Window

High Fidelity Prototype

Authors and Roles:

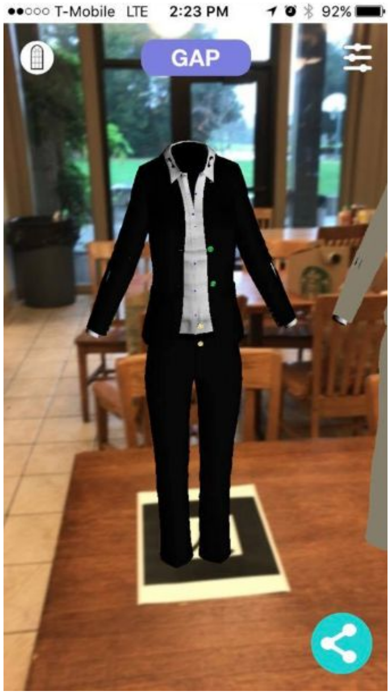
Max Freundlich: Team Manager, Software Engineering
Kristen Law: Software Engineering, Design
Elisa Lupin-Jimenez: Design, Usability Testing

Mission Statement

See where your style takes you.

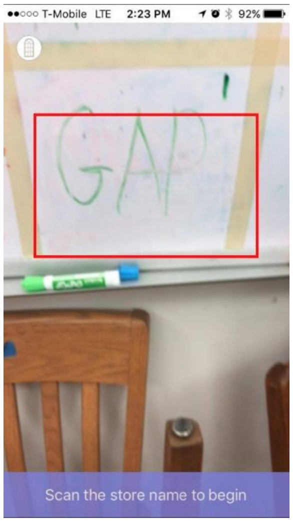
Problem/Solution Overview

Shopping is frustrating when you spend hours looking for clothes but don't like what you see. This can lead shoppers to feel disillusioned or despondent because they may have been hoping to find inspiration for an outfit or to find the perfect attire for a special event such as a date or an interview, but instead they are unsure of where else they can look. Using augmented reality on a mobile platform, we place a custom mannequin in the storefront window so you can easily view what a store has to offer. We give you the freedom to spend your time on what matters most to you.

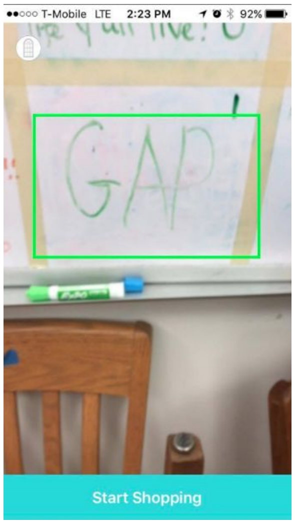


Tasks and Final Interface Scenarios

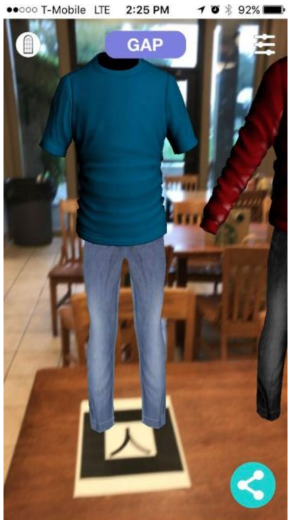
1. **Simple: Browse through a store's inventory.**
   We chose this task so that shoppers could easily access our application's main functionality. We were hoping that this would highlight any major UI flaws when they navigate through the basic screens of the application. We also wanted shoppers to grow accustomed to the application before challenging them with the more complicated tasks. See storyboard below.



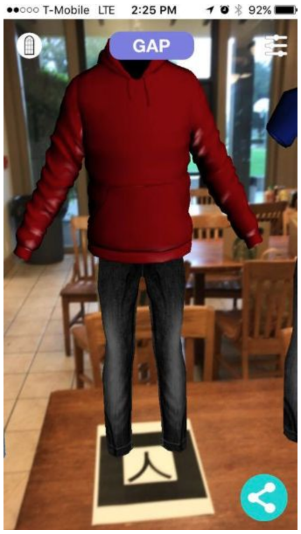
Click on the red square



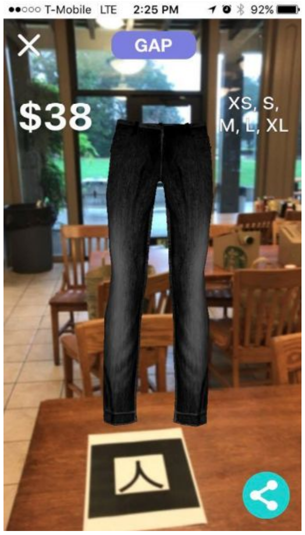
Click on “Start Shopping”



Swipe to the left

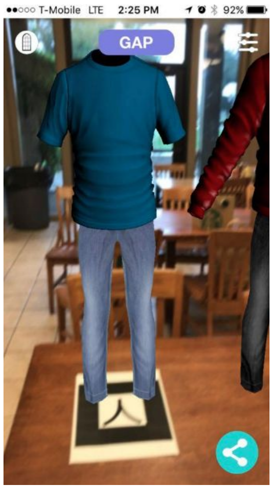


Double click the pants

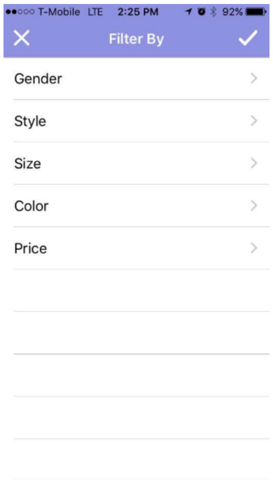


Done!

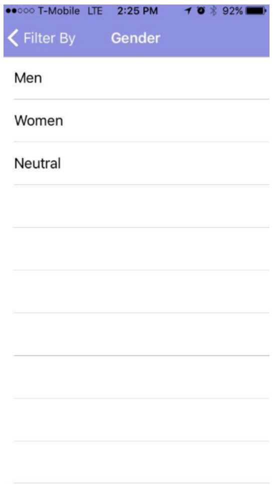
2. **Medium: Find a top for an interview that fits within a given budget.**
   We chose this task because we want our shoppers to use our filtering options to find formalwear and to check out the details of items to check for their prices. We hoped that the filter options would be intuitive enough for the users to spot before they manually swipe through the range of clothing available. See storyboard below.



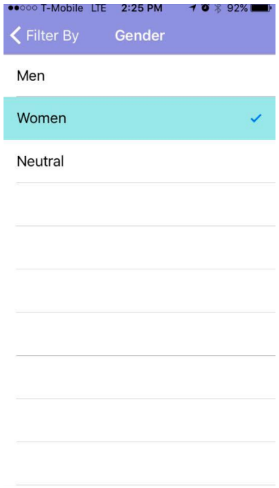
Assume the store has already been scanned. Click on the “dials” icon in the top right corner.



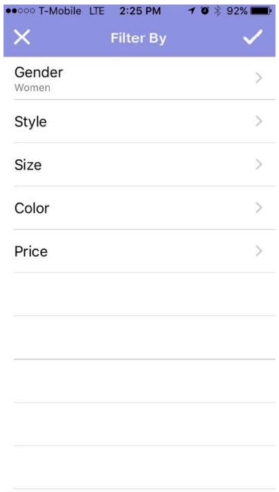
Click on “Gender” (I am female)



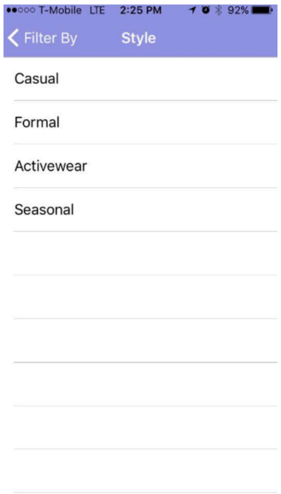
Click on “Women”



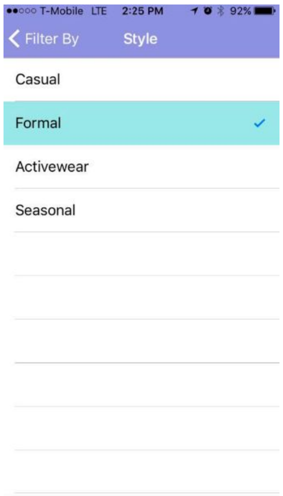
Click the back arrow



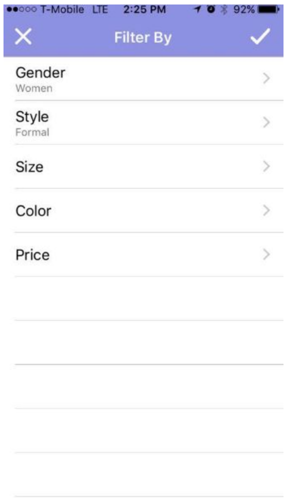
Click on “Style”



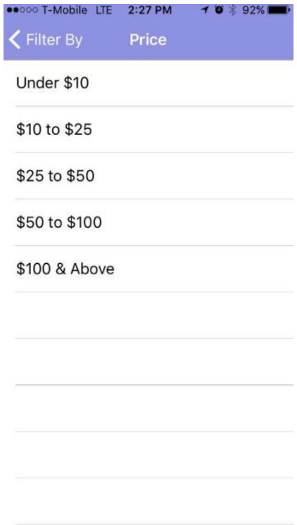
Click on “Formal”



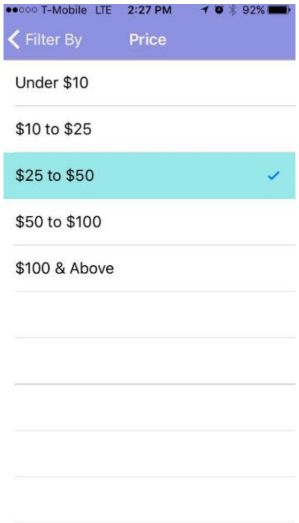
Click on the back arrow



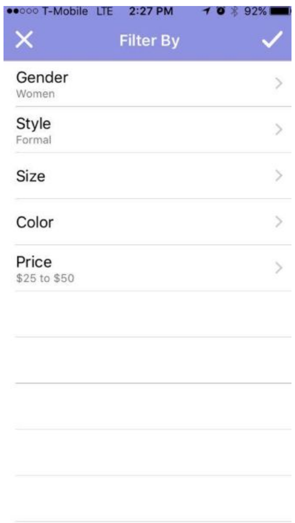
Click “Price”



Click on a price range



Click back



Gender
Women

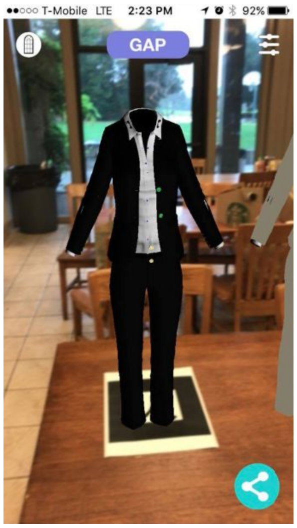
Style
Formal

Size

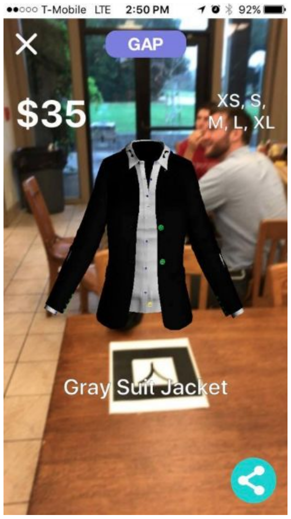
Color

Price
$25 to $50

Click the check



Double click the top



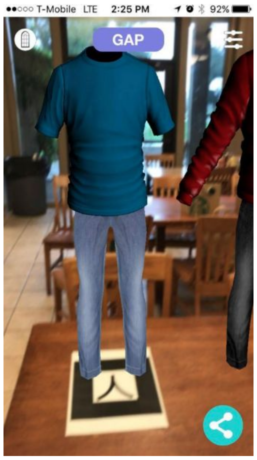
$35

XS, S, M, L, XL

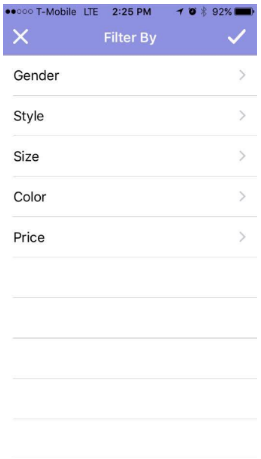
Gray Suit Jacket

Done!

3. **Complex: Send an outfit to a friend of the opposite gender.**
   We chose this task so that shoppers would use our gender filtering options to find an item, and then use our sharing option. We imagined that this task would be the most difficult because it combines both the simple task (browsing through a store) and the medium task (using filtering options), but adds on top of those tasks the process of sending an item to a friend. We hoped that this would illuminate any issues that users have with our interactivity features. See storyboard below.

