GitHub Username: https://github.com/marinbcaro

Style Stumble

Description

Problem:

The user enjoys shopping, but finds it hard to come across good deals. He/she wants a way to be presented with different clothes and accessories and decide if it's something they like, and be notified when all of their liked items go on sale.

Proposed Solution:

Design an app that allows a user to find clothes and accessories that they are interested in. They will choose a category, and one at a time be presented with different options. The user can swipe the screen left if they dislike the presented item, or swipe right if it's something they're interested in. The items that are swept to the right will be added to the user's wish list, where they can browse all of the items they liked, and have the opportunity to purchase it on the retail site. Anytime an item that is on a wish-list goes on sale, the user will be notified of the sale price.

Intended User

This app is designed for women and men who enjoy finding good deals on products that they are interested in. The App uses ShopStyle API which allows client applications to retrieve the underlying data for all the basic elements of the ShopStyle website, including products, brands, retailers, and categories. The API is a REST-style web service, composed of HTTP GET requests. Data is returned to the client in either XML or JSON formats.

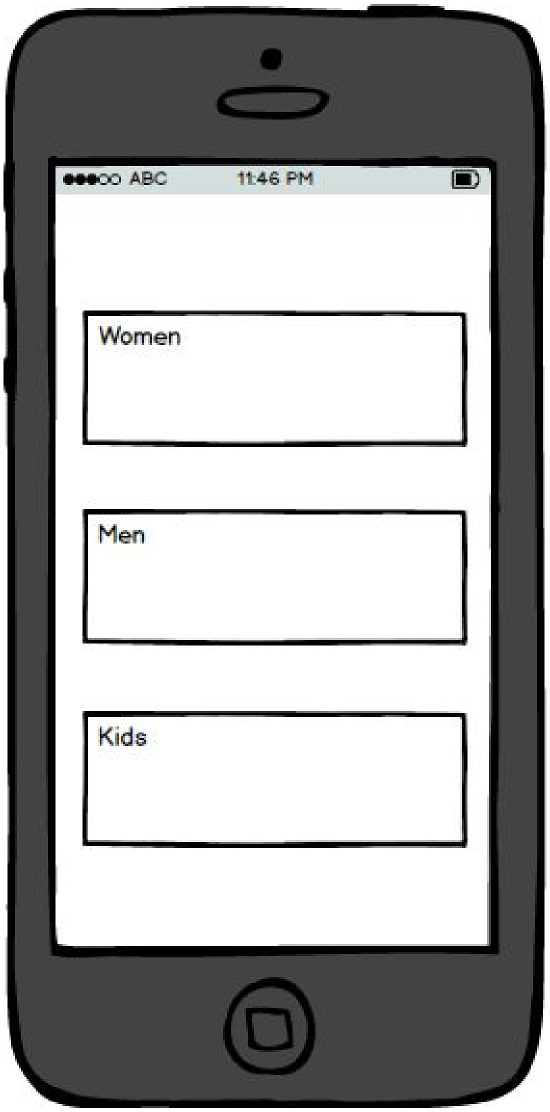
Features

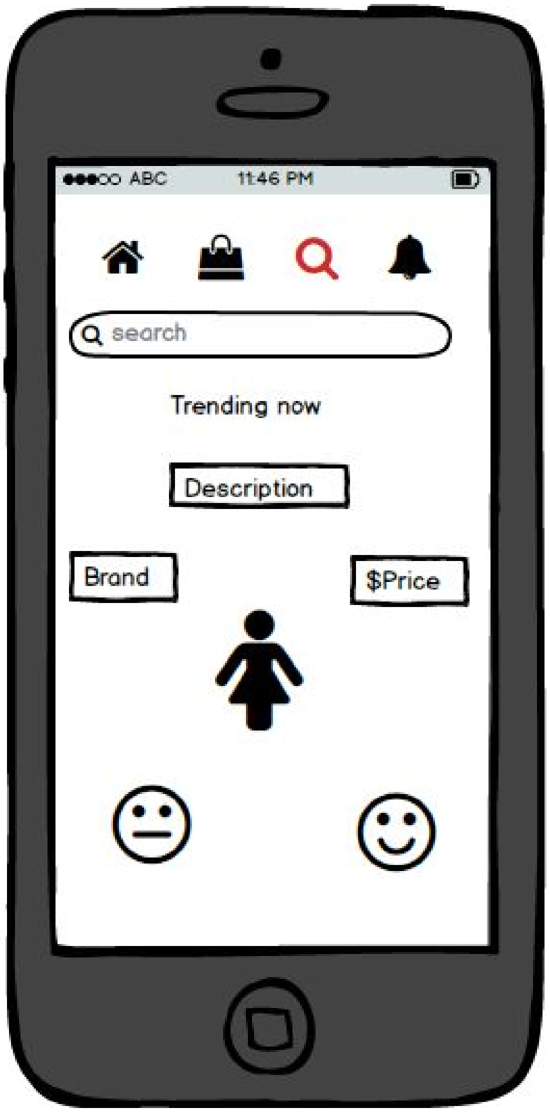
- Shows Information about products from different retailers and brands.
- Allows the user to select products, and save them in a wish list where they can get notifications if the product goes on sale.
- Allow the user to know which products are popular right now.
- Allow the user to search for different brands and product styles.

User Interface Mocks

Flow 1

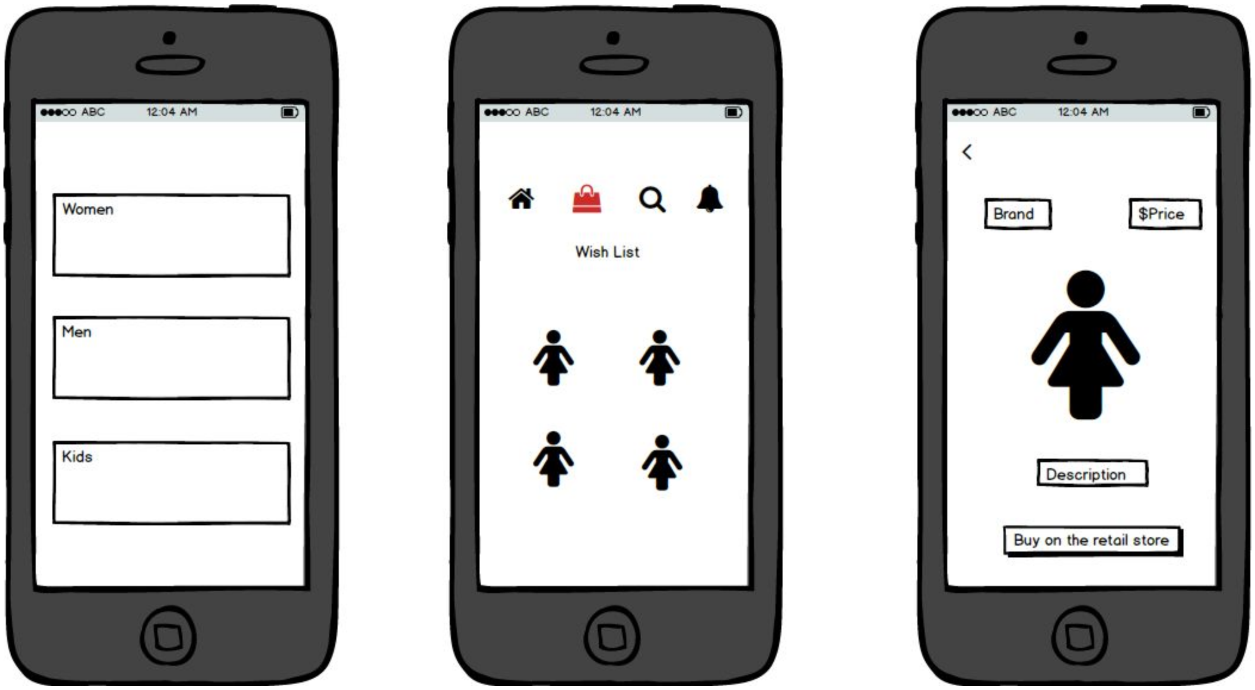
The user selects a category and is presented with a related piece of clothing or accessory. The picture, description, brand, and price are presented to the user. They can swipe left if they dislike the item, or swipe right if they are interested in it. Items that are swept to the right will be added to the user’s wish list.





