changes to the bottom navigation bar and keyboard, and fixed stretched images in the app, as well as better search and closet functionality.

## Hi-Fi Prototype #2

In our next iteration, we added a rudimentary backend using Firebase and inbox functionality.

Based on usability testing, we learned an important insight that users have different mental models for “news” vs. “inbox.” In addition, users did not think news was intuitive. From here, we also clarified the navigation bar: got rid of the “news” tab, made the “closet tab” to keep track of borrowed/lent items, added a button for easy camera access, and changed some visual elements of the menu. See Figure 18.

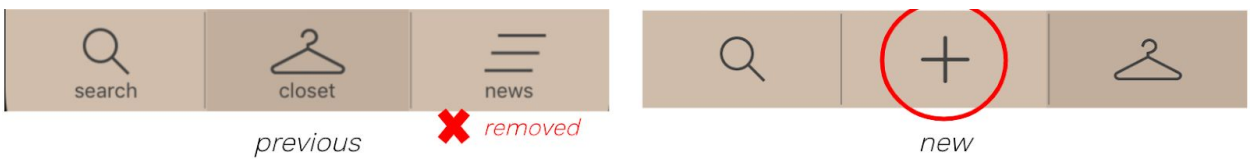



**Figure 18**. Change in navigation bar

We added user onboarding screens to establish explicit and implicit norms for uploading items, exchanging money, and what kinds of clothing are sold.

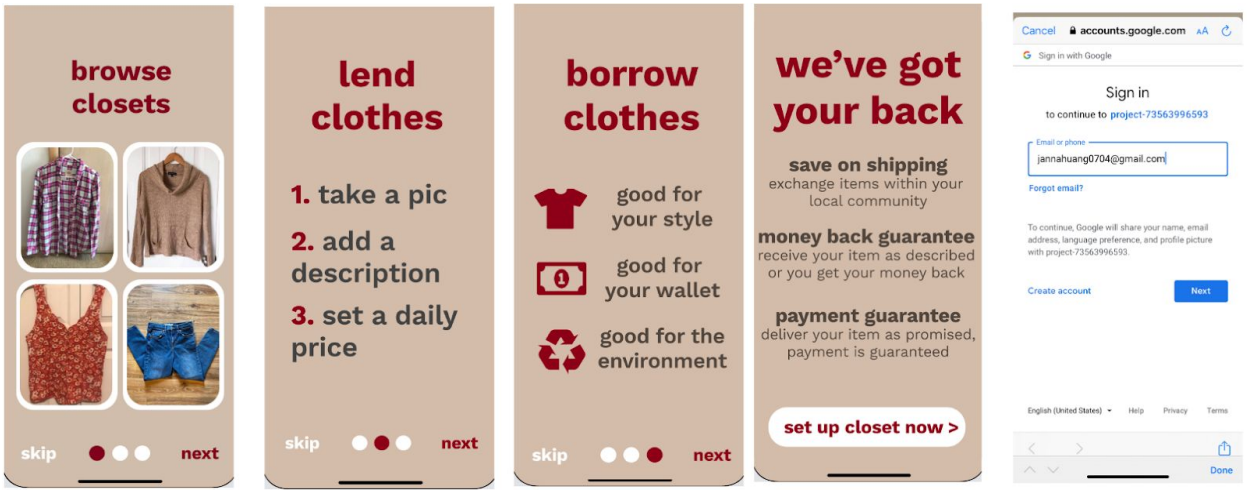



**Figure 19**. Onboarding screens

## Hi-Fi Prototype #3

We found that our tasks were not comprehensive enough to cover both borrower and lender needs. So, we broke our tasks up into the borrower and lender tasks outlined in Section III. This prototype consisted of many backend developments. For example, we implemented better inbox functionality, fixed item loading bugs, added functionality to the closet to connect to clothing database, added clothing item details accessible from browse or closet, and extended database functionality in general. On the UI side, we updated our app color scheme and refreshed our fonts to look more modern after consulting with an experienced visual designer (Elizabeth Lin). We also clarified the onboarding "browse closets" screen with a screenshot of the browse page to show that this was a direct feature in the app, as people tried to tap the icons in the original one during field usability testing. See Figure 20.