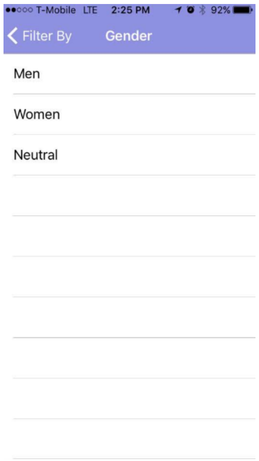


Assume the store has already been scanned. Click on the “dials” icon in the top right corner.


Filter By

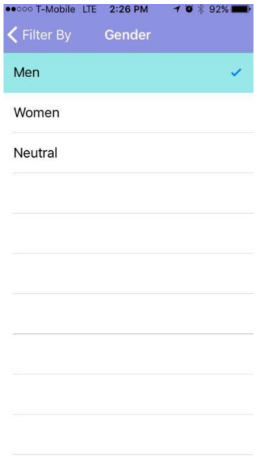
- Gender
- Style
- Size
- Color
- Price

Click on “Gender” (I am female)


Filter By — Gender

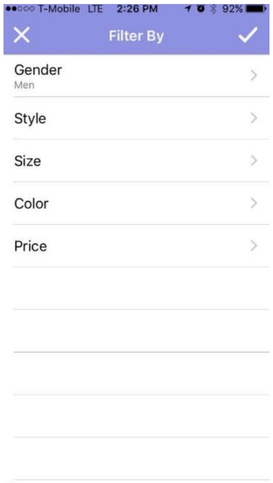
- Men
- Women
- Neutral

Click on “Men”


Filter By — Gender

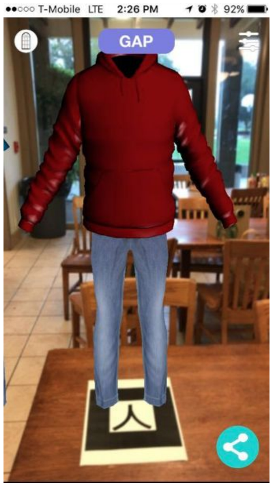
- Men ✓
- Women
- Neutral

Click the back arrow

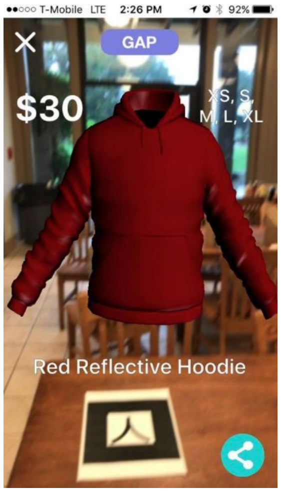

Filter By

- Gender
  Men
- Style
- Size
- Color
- Price

Click the check



Double click the top

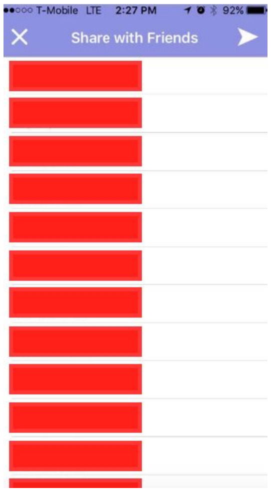

GAP

$30

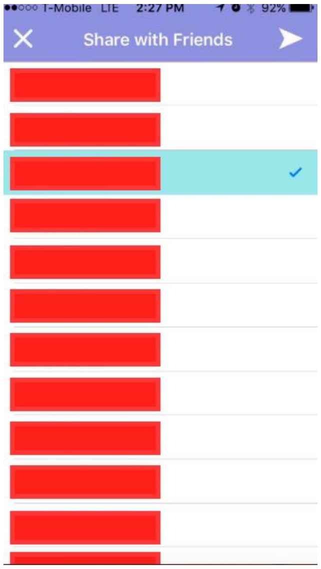
XS, S, M, L, XL

Red Reflective Hoodie

Click the “Share” button


Share with Friends

Select a friend



[redacted]

[redacted]

[redacted] ✓

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

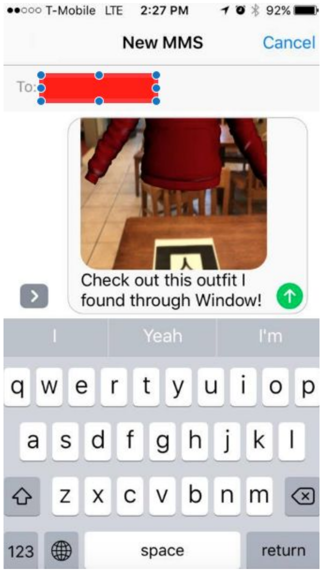
[redacted]

[redacted]

[redacted]

[redacted]

Click on the “Share” icon at the top right corner



To: [redacted]

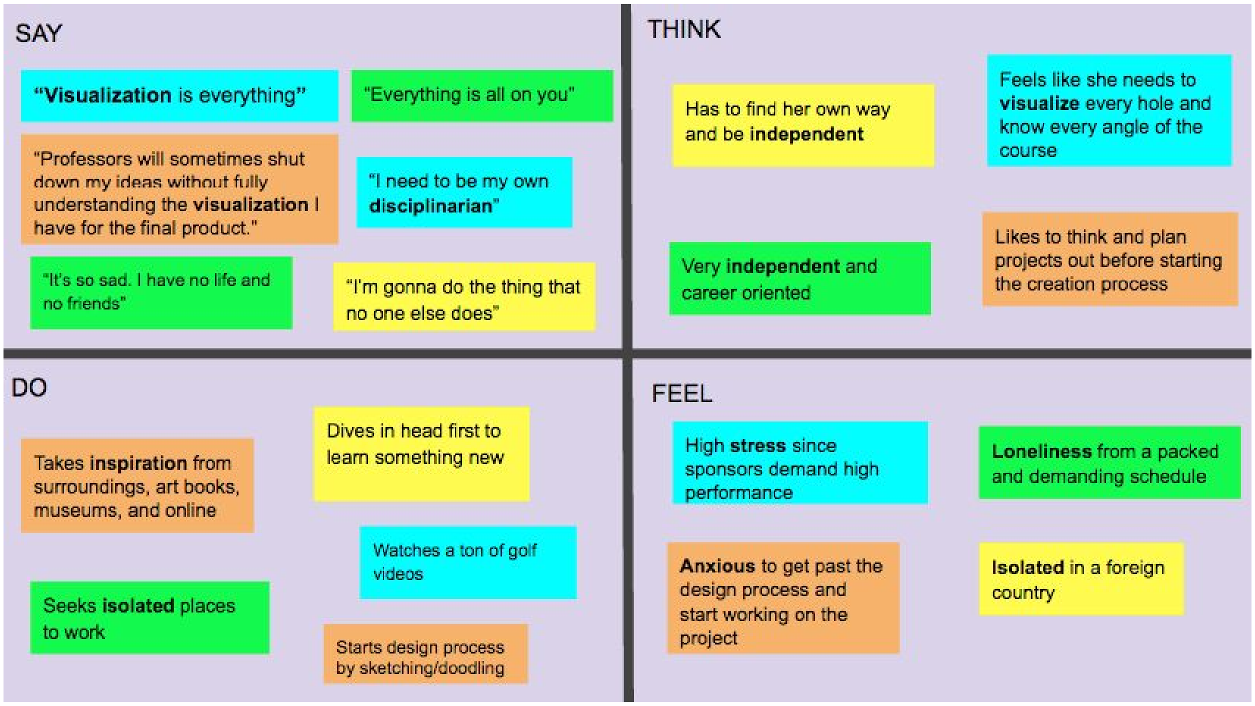
Check out this outfit I found through Window!

Done!

Design Evolution

Brainstorming Ideas

Initial Needfinding Map



SAY

"**Visualization** is everything"

"Everything is all on you"

"Professors will sometimes shut down my ideas without fully understanding the **visualization** I have for the final product."

"I need to be my own **disciplinarian**"

"It's so sad. I have no life and no friends"

"I'm gonna do the thing that no one else does"

THINK

Has to find her own way and be **independent**

Feels like she needs to **visualize** every hole and know every angle of the course

Very **independent** and career oriented

Likes to think and plan projects out before starting the creation process

DO

Takes **inspiration** from surroundings, art books, museums, and online

Dives in head first to learn something new

Watches a ton of golf videos

Seeks **isolated** places to work

Starts design process by sketching/doodling

FEEL

High **stress** since sponsors demand high performance

Loneliness from a packed and demanding schedule

Anxious to get past the design process and start working on the project

Isolated in a foreign country

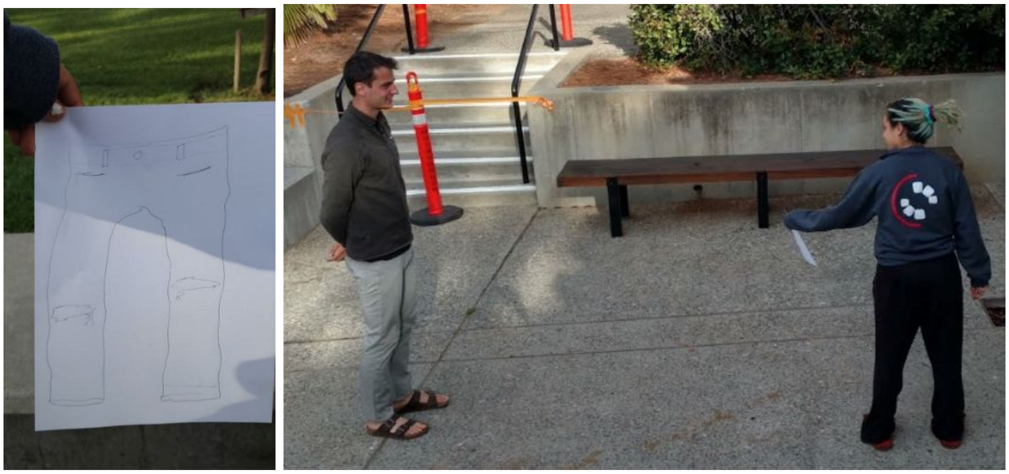
When we first came together as a group, we performed some initial needfinding to generalize the type of work that we wanted to do; we put together an empathy map (seen above) that informed us that informative visualization would make for a good foundation for an application. We wanted to explore informative visualization in regards to the shopping experience, so we went out to the Stanford Shopping Center to performing additional needfinding interviews. We interviewed several people from different backgrounds (e.g. a mom from Hawaii, a man from Sweden, a group of eighth grade girls, etc.), all of whom shared the common complaints about shopping. When we realized that the shopping experience (which many preferred simply because they wanted to be able to try the clothes on) was riddled with problems, we dove deeper by conducting point-of-view interviews with several more people such as a former employee at a retail store and a self-proclaimed "extreme shopper". From these interviews we found that the time-consumption of unproductive shopping among other problems were salient, so we drafted "how might we" questions to address various problems, such as the following:

1. HMW make it easier for people to find clothes online that fit?
2. HMW make shopping more bearable by gamifying the shopping experience?
3. HMW make it easier for customers to find items in stores?

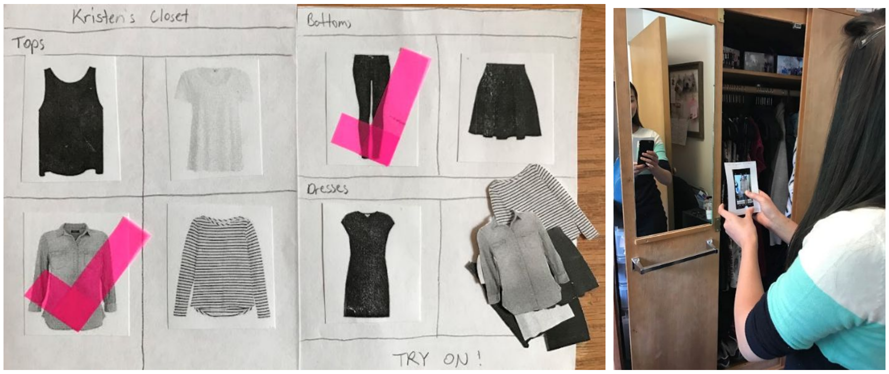
From this point we developed three experience prototypes as explained in the next section.