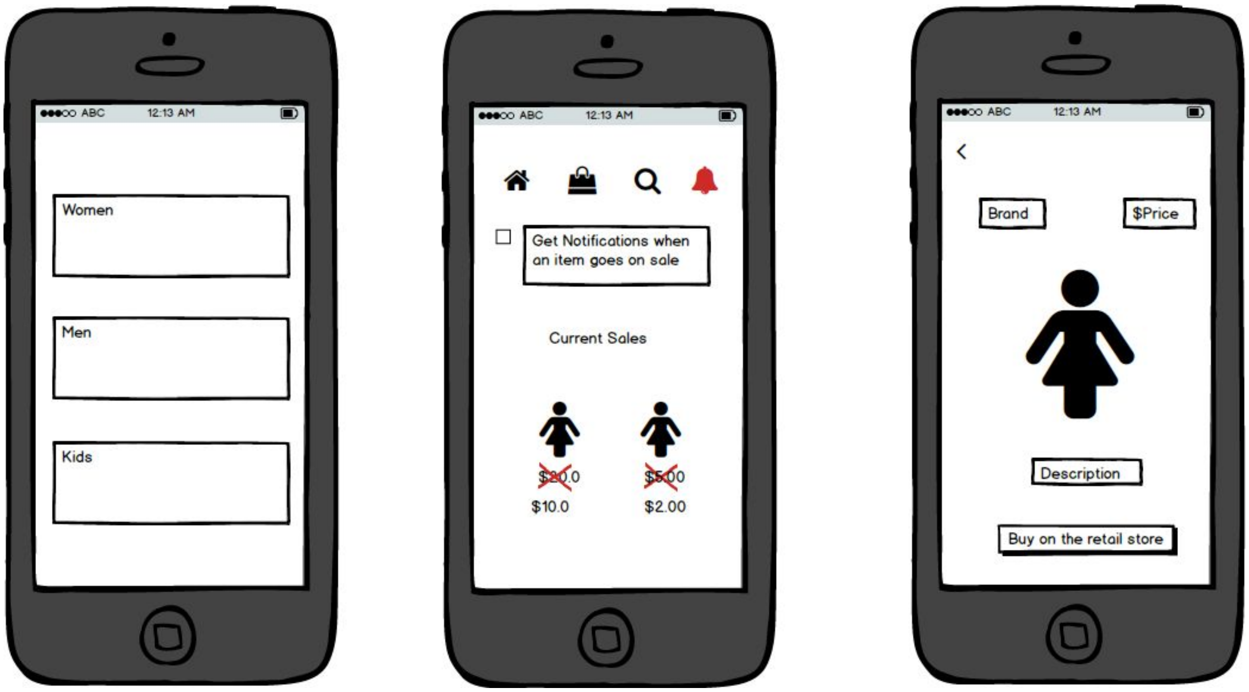
Flow 2

The user selects a category and is again presented with a related piece of clothing or accessory. At the top of the screen, there is an icon that will change the view and present the user with their wish list items. The user can select any individual item and be presented with the item's details, such as description, brand, and price. On the same screen, there will also be a button that directs the user to the item's retail website.

