PROBLEM AND SOLUTION OVERVIEW

Sometimes, people will see an image, a pattern, or a style that they really like and will want to own something in that same style. Unfortunately, there is currently no easy way for people to find items with a specific style that they are looking for. Our proposed solution to this problem is a mobile searching application that utilizes crowd sourcing to find its results. The application will allow users to take pictures of styles that they like and will use a crowd-sourced AI such as Mechanical Turk to allow other people to try and find items that are identical or similar in style to requests that users submit.

TASKS

[Simple Task] Search for an item you see in your environment and want to find

Imagine you are walking on University Way and girl wearing a leather purse that catches your eye. You wish you had one just like it. StyleEye allows you to quickly take a photo of this purse and find similar styles. Just press the camera button, take a photo, crop it, and submit it along with an optional text description of what you are looking for. Within minutes, you will receive suggestions for items that resemble the purse.

[Moderate Task] Stay up-to-date with the style of a person whose style you admire

You follow Jessica Simpson on StyleEye because you are interested in her style. StyleEye gives you the ability to stay up-to-date on the actions of users you are following. Simply press the “Popular” button, which is visible all the time, then press the “Following” tab to display the profiles of people you are following.

[Difficult Task] Browse the latest popular styles to find a style that you would like

As a curious StyleEye user, you are interested in what the community considers to be the best styles, and you would like to find a style that you like amongst the most popular so that your style doesn't become outdated. Press the “Popular” button. This brings you to a screen that displays a feed of styles that are on the rise.
REVISED INTERFACE DESIGN

After analyzing feedback from user testing, we made a few key changes to StyleEye’s interface and accounted for a few scenarios that users will run into.

[Added text to the three main buttons on the bottom] The feedback we received from user testing indicated that the functionality of each button on the bottom wasn’t obvious. To help with this, we changed the button icons (except the camera) and added text descriptions to the buttons to help the user immediately recognize what each button is supposed to do, even if the user is using StyleEye for the first time. The landing page for StyleEye, below, shows that the three buttons on the bottom are enhanced with a text description. See figure 1a

[A picture is now required with each request] Given the difficulty found by our testers while trying the former moderate task (find a clothing item without using a picture), it was decided that there would be less room for user error by requiring a picture to be attached with every request. It seems more intuitive to implement one easy way to submit a request instead of implementing several different ways to do so. This way, there are far less mistakes that the user is able to make. After a picture has been taken, and cropped appropriately, the next step is submitting the picture along with an optional text description (interface shown below). See figure 1b

[Got rid of “style preferences”] The purpose of “style preferences” was to help the Mechanical Turk workers show items to our users that were more likely to be of interest. For example, if a user indicated in their “style preferences” that they only like Nike hoodies, then every time they submitted a request for a hoody, the results would likely be only Nike hoodies. This is a problem because there is very likely a scenario in which this user may see a non-Nike hoody that he wants and requests. It is simpler for the user to add restrictions to what they are looking for right before they submit the request. The text box in screen, above, is where the user can state specifically what they are looking for.
We wanted to better organize how our users could see results from their past requests. We added a new section to StyleEye called “Popular”, which can be accessed at anytime via the bottom left button right “popular”. Within “Popular”, users can see what the greater StyleEye community is reposting most in the “Trending” section or what the people they follow are posting in the “Following Section.” By clicking on a style to see its details, the user can “repost” styles posted by other people as their own. We wanted to better organize how our users could see results posted by other users. See figure 1c.
PROTOTYPE OVERVIEW

We created the interactive prototype for StyleEye using a combination of different tools, mainly Adobe Photoshop and JustInMind. Using Photoshop, we were able to create the screens of our mobile application. By using layers, we were able to reuse a lot of our work by making simple modifications to the screens we had already made. This expedited the process of creating the numerous screens.

One alternative that we considered was creating the screens using JustInMind. This program provides a wide range of phone widgets and resources that would give our application the exact iPhone body, popover menus, buttons, navigation bars, and notification messages as seen on the real device. However, due to time constraint, we did not have very much time to explore the program and experiment with the available functionality. JustInMind could have been a lot more useful in creating our prototype if we had more time to learn the potential of JustInMind and create storyboards with a better idea of what is possible to create using the program. Since we had more experience and knowledge of how to use Photoshop, we decided to do most of our screen creation using Photoshop.
Though Photoshop was very helpful in creating most of the screens, it only allowed us to use hard-code data. Using Photoshop exclusively would limit our application in allowing user interaction, user input, and a deeper interaction with the prototype. However, JustInMind provided us the tools we needed to do this. Using JustInMind, we were not only able to link the screens and implement the functions of the application, but we were also able to save user-submitted input submitted on one screen and display the information later on another. This made our application more dynamic and feel more like the actual product.

Photoshop and JustInMind were the main tools we used to create our prototype. No wizard of oz techniques were necessarily to implement our features.

[Overview of UI] When first launching StyleEye, the user lands on the trending page under the popular section(p1). This allows the user to see what styles are being re-posted the most by all StyleEye users. If the user selects the “followers” button on the popular page, they see a feed of the most reposed styles limited to those people who they follow. The center navigation button is the camera (p4). We wanted this to be prominent but not the landing page because we think people will view pre-posted styles more frequently then posting new ones. Adding a new style is an easy process where the user takes a photo, crops it (1.1), and adds any addition information that pertains to the matches they are seeking (1.2). The style they added can then be seen on the user’s “my style” page (1.3). The far right button on the primary navigation takes the user to “my style” (p3) which is a page that displays photos the user took and re-posted. In the “my style” section, the user can view a list of the people who follow them in addition to the people they follow (2.1). If a user clicks on a style post anywhere on the site they are taken to the style’s detail page (3.1), which shows the original post plus any matches recommended by Mechanical Turk.

[Limitations / Hard-coded Features] The interactive prototype is limited in scope and only has the main functionality implemented. Some secondary functions such as going to the site of an item or browsing your phone’s photo library for a picture have not been implemented because they are not necessary in order to accomplish our main tasks. In some cases where a user interaction with the prototype may be redundant, only one of the interactions is implemented. For example, there are many style items posted on the “My Style” news feed; however, interaction with only one of the style items is supported in our prototype. The other style items would have the same functionality and would require more screens to be made, so these items were not included in the prototype.

Some parts of our prototype are hard-coded to give our application the illusion that there are other users already using StyleEye. For example, the “Popular” news feed contains items submitted by other StyleEye users. Thus, these parts of the prototype will not change.

We also did not account for saved states in our prototype due to time constraints. For example, after completing Task 1, if the user navigates away and back to the “My Style” page, the submission done in Task 1 will no longer be present on the page. We felt that this aspect was not needed in our prototype because it was not necessary in order to demonstrate how our main tasks worked.