Experience Prototyping

EP 1: Try on virtual clothes



EP 2: Create a virtual closet



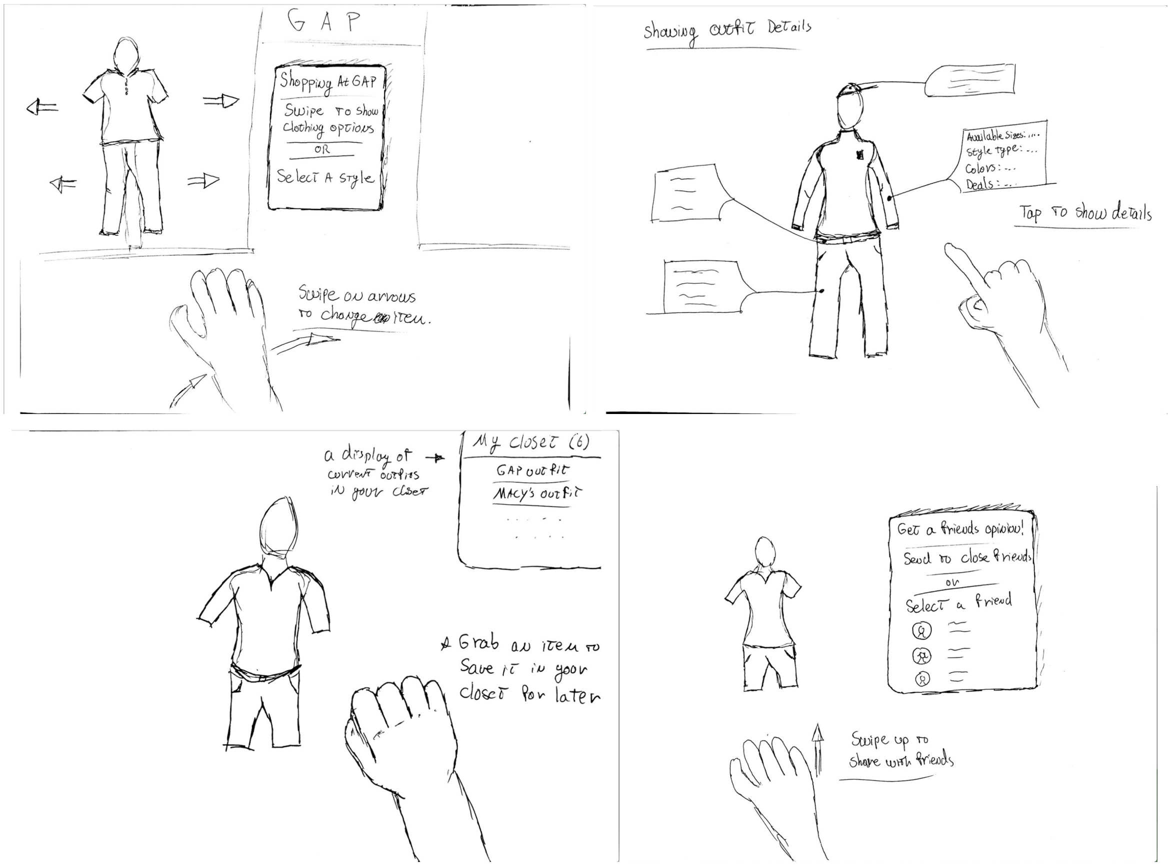
EP 3: AR Mannequin

We developed three different experience prototypes to explore the realm of shopping. We tested out using a virtual closet (where you could add your own clothes to the closet so your friends can see what you have), trying on virtual clothes from a store, and using an AR mannequin to display clothes. We went out into the field to have users try our experience prototypes, and after several iterations of user testing we decided that we wanted to pursue the AR mannequin experience prototype because it was the most intuitive prototype. Additionally, our user thought the idea was exceptional, and stated (without a prompt) that she would love to use something like it.

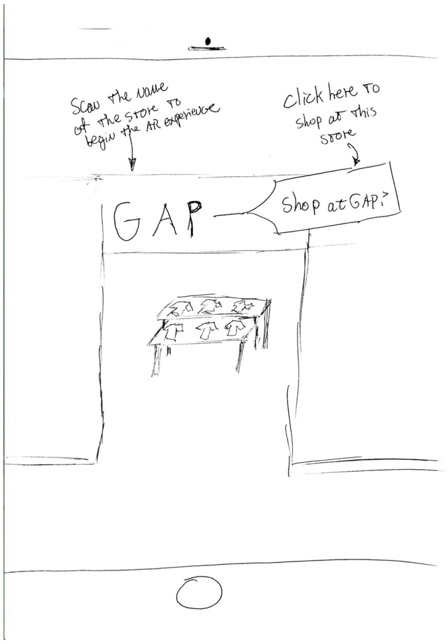
Initial Sketches

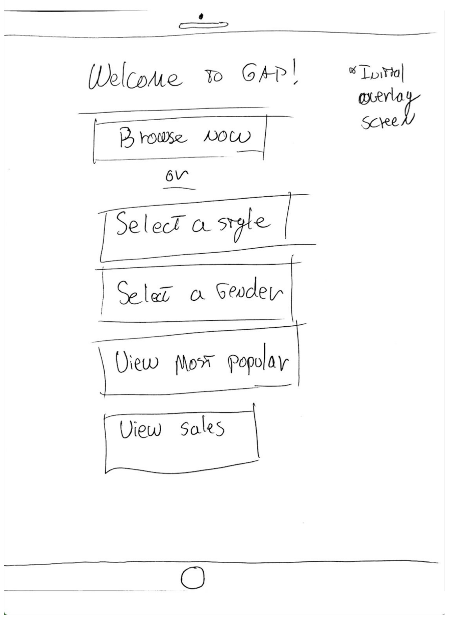
Once we had narrowed in on a problem and solution for shopping, we had to brainstorm different design mediums for our idea. We made sketches for our application on a headset, a mobile device, a projector, and a wearable (provided below).

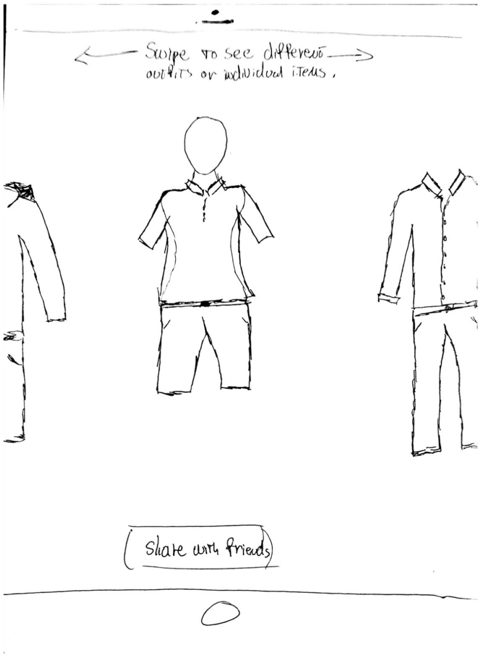
Headset

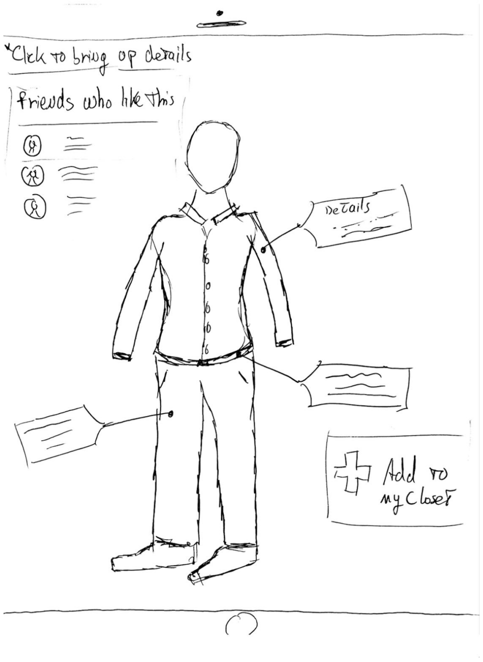


Mobile Device

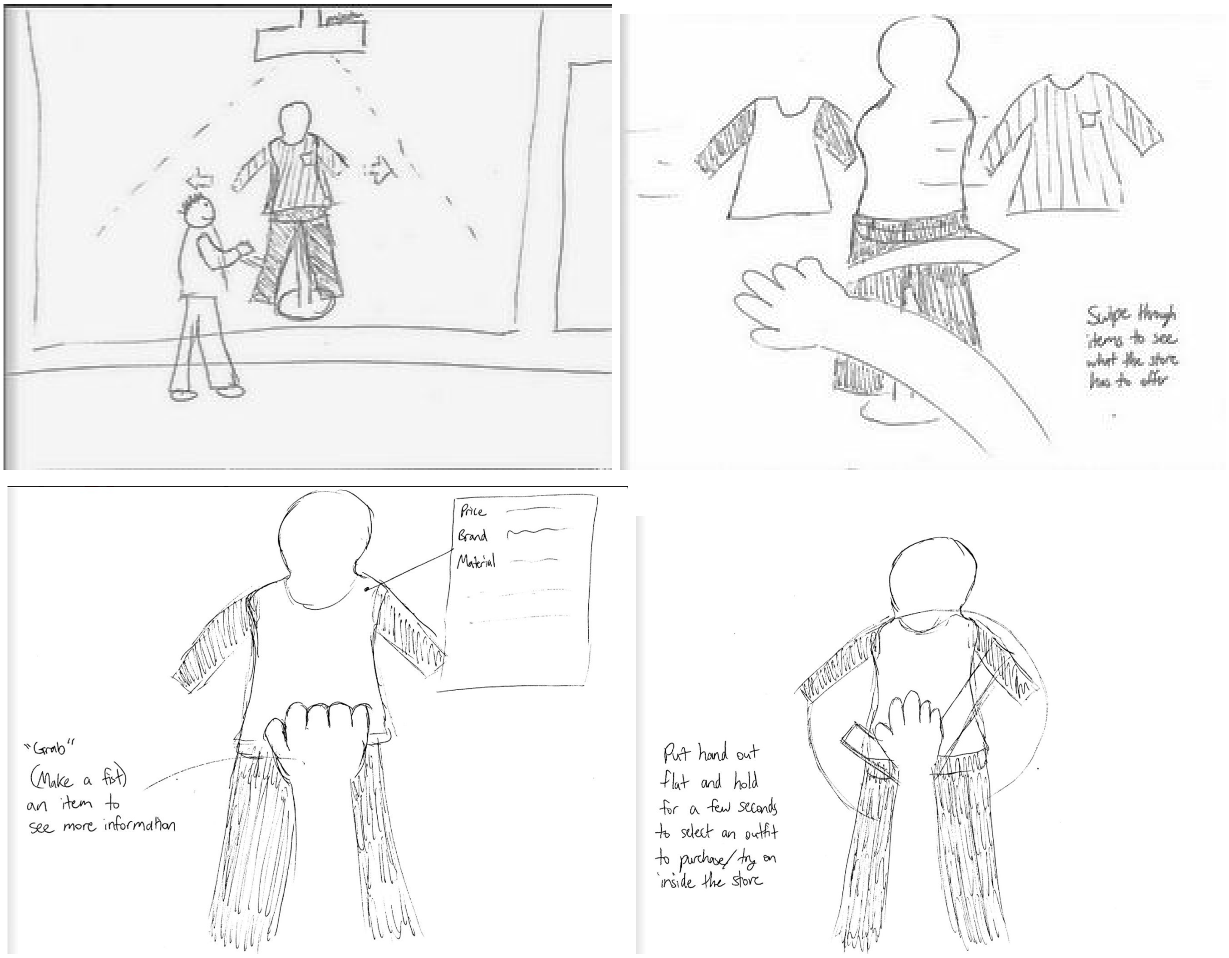

Scan the name of the store to begin the AR experience
Click here to shop at this store
GAP
Shop at GAP?







In-Store Projector

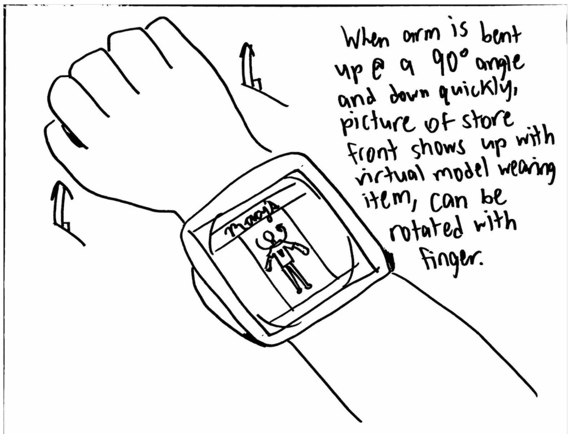


Swipe through items to see what the store has to offer

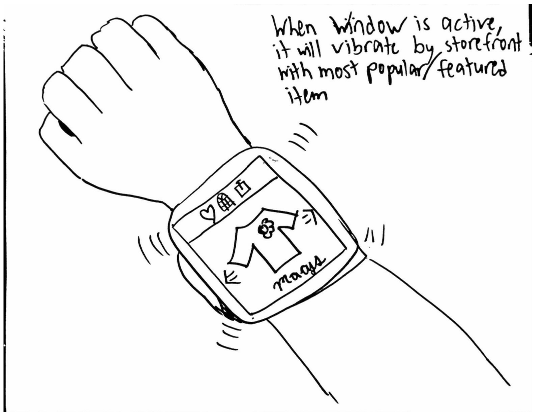
"Grab" (Make a fist) an item to see more information

Put hand out flat and hold for a few seconds to select an outfit to purchase/try on inside the store

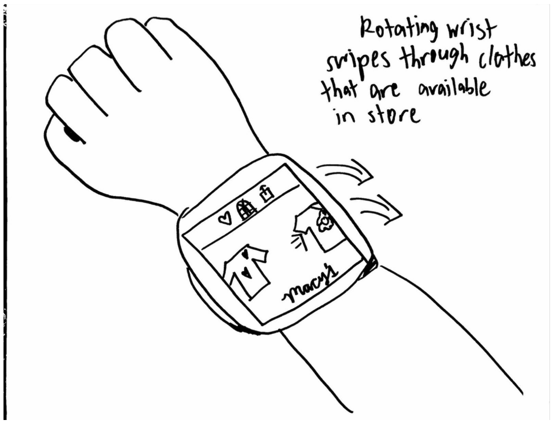
Wearable



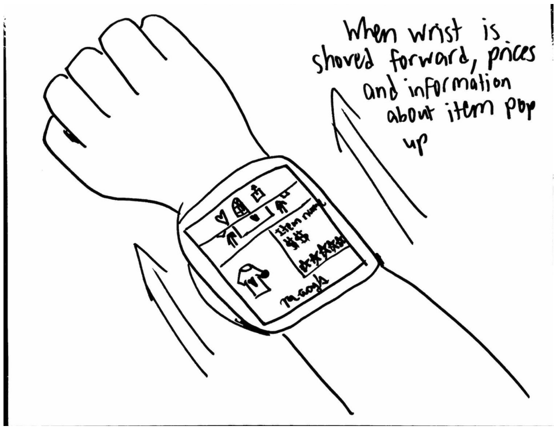
When arm is bent up @ a 90° angle and down quickly, picture of store front shows up with virtual model wearing item, can be rotated with finger.