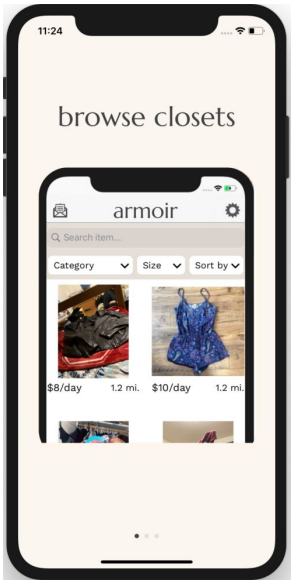



**Figure 20**. First page of new onboarding screen

In this overall design evolution, we found the usability testing to be the most helpful in refining our features and understanding what users felt were intuitive.

## VI. Final Interface

### Final UI Design + Functionality

#### *Onboarding*

When the user starts the app, they are taken through a series of onboarding screens to show them basic functions of the app and to set implicit and explicit norms about how to upload clothes and use the app. Then, they log in using a Facebook account to create their Armoir profile.

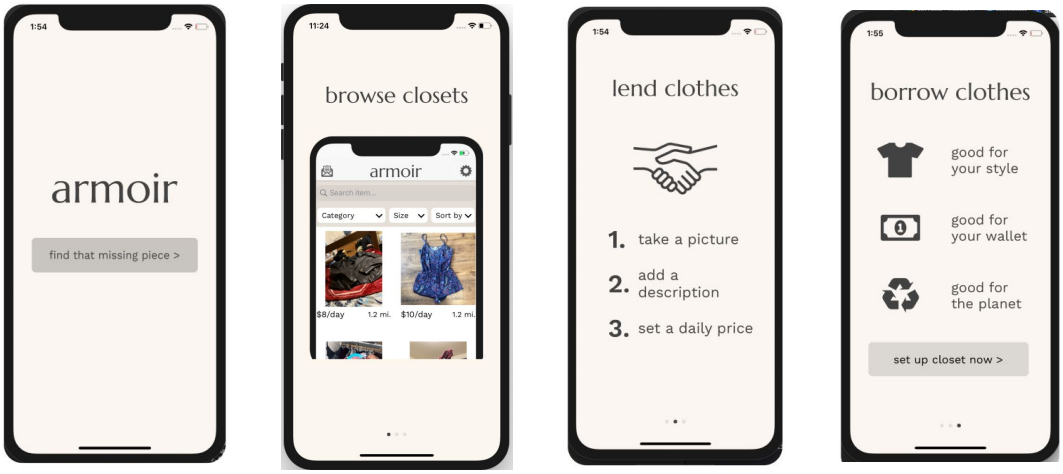



**Figure 21**. Final onboarding screens

### *Search Functionality*

The user can search among the catalog of all items that users on the platform have uploaded in several different ways. To browse, they can scroll through the full pages of clothing listings or search by item name. They also have access to various filters to further refine their search. Users can choose from overall clothing categories such as shirts, pants, dresses, etc., as well as filter by size and sort by price and distance.