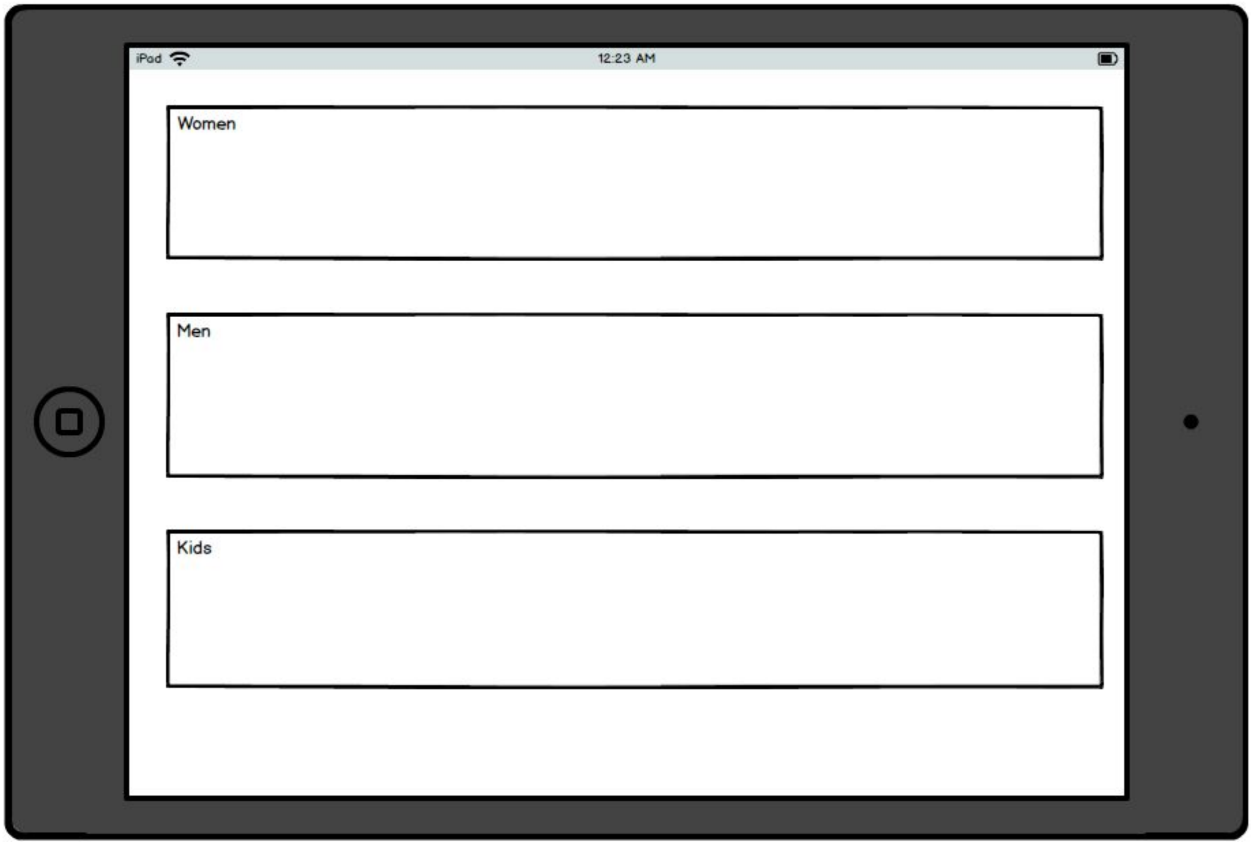


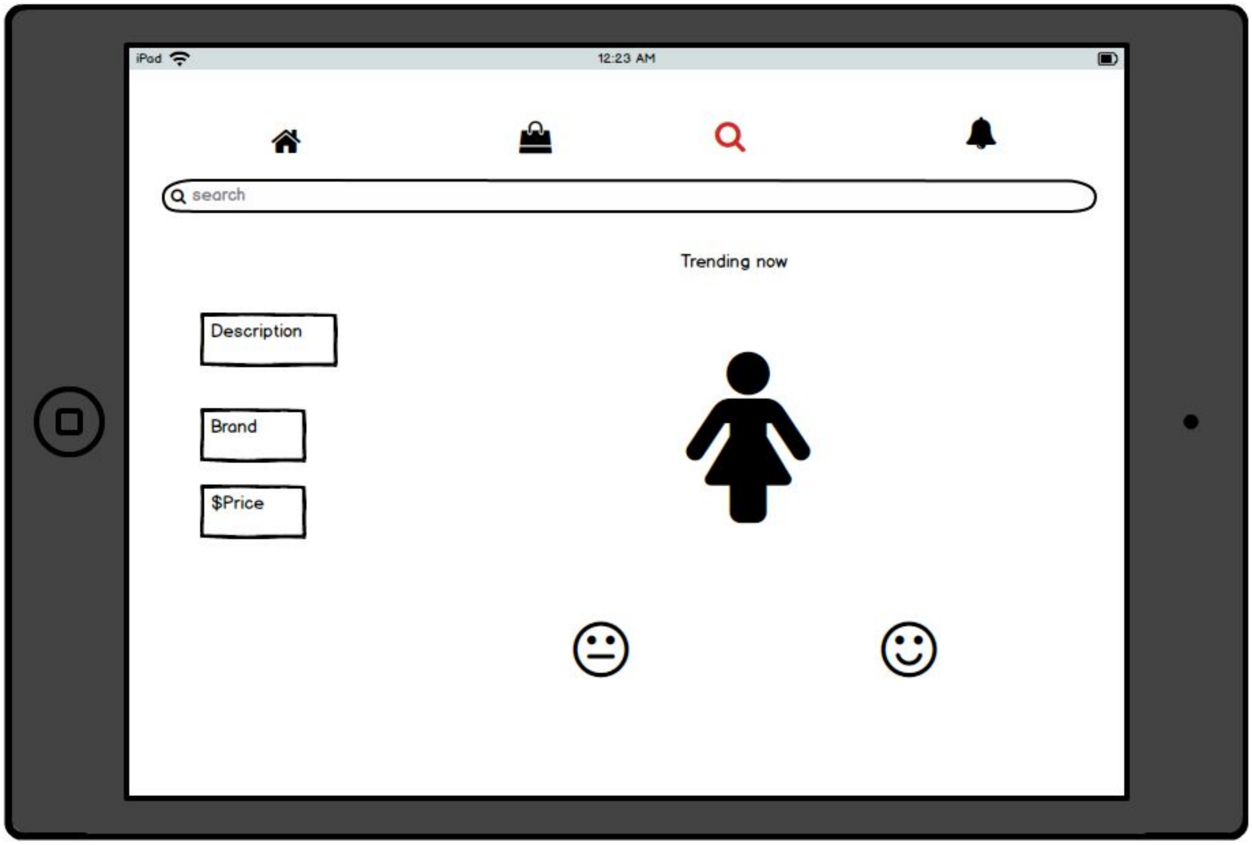
Flow 3

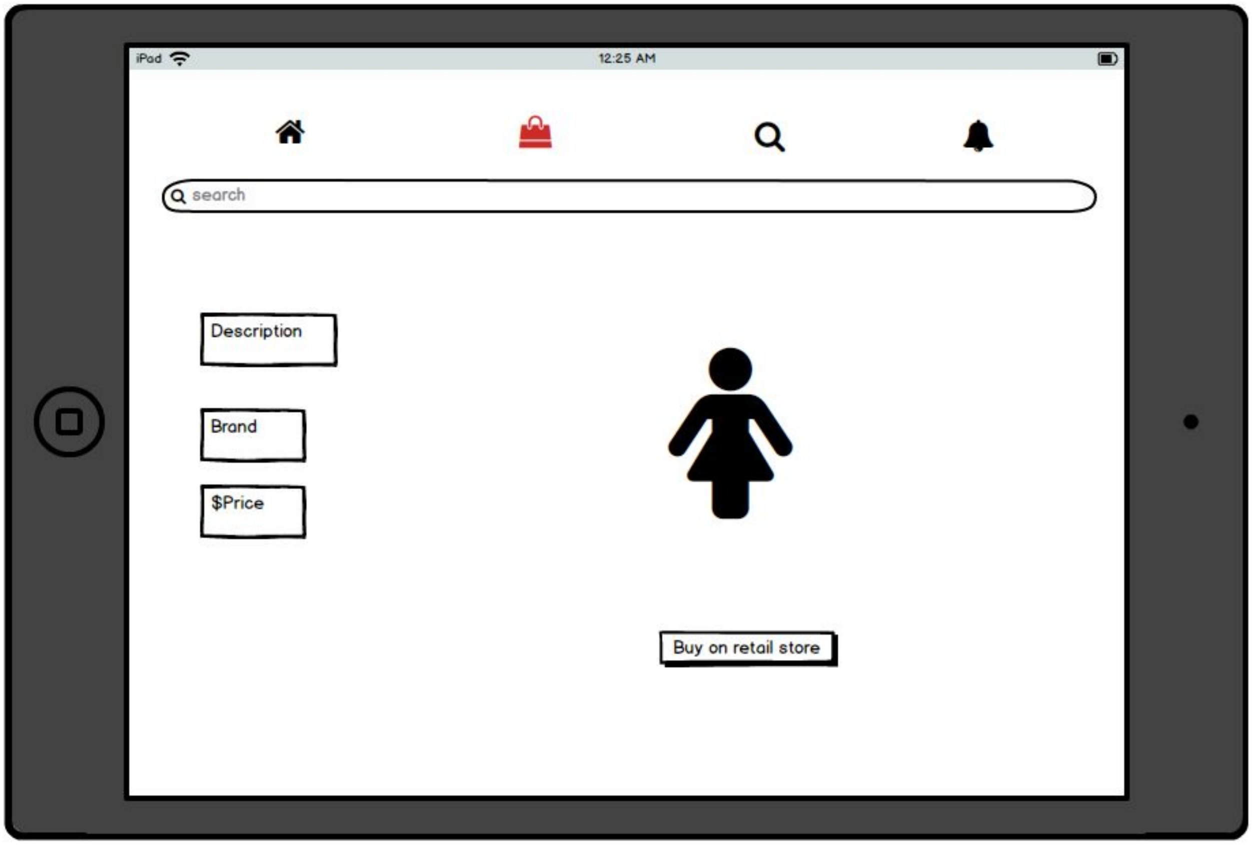
The user selects a category, and once again is presented with a related product. At the top of the screen will be another button that will take the user to the notifications screen. This screen will display which, if any, of the user's wish list items are on sale.