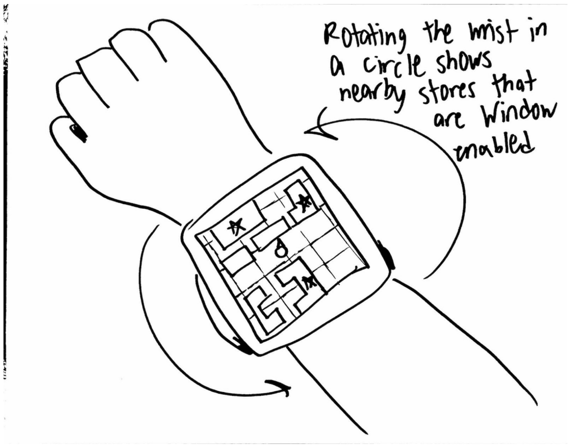
When Window is active, it will vibrate by storefront with most popular/featured item



Rotating wrist swipes through clothes that are available in store



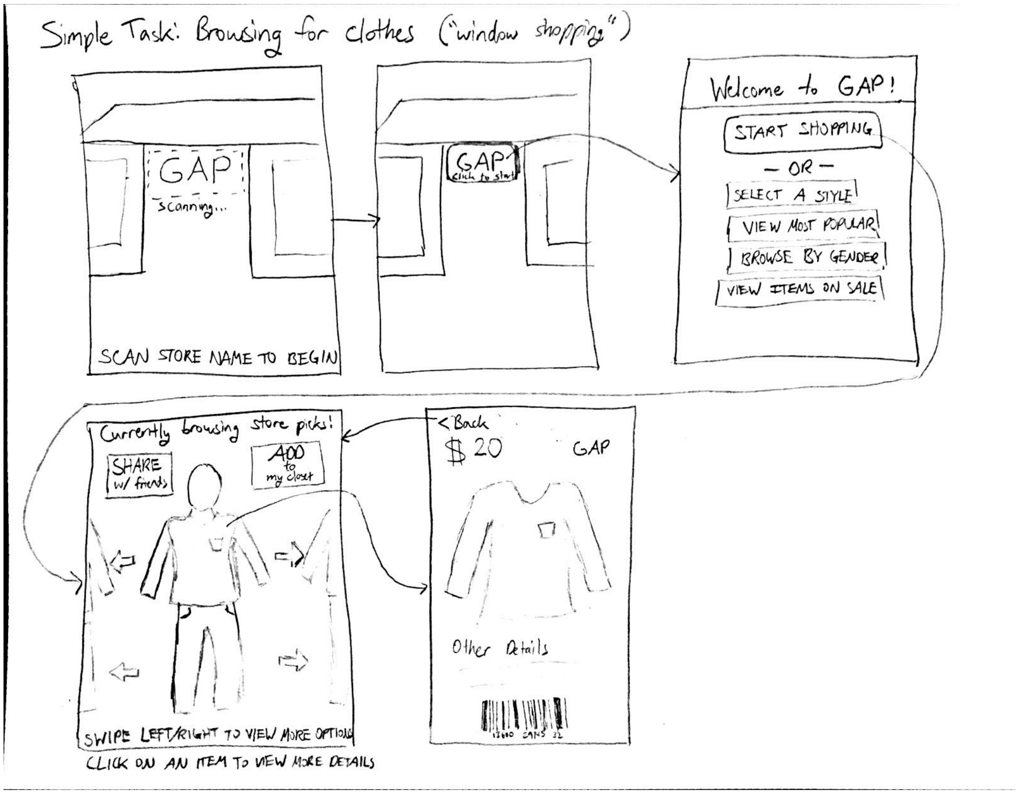
When wrist is shoved forward, prices and information about item pop up



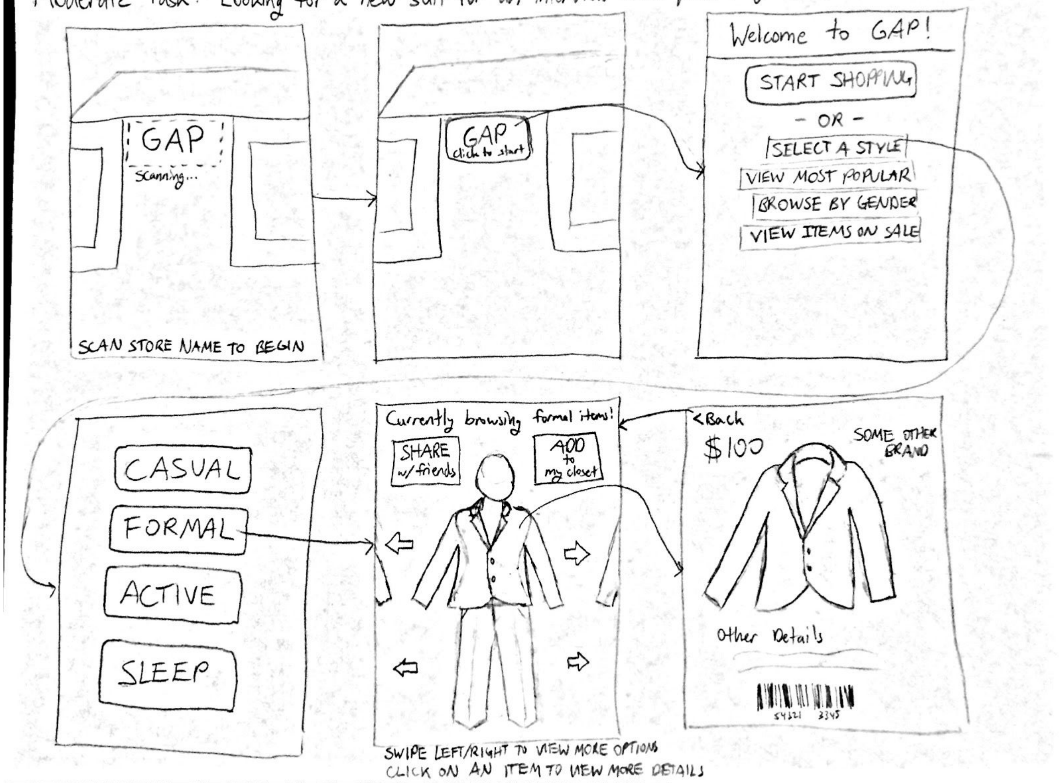
Rotating the wrist in a circle shows nearby stores that are Window enabled

After reviewing these initial sketches, we decided that we wanted to proceed with a mobile application for three main reasons: smart phones are accessible by exponentially more people than the other options, smart phones have cameras that enable augmented reality in a more comprehensive way comparatively, and smart phones lend themselves to social features that the other mediums do not.

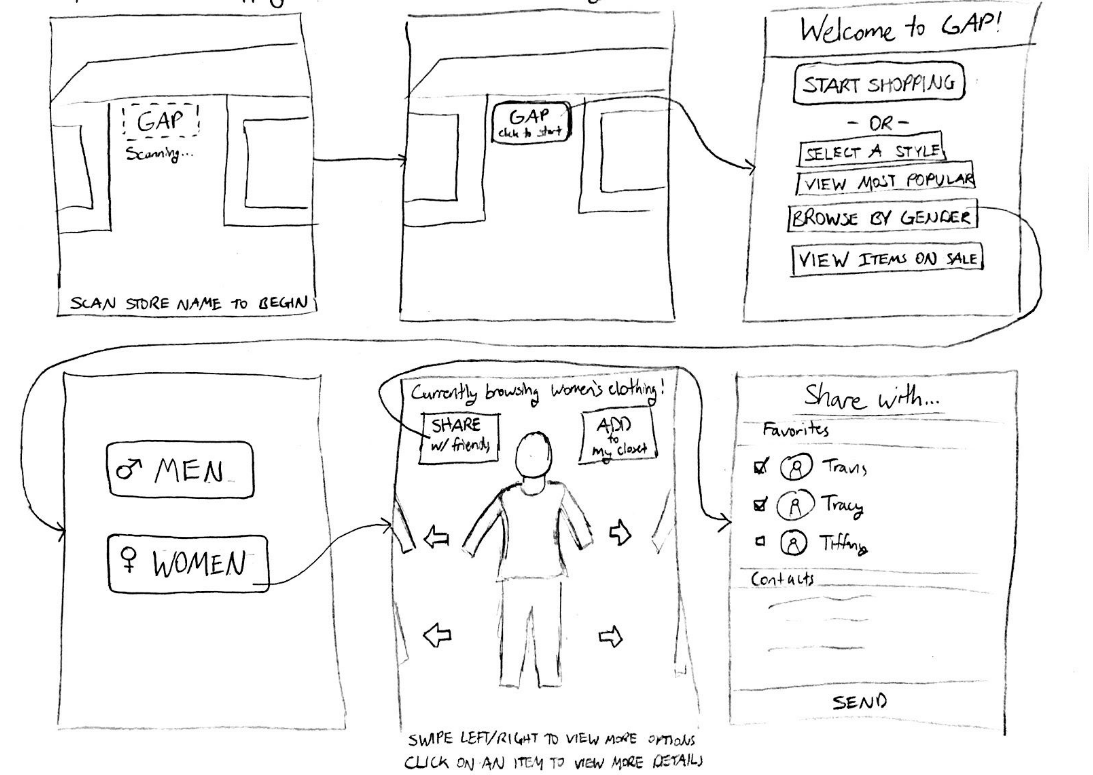
We then designed three tasks of varying complexities, which have not changed much from the tasks we presented at the beginning of this report. Below are pictures of the initial storyboards for these tasks.