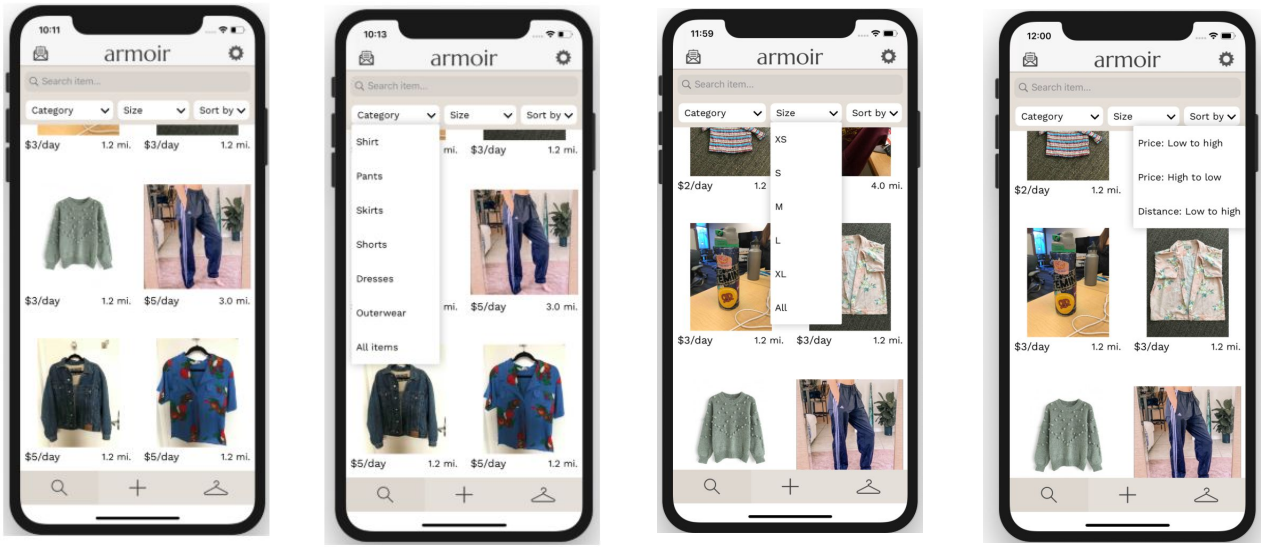



**Figure 22**. Final search feature

### *Upload Item to Closet*

The user can upload an item to their virtual closet using their phone camera. They access this either from the navigation bar or from their closet page, and once they take a picture they can label the item, name their lending price per day, and adjust the category and size of the item. Once these are filled out and the user taps “add item,” the item listing is added to our database backend and is immediately available for others to search for and borrow.

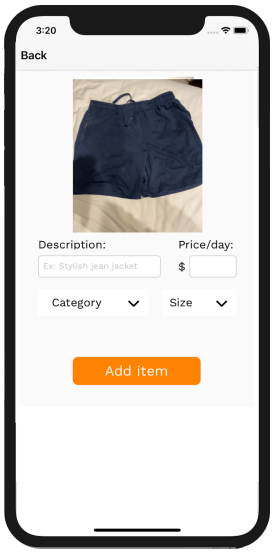



**Figure 23**. Final upload feature

### *Tracking items*

Using the Closet page, the user can track: a) the items they are lending out to people, located under the “Lending” tab, b) the items they currently have up for others to borrow in their closet, labeled “My Closet”, and c) the items they are currently borrowing, under the “Borrowing” tab. These values are real-time synced with the database backend and update accordingly when users borrow or lend items. Furthermore, the user can send reminders to or message the people borrowing their items to coordinate return/check in.

