Monica Chang

Programming Usable Interfaces

Homework 6 Reflection

Due 10/26/18 at 11:59pm

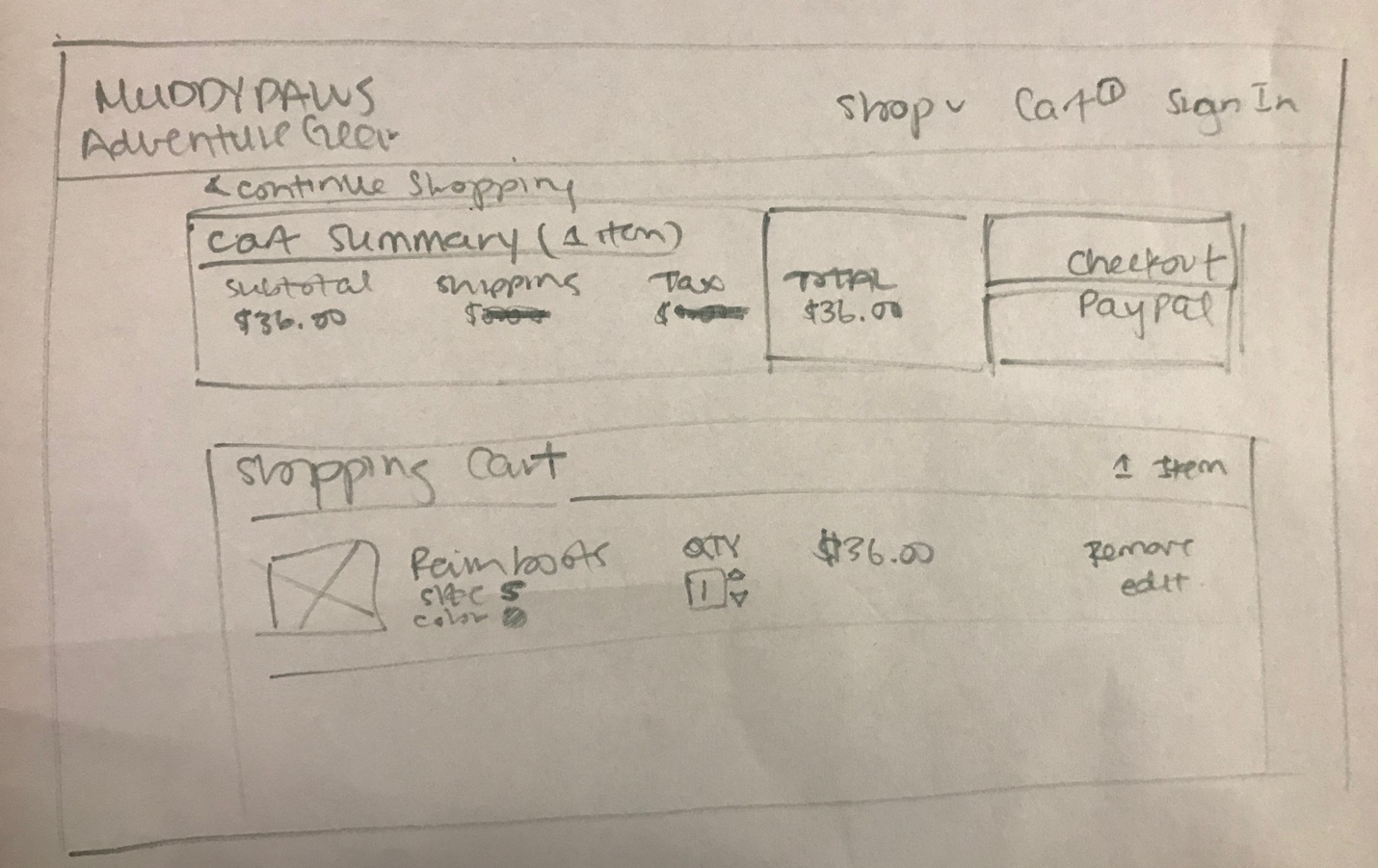
NOTE: To navigate the site, click on “Dogs”, then click on the item in the bottom right called “Hiker’s Fanny” (this has the cart implementation).