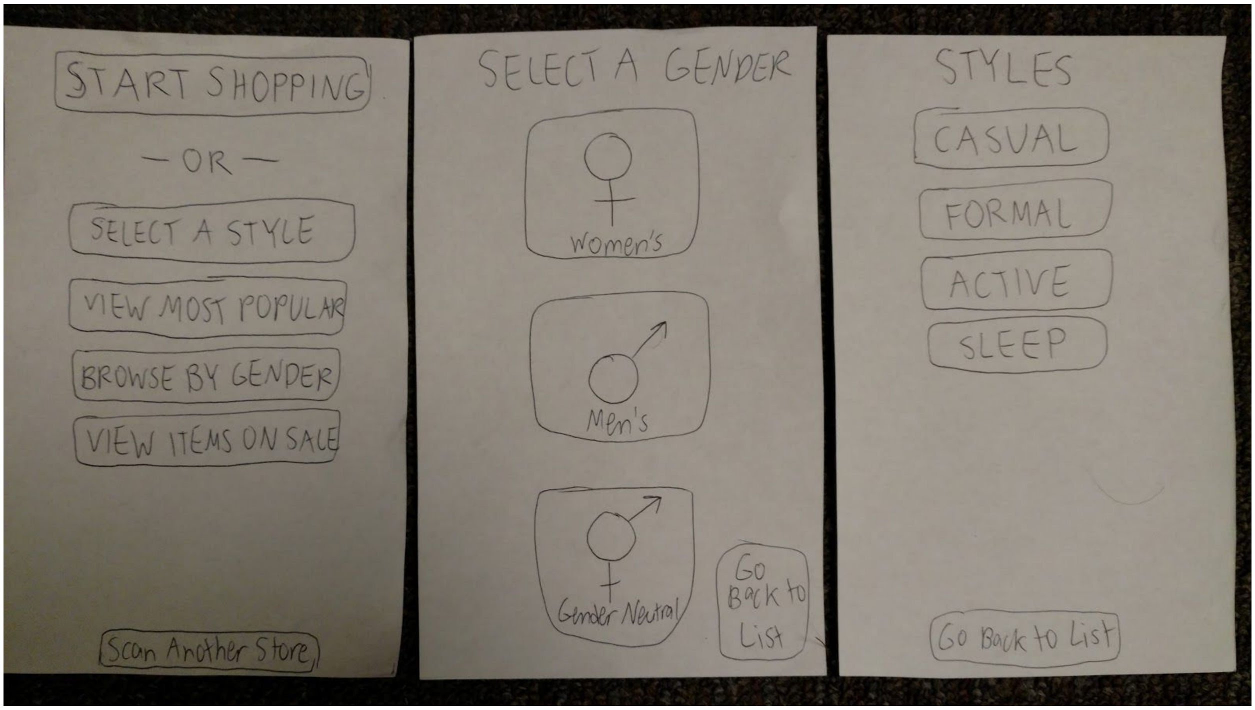
Moderate Task: Looking for a new suit for an interview within price range

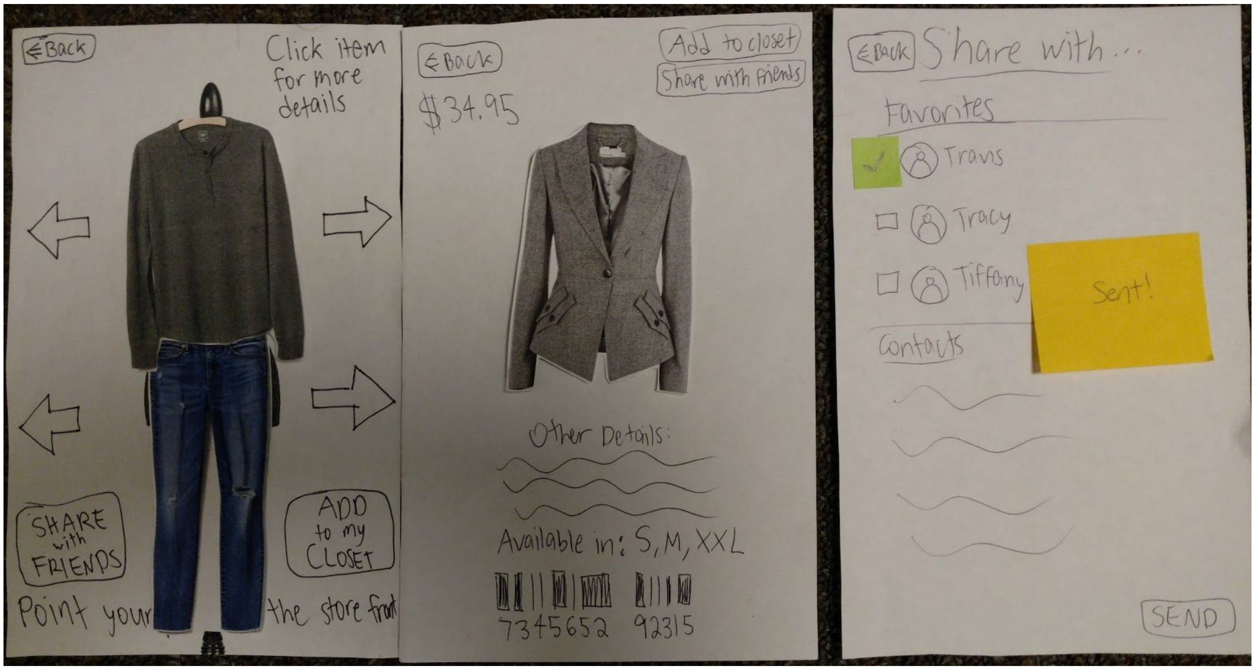


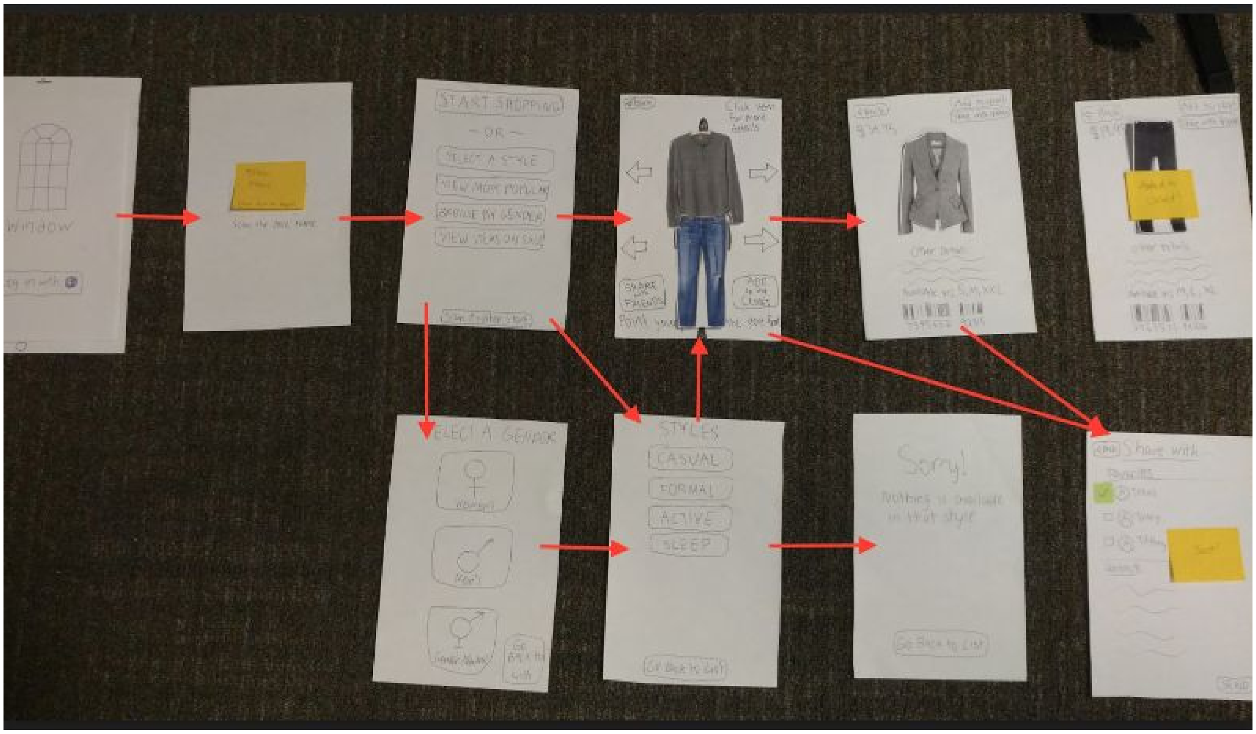
Complex Task: Shopping for someone else and asking for opinions on outfits

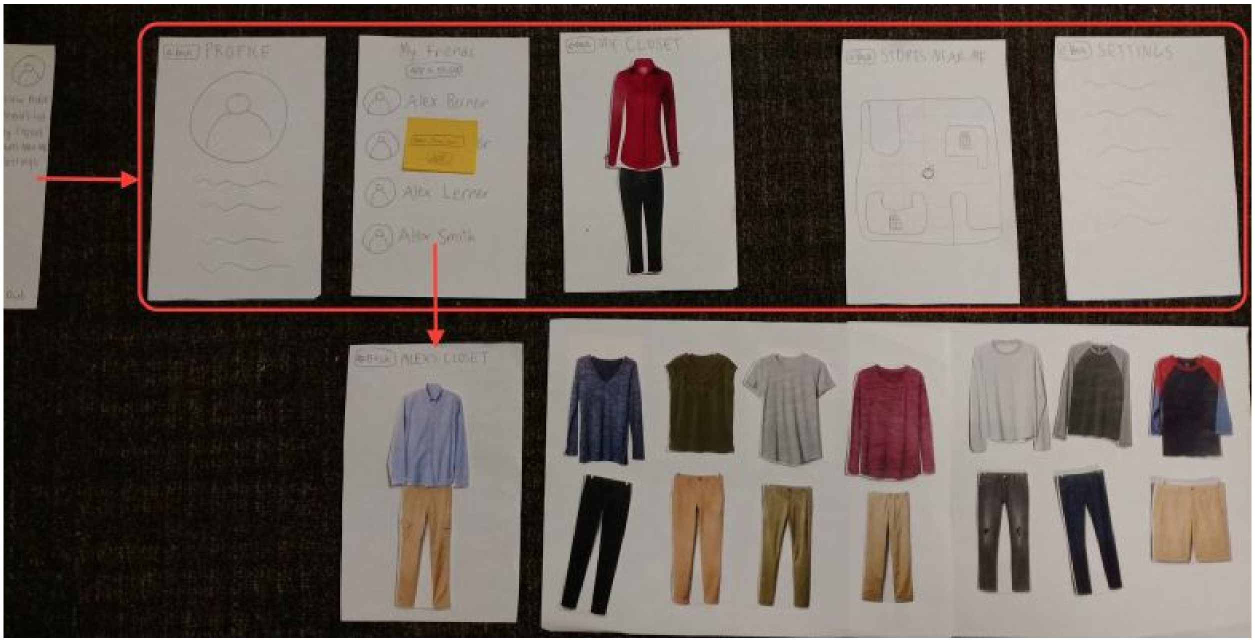


We developed our low fidelity prototype based off of these sketches. At this point we were only looking to piece together basic functionality, so we focused more on basic user interface than on aesthetics. Below is our low-fi prototype.









We chose to have the user start by scanning a store name, which would then land them on a home screen from which the user has the option to start shopping or to select for filtering options. Once the user has determined how they would like to browse clothing at the store they chose, the app transitions to a view with clothes on a mannequin. On this page, we are trying to convey to the user that this is an augmented reality experience where the mannequin on the page is actually viewed through a camera. They have the option to swipe through articles of clothing, to see more information about a particular item, add it their own virtual closet, or to send the outfit to a friend. From here, if a user chooses to click on an item they can see more details such as sizing options, prices, availability, and brand name. If they want to buy the item, a barcode is shown that will allow an employee to quickly identify the item and help them purchase it. Additional to the core features of the application, if the user swipes from the left of the screen a navigation drawer appears that allows the user to check the inventory of their virtual closet, see their friends, change settings, and logout of the app.

User Testing

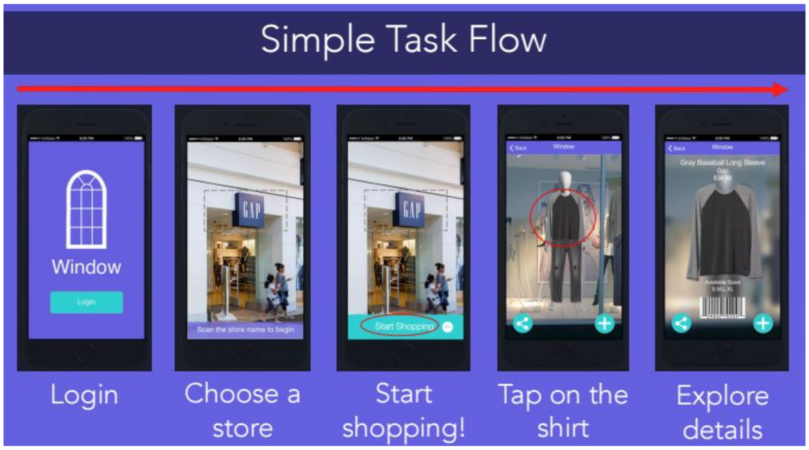
Once our low-fi prototype was completed, we returned to the Stanford Shopping Center to have random shoppers try our application. We assigned them the three tasks we had previously developed, then we allowed them to navigate the application by themselves, only assisting them when they initiated a user click (since the prototype was made of paper, we had to manually switch out screens and “clothes” for them).

We compiled the results from our testing to find that there were many improvements to be made. The main one was to get the user to the AR viewport faster; we originally had them scan a store’s name then click the store name to start the shopping process. This turned out to be confusing to our users since they wanted to start their AR shopping experience as soon as possible. Another change we made was to make it more clear that users are supposed to swipe to see more clothing options. Our users would often click the arrows to see other clothes instead of swiping on the arrows, and at times they wouldn’t explore their options all together. Because of this, we wanted to make swiping easier and more intuitive. The final big change we made was to our closet feature: it wasn’t clear to users that this was supposed to be used as a “favoriting” mechanism; instead, they attempted to purchase the clothes by adding it to their closet. As a result, we decided to rethink our closet feature, as it wasn’t integral to any of our task flows and seemed to be causing some confusion.