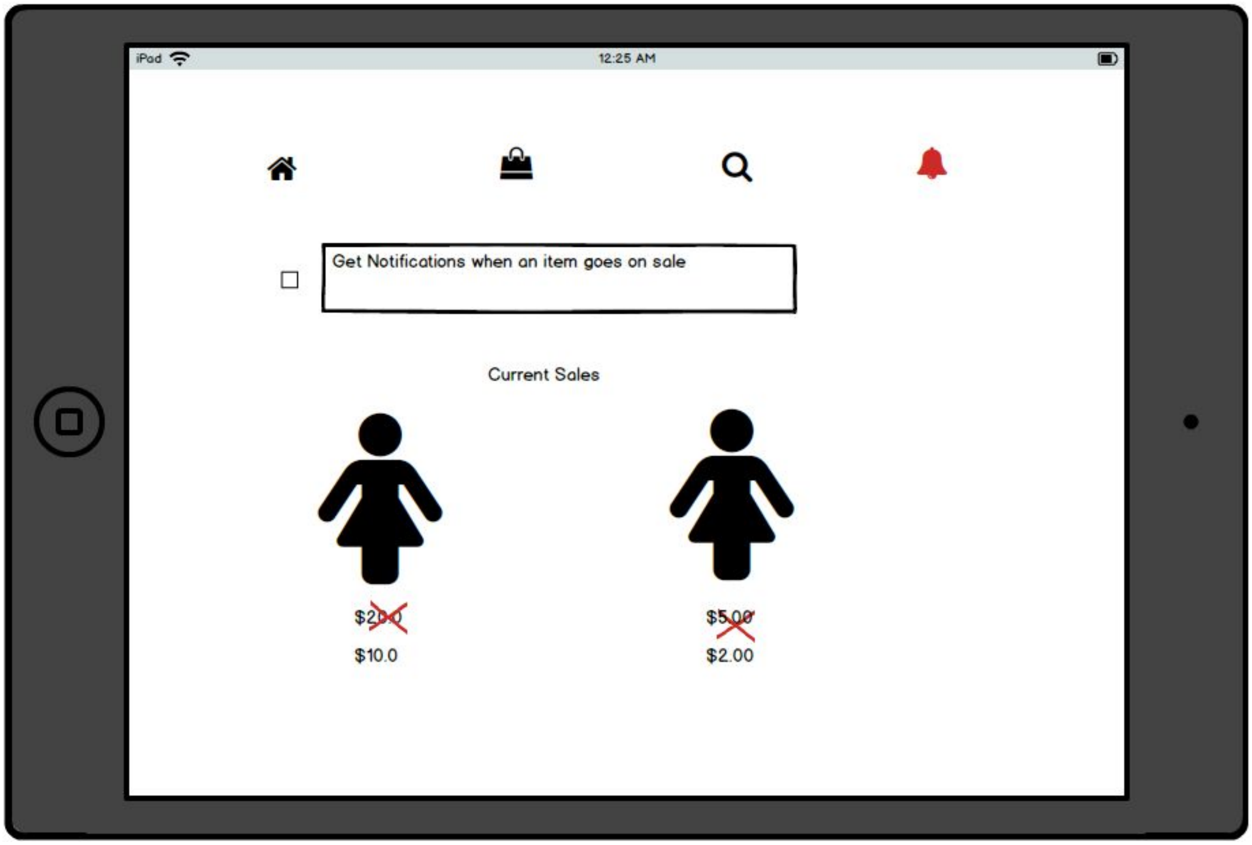


The tablet UI has the same flow as the phone the only difference will be the layouts. The following are the tablet screens without the description of the flows.




iPad
12:23 AM
search
Trending now
Description
Brand
$Price


iPad
12:25 AM
search
Description
Brand
$Price
Buy on retail store



Key Considerations

How will your app handle data persistence?

The app will use the library schematic to manage content providers. This library generates a Content Provider backed by an SQLite database.

Describe any corner cases in the UX.

At the top of the item screen, there is a home button which returns the user to the category selection screen.

Describe any libraries you'll be using and share your reasoning for including them.

The app will make calls to the Shopstyle API using the following libraries:

- Retrofit, to handle network calls.

- OkHttpClient, to handle timeouts, download sizes, and other troublesome network calls automatically.
- GSON, to map objects.
- Schematic, to handle content providers.
- Otto, to decouple different parts of the application.
- Picasso, to load and cache images.

Next Steps: Required Tasks

Task 1: Create project structure

The project will have a structure that will separate the data from UI to avoid code coupling. The idea is to have a clear definition of responsibilities in every part of the project.

Task 2: Setup libraries

Gather all of the libraries and place them in gradle file.

Task 3: Implement backend

Design and implement backend to get data from the Shopstyle API.

- Use retrofit and OkhttpClient to get data from API.
- Create a mapping of the objects using GSON.
- Create content provider.

Task 4: UI

Build layouts based on the mocks, and create fragments depending on the needs of the UI.

- Create UI for phone and tablet
- Add ad slot to layout

Task 5: Implement Google Play Services

Add the libraries required to display ads and analytics.

- Integrate libraries with UI to display ads.