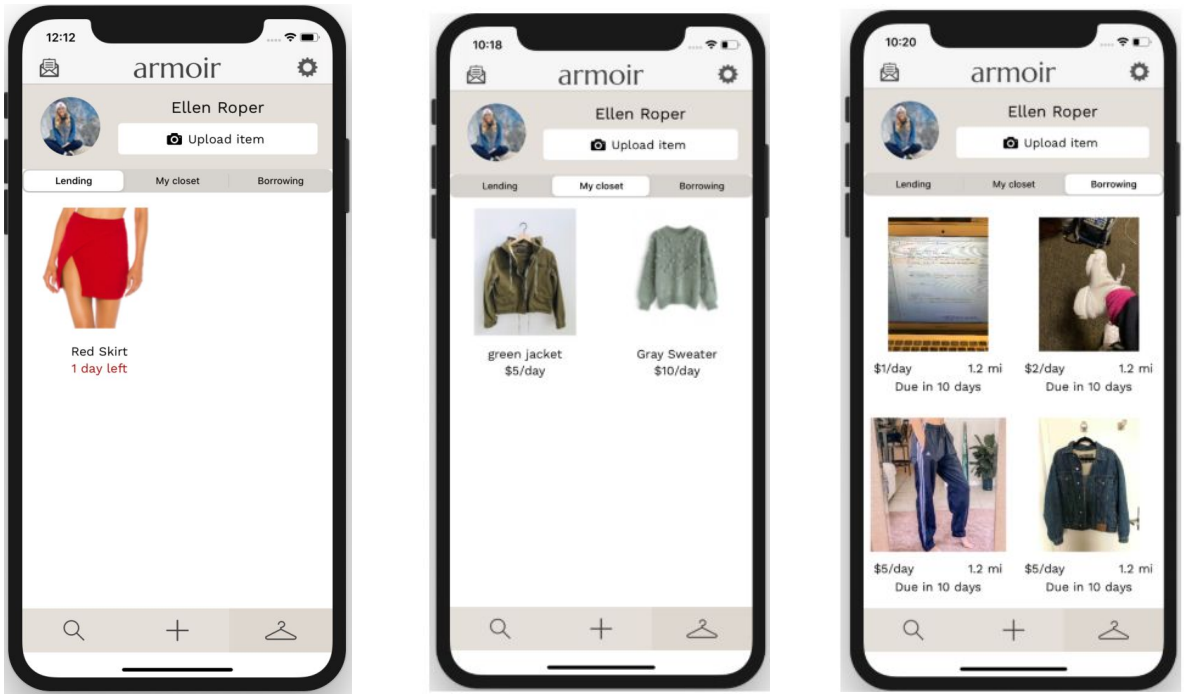



**Figure 24**. Final closet feature

### *Inbox and Messaging*

We have implemented a fully functional messaging system for our app that uses our real-time database. Each chat is automatically created when one user requests to borrow an item from another, and the chat shows up for both users in their inbox. From there, users can message each other to coordinate pickup/dropoff, payment, reminders, and any other logistical issues regarding the exchange. Users can also mark when the items are exchanged or returned in order to ensure accurate tracking of the items.

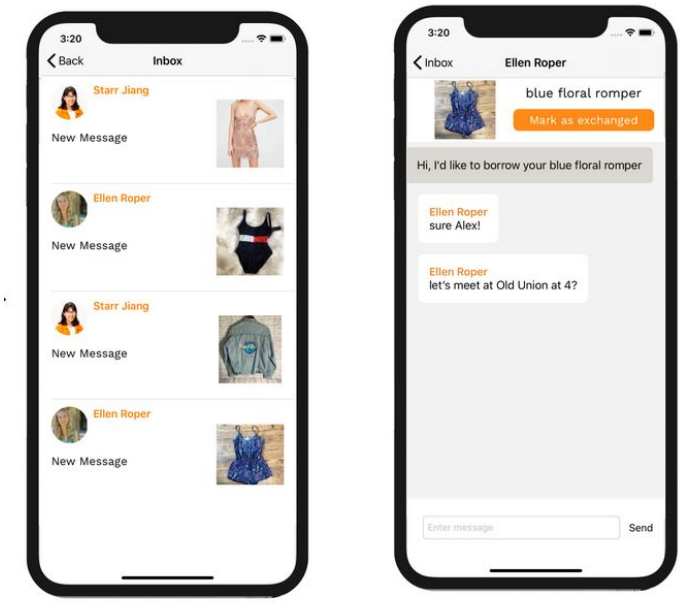



**Figure 25**. Final inbox feature

**Further Implementations**

One of the major implementations left is notifications. While notifications are an important part of an app experience, especially one that involves messaging and reminders, we decided to prioritize better functionality and user interface of browse, upload, and closet first. We had a lot of design changes to make and these took the greatest part of our efforts. We also had some problems with constraints (for example, the item preview at the top of each chat is too large for the screen), however since these do not hinder the functionality of the app, we prioritized other more important interface/functionality changes. Similarly, the UI of the chat previews in the inbox page is not exactly how we envisioned it (see figure). We simply did not have time to refine it with all the details we desired; however, these details are already being stored in our database and would just need to be hooked up to the app. Finally, we are having some loading issues with reading from

the database, which we believe has to do with compressing and caching the images but which we have yet to finish. We are also currently using Wizard of Oz techniques for the settings page and the due date on borrowed items; however everything else is completely functional.

**Tools we used**

We used Figma for prototyping, which was a very useful tool for laying out our UI designs. For our front-end, we used XCode and Swift, which presented a challenge as most of us did not have extensive programming experience. We found that this programming environment was adequate for our needs, but many of us spent nearly as long trying to figure out how to use it as we did actually coding. Finally, we used Firebase as our backend, which was fairly intuitive to use but which was not free; we quickly ran out of storage and had to pay for extra space.