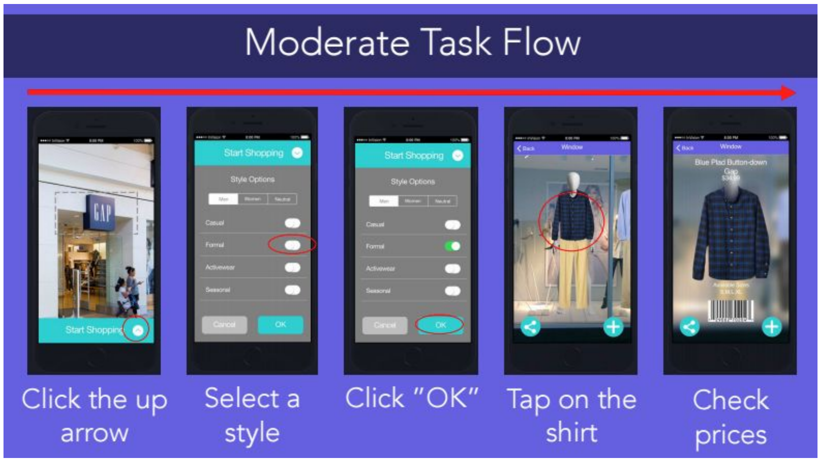
With this valuable information regarding our initial design, we moved on to develop our medium fidelity prototype.

Medium Fidelity Prototype

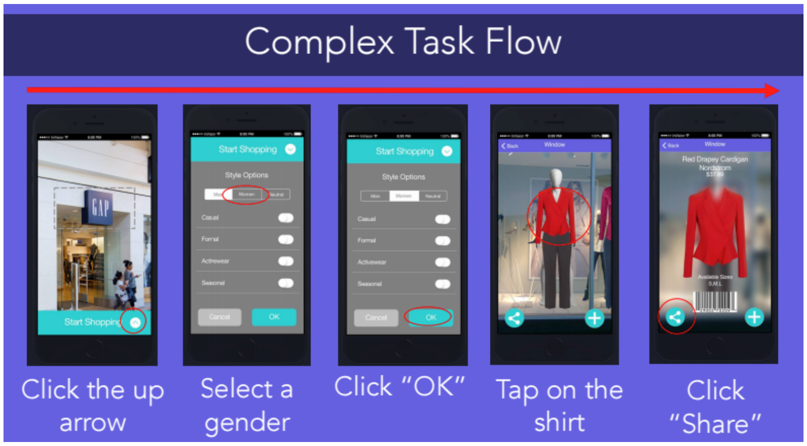
We used Sketch, a graphic design application that provided us with the tools to paste together different elements for our application, and InVision, a mobile prototyping application which allowed us to flow between different screens of our application based on different input, in order to develop our medium fidelity prototype. We used the results from user testing to make design changes to our prototype such as: a revised closet feature which is represented by a simple plus button, quicker access to the AR shopping features by jumping directly to the AR screen when the user clicks “Start Shopping,” and a slide up menu from the “Start Shopping” button which allows the user to access the filters and their custom measurements. We also added the ability to customize your measurements for the mannequin as requested by some users. Below are pictures of the task flows for our medium fidelity prototype.


Simple Task Flow

Login → Choose a store → Start shopping! → Tap on the shirt → Explore details


Moderate Task Flow

Click the up arrow → Select a style → Click "OK" → Tap on the shirt → Check prices


Complex Task Flow

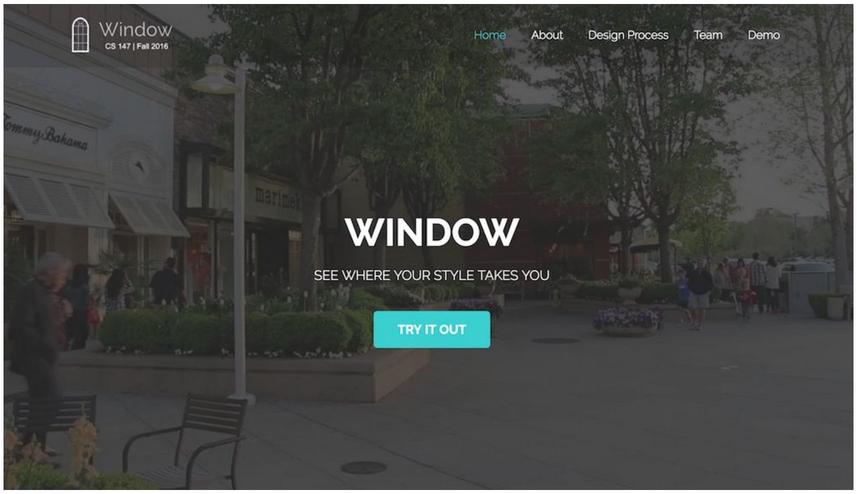
Click the up arrow → Select a gender → Click "OK" → Tap on the shirt → Click "Share"

We put our medium fidelity prototype on our website for other colleagues in our class to perform heuristic evaluations.

Website

In the midst of prototype development, we built a website to show the schematic of our progress through the iterative design process. We used a Bootstrap template for the preliminary website aesthetic, then we modified it using HTML, CSS, and Javascript. We wanted it to have a simple, sleek look to emulate the "sleekness" of our own application.

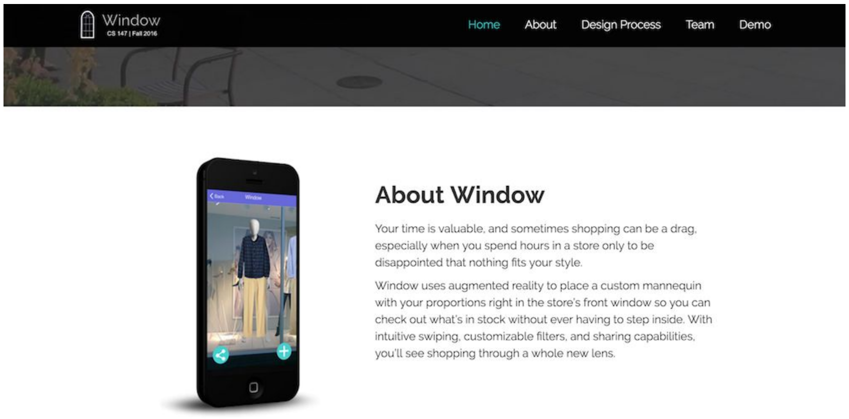
Landing Page



WINDOW

SEE WHERE YOUR STYLE TAKES YOU

About Page

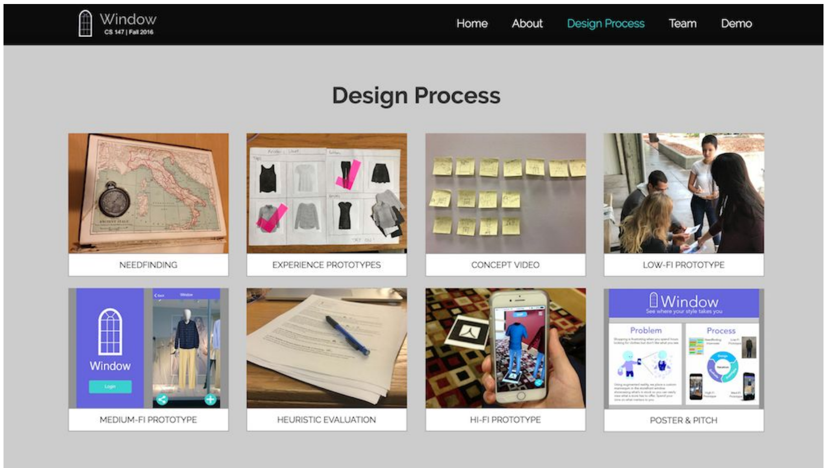


About Window

Your time is valuable, and sometimes shopping can be a drag, especially when you spend hours in a store only to be disappointed that nothing fits your style.

Window uses augmented reality to place a custom mannequin with your proportions right in the store's front window so you can check out what's in stock without ever having to step inside. With intuitive swiping, customizable filters, and sharing capabilities, you'll see shopping through a whole new lens.

Summary of our Design Process



Design Process

NEEDFINDING

EXPERIENCE PROTOTYPES

CONCEPT VIDEO

LOW-FI PROTOTYPE

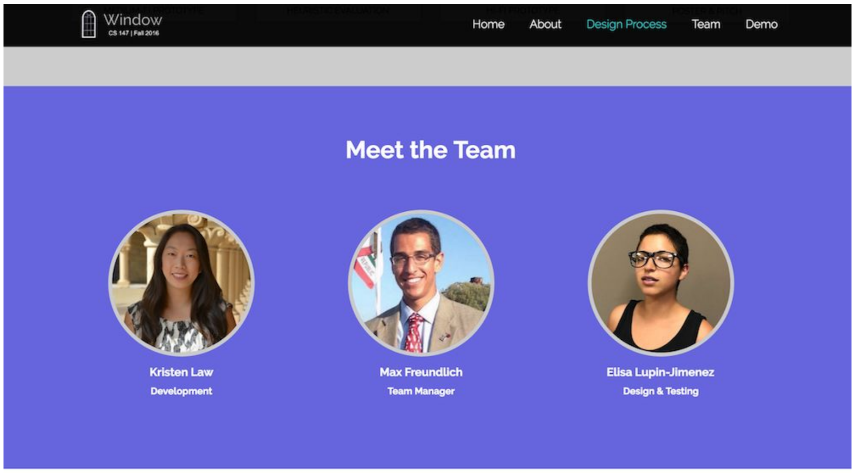
MEDIUM-FI PROTOTYPE

HEURISTIC EVALUATION

HI-FI PROTOTYPE

POSTER & PITCH

The Team



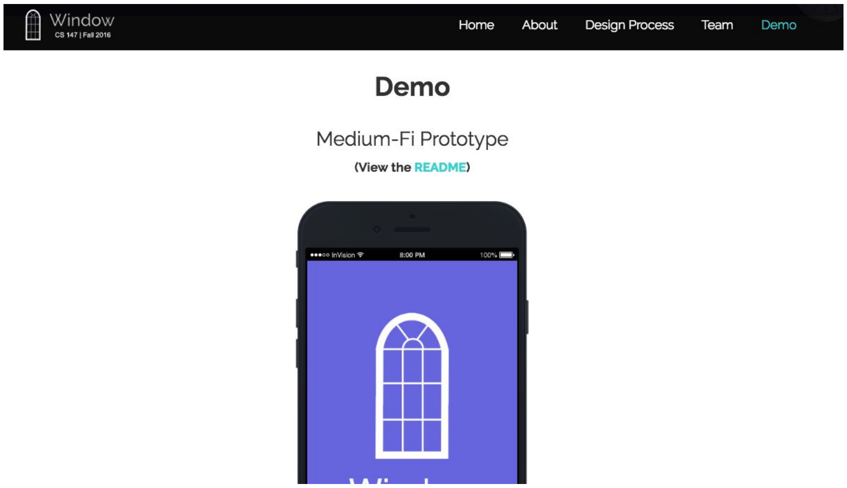
Meet the Team

Kristen Law
Development

Max Freundlich
Team Manager

Elisa Lupin-Jimenez
Design & Testing

Our In-Web Prototype

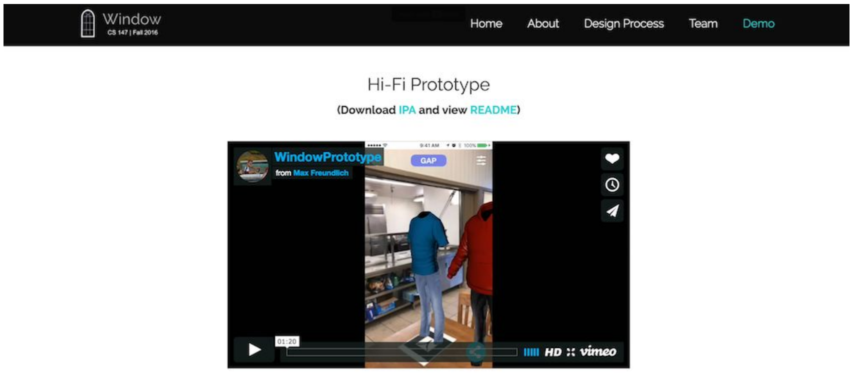


Demo

Medium-Fi Prototype

(View the README)

Hi-Fi Video



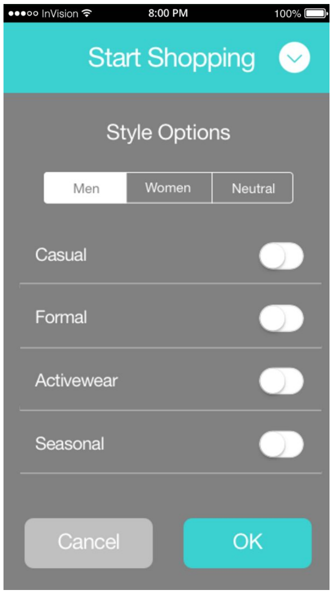
Hi-Fi Prototype

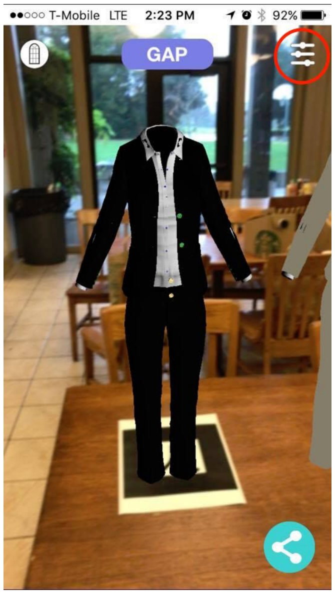
(Download IPA and view README)