**Low Fidelity Prototype**

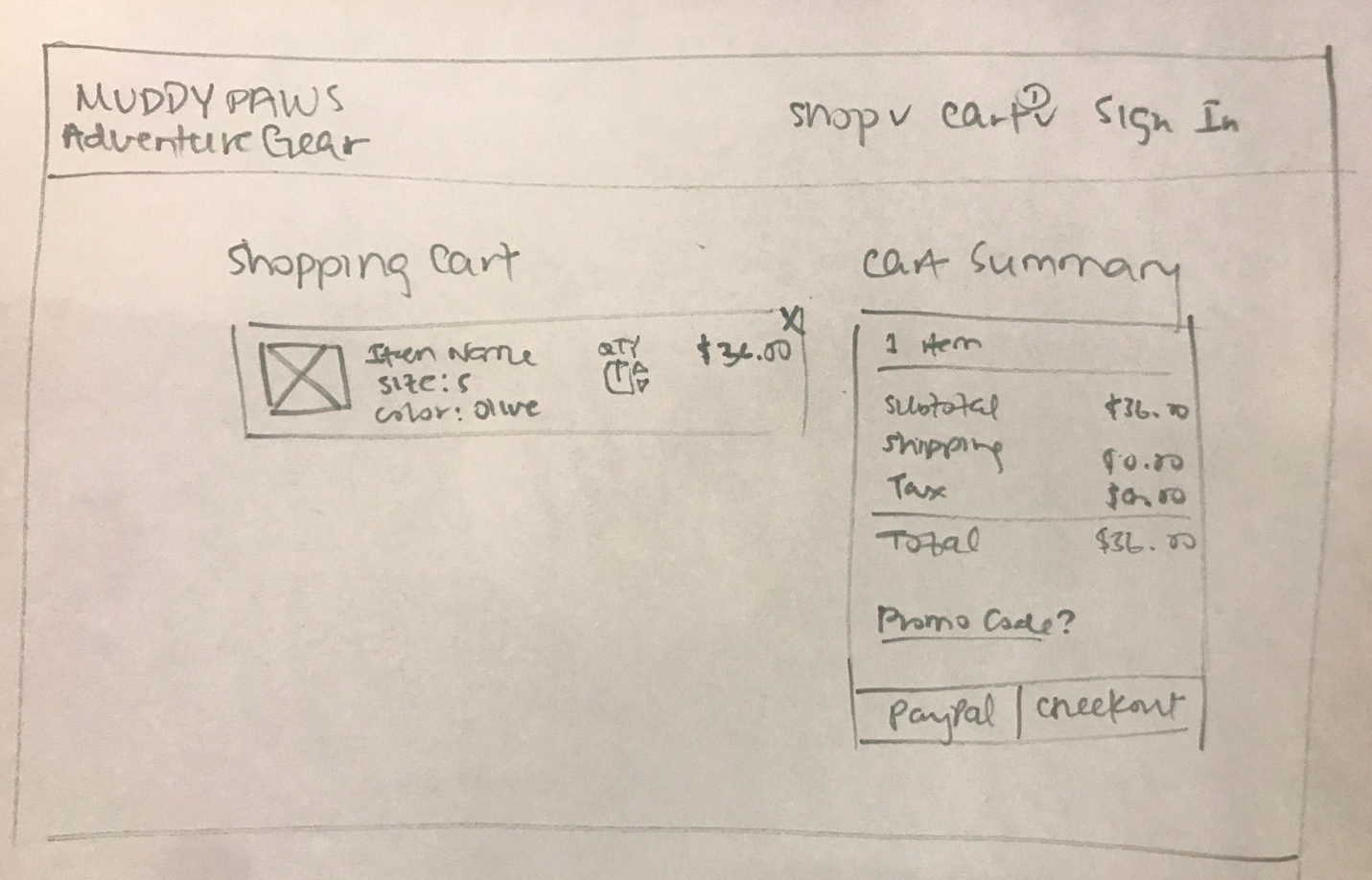
Design Choices:

* identified the shopping cart and cart summary as the two most important blocks in viewing the cart
* got rid of all navigation except for in the top navigation bar to encourage purchasing, but included a small “continue shopping” link on the top left
* changed from a stacking structure to a side-by-side view for the shopping cart and the cart summary because it allows you to view both at once (the shopping cart also does not need that much horizontal space)
* placed item specific properties such as name, size, and color in one column and separated out universal properties like quantity and price.
* allow for easy removal and editing of an item, decided that removal could be done at the top right corner of each item car and edit would be appropriate under the item-specific properties on a card
* show a clear breakdown of cart summary to fully inform users
* provide easy promo code and return policy access to encourage fast and non-hesitant purchase
* prominent checkout button to encourage purchase