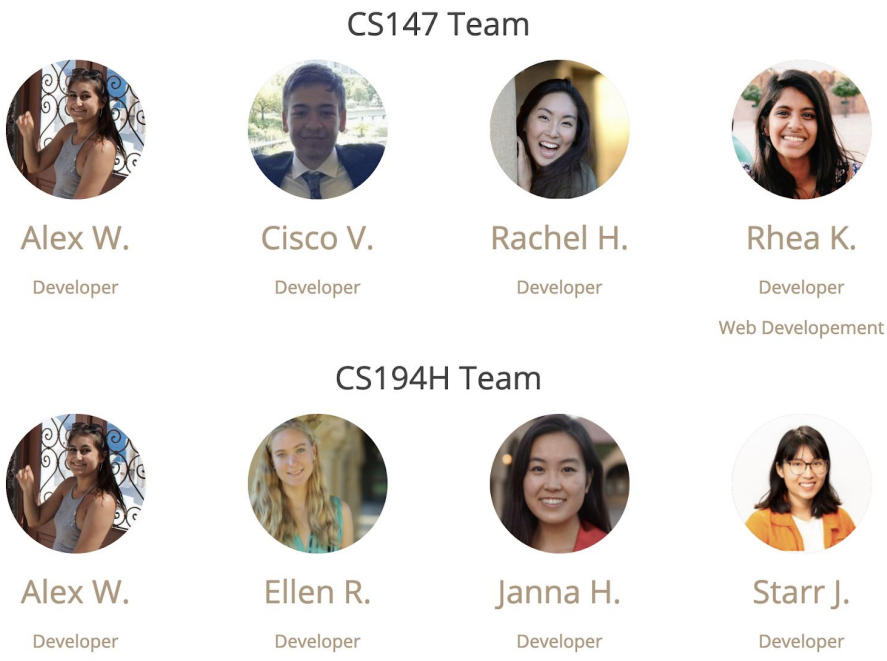
**Download**

https://github.com/asweitzman/armoirdemo

## VII. Making it Real

**Team**

Armoir is a product of two quarters of work and collaboration in teams. The project started in CS 147 with Alex, Cisco, Rachel, and Rhea. Alex continued to work on Armoir with Ellen, Janna, and Starr in CS 194H. Across these two teams, people brought their backgrounds and skills in coding, app development, visual design, and usability testing to the table!



**Business Model**

In early stages of growing the app's user base, we hope to attract new users through network effects and by offering 3 free clothing items per month. When the user goes beyond 3, we plan to take a 7% fee on customer exchanges, which is comparable to the cut on other similar platforms, i.e. Mercari. We will also offer a monthly subscription fee for access to premium features, such as limited-edition clothing items, covering cleaning fees, and access to brand-sponsored closets. We will also have partnerships with popular clothing companies to further grow our brand through pop-up events and clothing swaps.

We plan to start growing our user base from college campus communities, as they are the perfect combination of young adults (shopping on a budget, using fashion as self-expression, and tech-savvy) who live and work in close communities (to facilitate easy exchanges). We believe that college campus communities also better facilitate strong network effects of growing our platform organically. If users love Armoir enough, they will continue to use it even after college and bring it with them wherever they go next, thus further growing the market size. We envision that Armoir can rival top clothes shopping platforms like Facebook Marketplace or Poshmark, thus having a quite sizeable market size.

## VIII. Summary

Armoir sets itself apart from popular online clothes shopping platforms because it is a **peer-to-peer clothes lending platform** that leverages the power of community to help users build their ideal wardrobes. In the long term, we want to make an impact in people's everyday lives by connecting them to their local community through fashion.

With Armoir, we created a platform that enhances people's clothing options without breaking the bank. Fast fashion is the current industry solution to making cheap and trendy clothes accessible to people, but it is extremely environmentally wasteful. We see clothes-lending as a delightful and forward-looking alternative that promotes better self-expression through clothing, strengthens local community ties, and benefits the global environment.