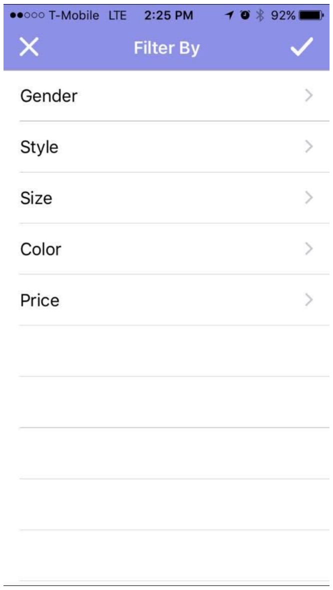
Major Usability Problems Addressed

Heuristic Evaluations

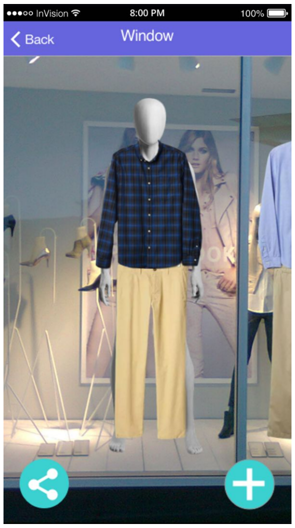
- [H2-1: Visibility of System Status] [Severity 3.5]
  - At this point in the design process there were not too many items to flip through, but when the user looks through a real store, there could potentially be many more items. Without any clear indication of how many items the user has already viewed, how many items there are total, or any ability to quickly scan through and choose what items to look at more closely when the user finds them interesting, the user could easily get frustrated.
  - Fix: We decided that we didn't want to display all of the items; rather, it is (in theory) a selection of items that the store chooses to display in this application.
- [H2-2: Match Sys & World] [Severity 4]
  - The user is unclear what the closet would be used for in the context of shopping or what it means. Without prior knowledge the user might assume that it is a list of items the user already owns or believe that the application enables machine learning to make recommendations for clothes. Users are unclear if the closet would add clothes that the user bought, will buy, or are considering buying to their saved items, since in each case, the mental model could either be a shopping cart, a fitting room, or a closet depending on the intention for saving the clothing article.
  - Fix: We realized the closet feature wasn't very clear and wasn't necessary to complete the tasks we had designed, so we decided temporarily remove it. It would be nice in the future to have the ability to use the closet as a way to bookmark clothes from certain stores, but we would need to design it in a way that makes the intent more clear.
- [H2-3: User Control & Freedom] [Severity 3]
  - After the user enters their preferences, there is no way to quickly tweak or edit them once they are looking at clothes from the menu. The user would have to go back to the "start shopping" screen, which feels like completely starting over.
  - Fix: We fixed this by adding the filtering options to the main shopping screen rather than setting them only once in the beginning.

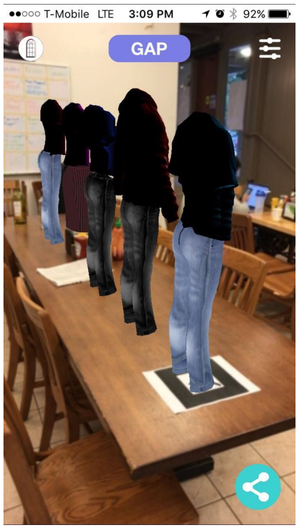




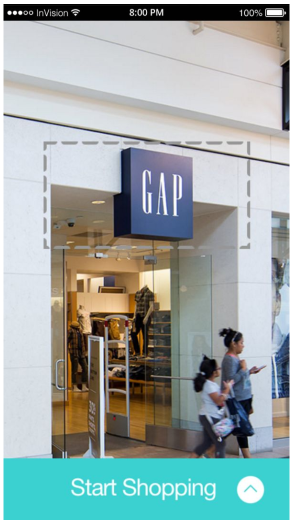


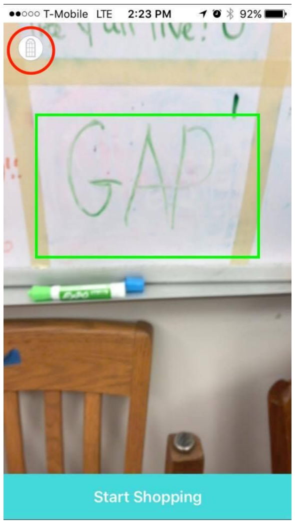
- [H2-3: User Control & Freedom] [Severity 3]
  - Users are confused by what the plus button does. They wondered if there is a quick way to star or favorite an item and come back to it. If the item is added to the closet, user didn't receive any signal that the item had been added.
  - Fix: As explained earlier, we decided to remove the closet feature for now to avoid any potential confusion.
- [H2-3: User Control & Freedom] [Severity 3.5]
  - If users were interested in what the back of the clothes look like, or if there are multiple color options, they had no ability to see these different options.
  - Fix: The clothes are displayed as 3d models rendered above the marker, so users can move around the marker to see different angles. We agree that it would be useful to be able to see if a clothing item has other color options and hope to add that feature in the future.

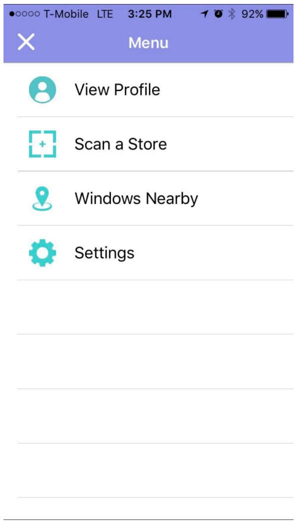




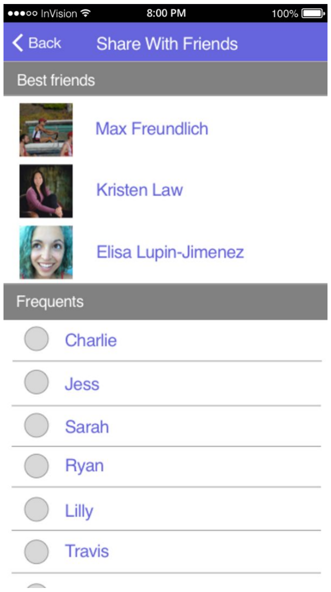
- [H2-3: Use Control & Freedom] [Severity 3]
  - When choosing "scan a store" from the menu there is no way to go back to where the user was in case they change their mind.
  - Fix: We added a menu button in the top left of the "scan a store" screen.

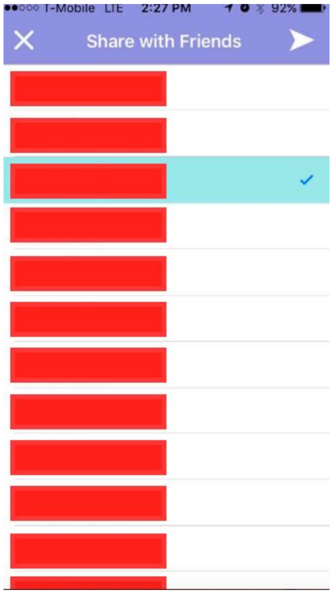




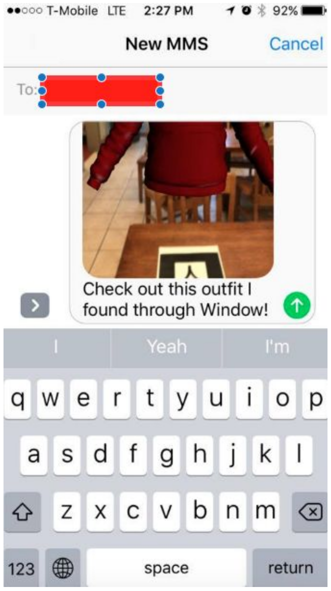


- [H2-3 User Control & Freedom] [Severity 3]
  - The app doesn't have the option to remove items from the virtual closet.
  - Fix: As explained earlier, we decided to remove the closet feature for now.
- [H2-3: User Control & Freedom] [Severity 3]
  - The user can send pictures of items to friends, but were unclear how it is delivered. The user wondered if messages happened in the application, and if they had their own inbox; users also wondered how they can see items that they have previously sent to friends.
  - Fix: We screenshot the user's current screen of the outfit or item they are looking at and allow the user to send the image to someone in the contacts via SMS.





[redacted]

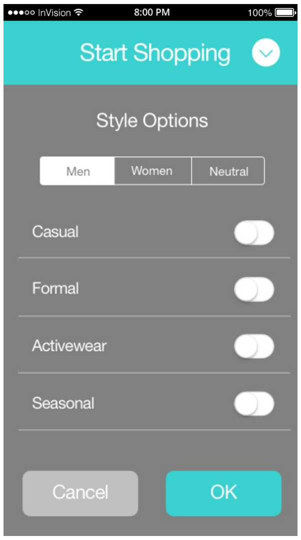


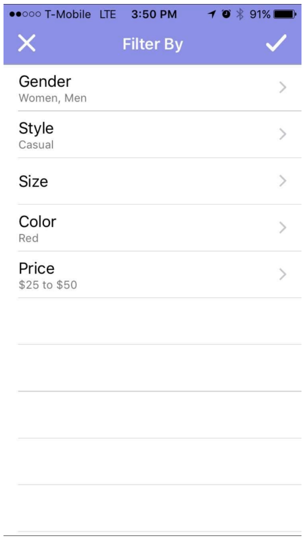
[redacted]

Check out this outfit I found through Window!

- [H2-3: User Control & Freedom] [Severity 3]
  - If the user want to send something to someone who doesn't have the app, they were unsure how to send it through a third party service.
  - Fix: As mentioned earlier, we allow users to share via SMS.
- [H2-4: Consistency & Standards] [Severity 3]
  - Users were unclear if there is a way to buy online in case the user doesn't want to buy the item in store for some reason.
  - Fix: This is a fix that we will add in the future if stores were to adopt our product. There would be a clear way to purchase the item from within the app and have it shipped to their home. However, for the purposes of this prototype if a user liked the item they were looking at they could go into the store and by it there. The app shows only items that are available in the store.
- [H2-4 Consistency and Standards] [Severity 3]
  - In the filtering menu, the user was confused that when all filters are off it means that the application will 'show everything'. Users wondered if filtering used AND or OR logic operators. For example, if the user has 'Active' and 'Formal' selected, they were unsure if they will see activewear and formalwear, or if they will see just clothing that fits both categories.
  - Fix: With our new workflow, the user does not specify any filters before entering the AR view. The 3D models that are initially shown are defaults which makes it clear that no filtering options means show default models. Additionally, selecting filtering options in the

filter view now clearly displays models under all categories selected. For example, “Active” and “Formal” selections will display active and formal clothing.





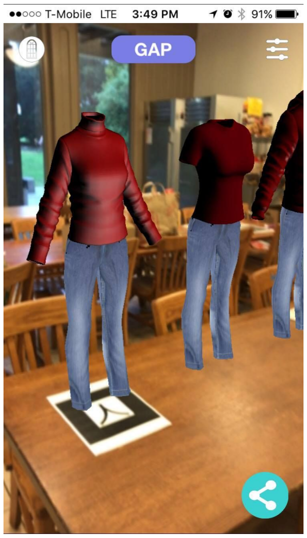
Gender
Women, Men

Style
Casual

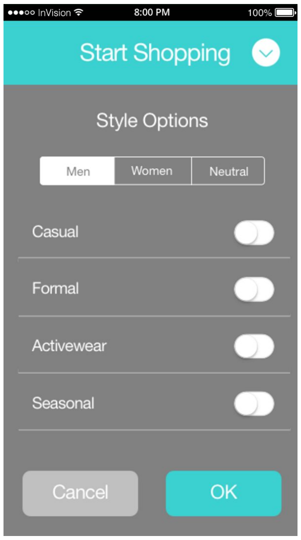
Size

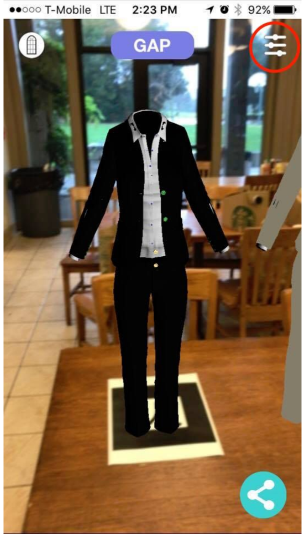
Color
Red

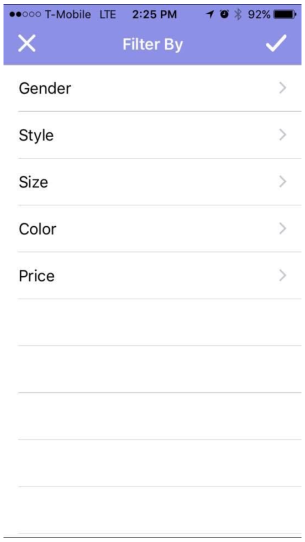
Price
$25 to $50



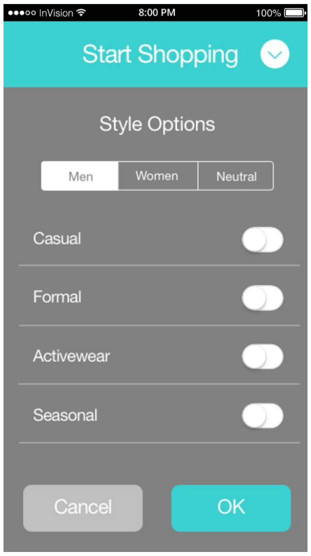
- [H2-5. Error prevention] [Severity 3]
  - It’s not entirely clear that the up arrow and “start shopping” buttons are separate items in the main screen. Users could easily make the error of entering one when they meant the other. It’s also unclear that there’s a way to just “start shopping” without setting user preferences every time. It is convenient to set user preferences once and have them stored, but it is not entirely obvious from the visual design.
  - Fix: We reworked this workflow such that the user scans a store and immediately beings shopping. Filtering options are then specified by the user in the AR view. The initial options are default options that would be specified by the store. All user preferences are saved to the phone if specified.