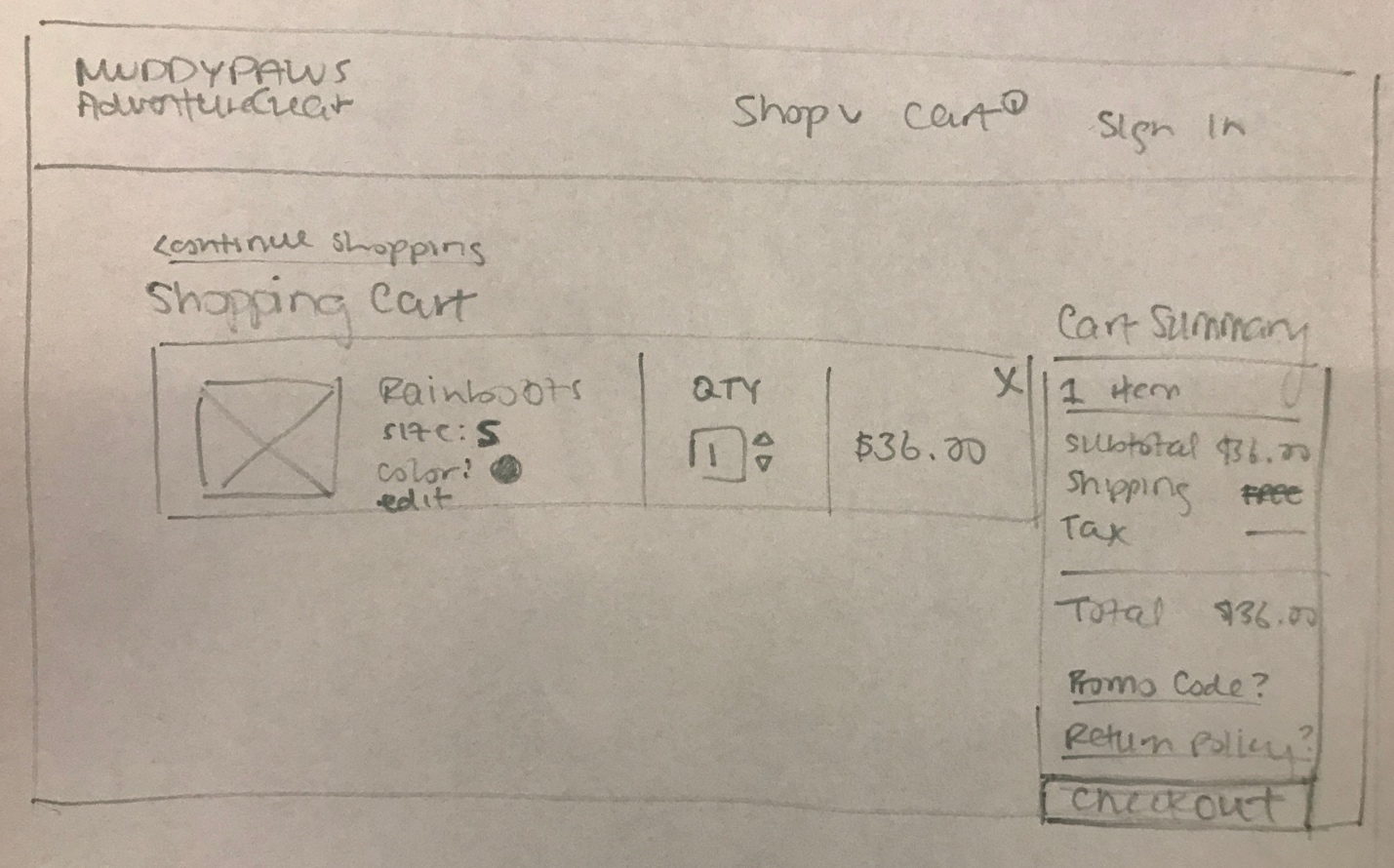
Iteration 1



Iteration 2



Iteration 3



**High Fidelity Prototype**

<https://invis.io/EKO5JW064SF#/320960614_Desktop>

Design modifications from the last paper prototype

* The edit button on the cart page is underlined and in a different color to make it more visible and differentiable
* I made the shipping and tax value $0.00 instead of lines so it doesn’t look like they haven’t been determined yet
* I added a PayPal button underneath the checkout button but is outlined instead of filled black so it is less prominent than the checkout button but still visible
* Added a wishlist section underneath the shopping cart to remind users of past items they saved and encourage them to buy them

**Reflection**

One of the challenges I faced was adding variables together due to type issues. This is because when I was trying to do something like incrementing the quantity value of an item, it would add the string “1” and “1” together, instead of incrementing it as a number. This took me awhile to debug and fix because I initialized them as integers but I didn’t realize that they would not remain as so. On a larger scale, it was challenging to anticipate what variables I would need and how to store them for future use. It was only after creating cart attributes, cart items, and a cart that it became clearer how to store these objects. Even though it was annoying to go back to modify variables, it was due to a lack of familiarity and part of the learning process. One of my biggest challenges was being able to load the data from one page into another page. I tried many ways of storing the data but I had a lot of trouble debugging my code when trying to retrieve the data after loading another page. I use a lot of console.log statements as well as the inspector to debug my code. it took my a long time to realize that I had to parse an array when getting data because printing the result looked very similar in the inspector, but they are actually stored very differently. After successfully getting all the data, I encountered a strange bug in my code that caused my cart items to be added exponentially, which caused my browser to crash. Based on how my item cards were being adding to the cart page, I finally discovered that because I was adding new items based on class names and not IDs, I was appending every new element to every previously created element with a certain class name. This is when I decided to use IDs to create each item card, but to keep class names for easy styleing purposes.