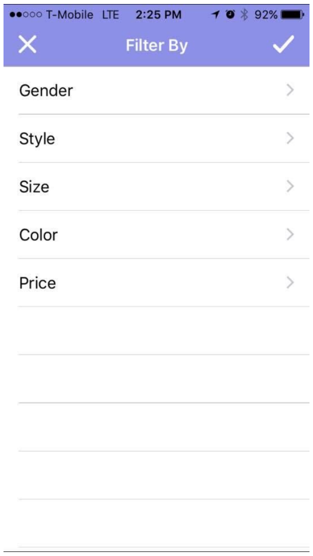


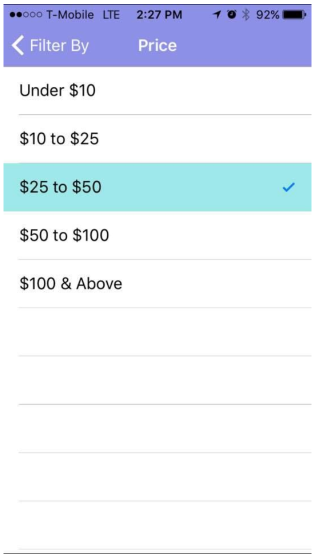




- [H2-7 Flexibility and Efficiency of Use] [Severity 3]
    - Users were unsure how to set a budget, such as for the moderate task.
    - Fix: We added a price range to our filtering options. This allows users to quickly set a range of prices that they are willing to pay for clothing.







- [H2-7 Flexibility and Efficiency of Use] [Severity 3]
    - Users were unclear what the distinction was between best friends and frequents. Also, they were unsure how to view their normal friends.
    - Fix: For simplicity, we completely eliminated friends from the app. Now all sharing is done with SMS. We take a screenshot of the article of clothing and format a text to send to friends.
- [H2-7 Flexibility and Efficiency of Use] [Severity 3]
    - Users were unsure how to add friends.
    - Fix: For simplicity, we completely eliminated friends from the app.

Other Changes

We decided to not put a mannequin in the AR viewport since we didn't implement the custom measurements aspect of the app. Ideally this mannequin would be custom sized to the dimensions of the user however we could not implement this in the time frame, or graphics experience, we had. Instead we decided to display models of the individual clothing items to represent different outfits.

We entirely removed logging in to the app. We decided that we could store all user information locally to the device upon the first time opening the application. This makes the user experience much more seamless and avoids any confusion that comes with creating an account. Additionally, it helps get the user into the core aspects of the app faster which, ideally, will help in retaining a user

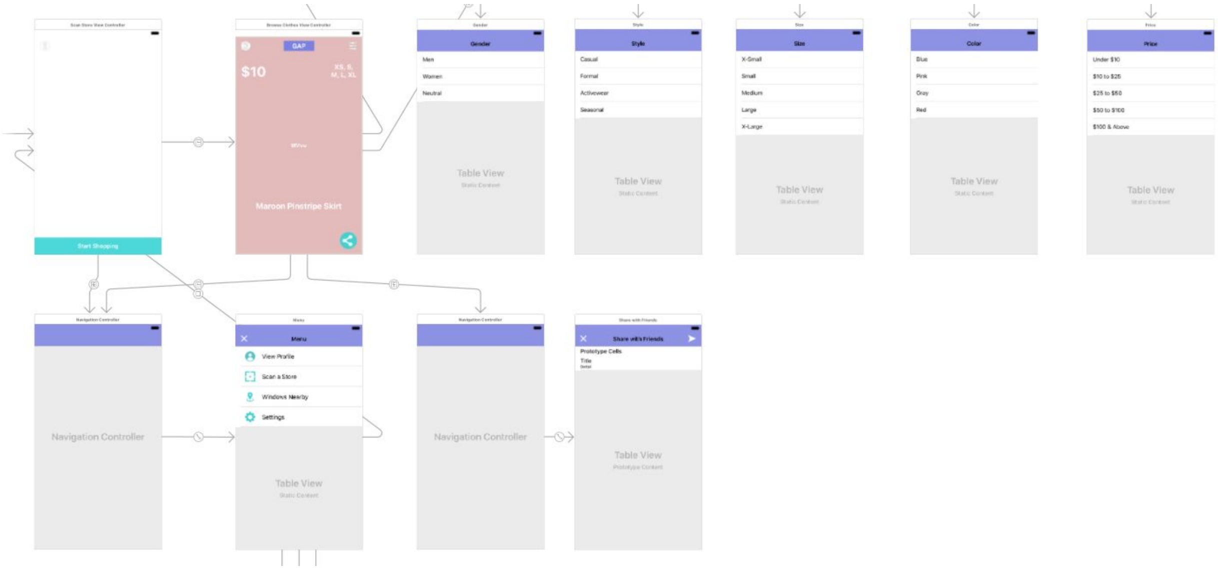
In the medium fidelity prototype had a barcode on the detail screen that the user could bring into the store and have scanned to pull up an item. This confused many of the people we showed the demo to so we decided to eliminate it entirely. Instead, if the user finds an item they like they can do into the store knowing that it is in stock and fill fit their body type. Additionally, we want to implement a feature that allows the user to purchase the item online.

Prototype Implementation

Tools

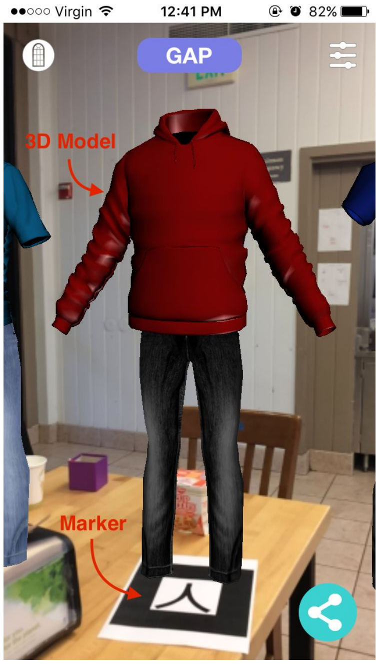
We developed this prototype in the XCode IDE entirely natively using Objective-C. Both Kristen and Max have had experience building mobile applications in Swift and were familiar with the XCode development environment which made this tool easy and quick to use. However, both developers had to write the app in Objective-C due to constraints by the ARToolkit SDK, which neither had any experience in. Luckily, there is plenty of crossover between Swift and Objective-C so they were both able to pick the language up easily.

XCode allowed Kristen to create the user interface of the application via its storyboard tool. Here, Kristen was able to declare views and specify their functionality in view controller files written in Objective-C.



The storyboard interface is an incredibly useful tool for maintaining a clear representation of every state of the application. It was also really useful for designing the app interface and interactions and being able to view all the screens at once.

Additionally, we used the ARToolKit SDK for iOS; a free open source SDK designed for building augmented reality applications. After downloading the source code from Github as well as multiple examples and we were able to read through the code and get an understanding of how everything was implemented. ARToolKit is essentially an Objective-C wrapper for OpenGL that has two major functions. First, it uses blob detection to recognize pre-specified markers. This allows the app to continuously keep track of a marker, which is just a print out of any image we trained the app to recognize, in real-time. Second, it continuously renders 3D objects onto the location of the tracked marker. It does this by first storing the 3D objects as well as their textures and then uses the x,y, and z coordinates of the tracked marker to determine where and how to orient the 3d model.



After having a thorough understanding how the ARToolKit SDK worked, we were able to go into its source code and alter the functions relevant to our needs. This proved very useful as we were able to load in our own 3D models, and easily position them relative to each other in the viewport. Additionally, the SDK supported various scaling tools which we were able to use to properly proportion our models. The toolkit created its own OpenGL “view” and with that it provided a method for detecting clicks on screen which we took advantage of.

Unfortunately we were somewhat limited in detecting where specifically the user touched on a model. The methods provided by ARToolKit couldn’t compute the proper matrix multiplication that was necessary to transform a tap on the screen to a traced ray in the 3d scene for detecting a click on a 3d model. Thus, we had to use a Wizard of Oz technique where the top of the screen meant the user wanted details from a top piece and the bottom of the screen meant the user wanted details from bottom piece.

Another slight letdown of this toolkit was the issues it had with loading in complex models. We found that models that had more triangles and more complex features were unable to load properly in the viewport. Using OpenGL to render the objects on a Mac proved to work fine, and thus we concluded that it was the ARToolKit’s OpenGL wrappers that were failing.

Wizard of Oz

The “Scan a store” view doesn’t actually use Optical Character Recognition (OCR) to detect the store name. Instead, we simply present a red box which is the view in which we fake OCR. Click on this red box will “scan” the store name and the turn the box to green indicating a successful scan. The user can then move to viewing 3D models. If we were to implement this feature we would use a framework like Tesseract to perform OCR.

Prices are randomly generated depending on the pricing filters. For example, if the price range is 0-100 dollars we will randomly generate a price between 0 and 100 dollars and associate it with a model in the dictionary.

Users cannot click directly on an individual item of clothing to bring ups its detail view. Instead, the need to click on the top of the screen for the top piece of clothing and the bottom of the screen for a bottom piece.

Hard-Coded Data

The user's profile is entirely hard coded information. When a user signs up for the first time they would be prompted to fill in a photo of themselves, their name, as well as their dimensions. For the purposes of this prototype we prefilled all of this information.

On the view where the user can search for stores that have window technology, we hard coded the map to be of stanford shopping center as well as the locations of each pin.

The filtering options are all hard coded. Filtering is dependent on the store the user is at but since in this prototype we aren't actually hooked up to a store the filtering is predefined.

Every time an AR view is loaded it pulls the objects from a .DAT file. This file contains a list of objects and their dimensions. When filtering, we hard coded which filter combinations went to which .DAT file. This means that we hard coded multiple .DAT files as well as multiple filtering combinations. This was the alternative of dynamically generating .DAT files depending on the set of filters chosen by the user. We made sure that we could handle the filtering options for the tasks as well as some additional filters.

Future Modifications

We wish that we could have implemented the custom measurements for clothing (so that users can see what an article of clothing would look like on their body type), but due to time constraints we were unable to add this feature into the high fidelity prototype. We acknowledge that this would be a major plus to our application, so enabling this functionality would be our top priority in the future.

Another feature that would make our application more marketable is to actually enable store scanning features. Because we prioritized displaying our application's core functionality, we put store recognition on hold. If we were to implement this feature, we would have to work with the stores to receive their scanned collection of clothes and use some sort of recognition technology to "read" the names of the stores from photos.

One more major feature that we would like to add is our original "virtual closet" idea. We had to table the idea in order to focus on more of the core functionality, but we would have liked to hash out the details of what the closet is as well as match our mental model of what the closet is supposed to be used for to the mental model that most users will have. We would hope that the closest feature could be used for favoriting items and perhaps returning to them from other store locations.

Summary

Over the past ten weeks, we had our ups and downs. There were many nights where we stayed up late to get the perfect aesthetic for a design, and there were many times that we banged our heads against a wall trying to figure out what our next step was. Now that we have reached the final stages of this class, though, we feel a great appreciation for the assistance that we received from our CA John as well as the material and resources that were provided to us throughout the course. Mostly, we find it hard to believe that we, a small team of three people, were able to produce a working AR application that many critics have expressed a desire to use in the real world. Not only have we learned valuable design strategies that we can employ in our future endeavors, but we also formed new friendships between each other and learned how to balance teamwork with fun. Even though this may be the end of CS 147, it will not be the end of our growing knowledge base for human-computer interactions.

Thank you!