Annie Lin
Tim Mekari
Johnathan Duong

Mantis: A Mobile Application for Color Identification

Abstract

Mantis is a mobile application that identifies colors based off a user-given image. The user would be able to take a picture or upload a picture within the mobile application and select a specific spot by tapping on the picture. That selection would return a specific color name and three color codes: hex, RGB, and CMYK. In addition, the user would be able to generate a color palette which produces five additional colors that complement the initial chosen color. If the user likes a certain color or palette, he or she would be able to save it in a personal color library which keeps track of saved colors and palettes chronologically over time. Within each saved color or palette listing, the user would be able to click on each to see information about the color names and color codes.

Introduction

Have you ever stopped to ask yourself what color something really was? Have you ever seen a color and wondered exactly what that shade was?

From a general standpoint, anyone could be curious about a color because it is appealing. Suppose you were hiking on a Yosemite trail and you encountered a dainty patch of flowers that was the utmost shade of light purple, but you could not really think of a color that best fit the shade. It was not quite a purple shade because it had some pink tones, but it was not quite a lavender shade either. You just had to know what the name of the color was in case you would use it sometime in the near future possibly for a wedding theme or for painting a room, but there was no way to find out. This also applies to a designer's standpoint. One of the most important aspects for design projects is to choose a color scheme for the projects. Because colors can carry emotional resonance, the right colors will set a mood for the design and elicit certain emotions from the audience or consumer. Thus, if the designer does not choose a color that matches well with the design, it is likely that the design will be unsuccessful. If a designer sees a color he or she likes throughout the day, he or she should be able to identify the color and use it for future designs.

From these reasons, we decided to implement a mobile application because of its practicality. A web application is impractical because it does not make sense for a user to take a picture through

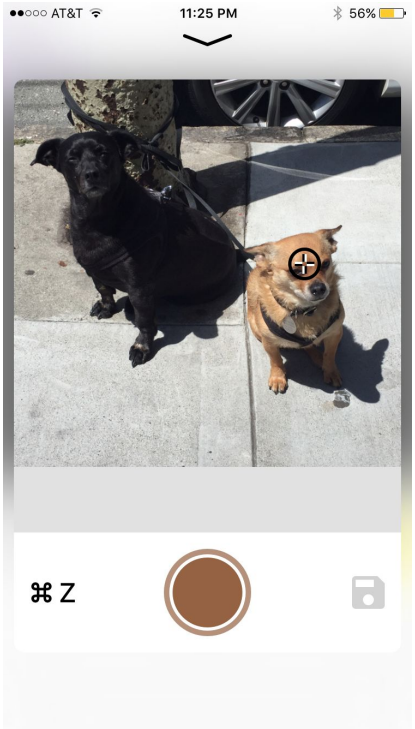
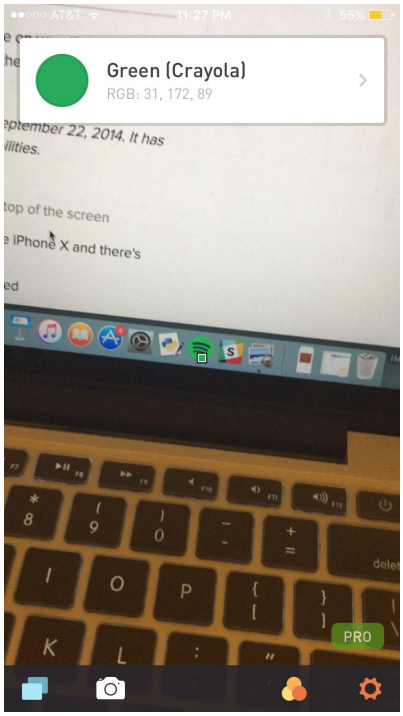
their phone. The purpose of a mobile application makes the action of taking a picture more efficient. If someone sees a color right away, chances are they would rather take a picture on their phone, pull out their computer to upload and identify a color. With Mantis, users will able to quickly whip out their phone and snap a picture.

Then we decided to make sure our color identification provides users with exact information like color codes, so they would be able to use chosen colors in their projects or applications. A color scheme generation is an added bonus we decided to implement because we wanted users to feel inspired and have our app add to ideation for their projects. Then of course we wanted to implement a way to save favorite colors and palette because it would help users keep track them for future uses. People are incredible busy nowadays and have so many projects and apps to look at every day. It becomes more difficult for people to keep track of every single detail, so our app's save feature would prevent that.

Background

Surveying the app store for related work, we found apps that had the general idea of what we were trying to but executed poorly. We knew going into this project that we were not trying to reinvent the wheel, but instead trying to improve it. We believed that Mantis could become a staple on every phone, while the apps on the app store saw it as more of a novelty.

The pictures below come from two other color identification apps on the app store. The one on the left provides minimal information for the user because it only displays the selected color which does not give the user any other information whatsoever. The app on the right employs AR to identify live colors, which is innovative but annoying because the color identification is always moving since a user cannot pinpoint a single color due to minute movements from the hand.

Palette	Color Name AR
	

Also the apps on the app store did not allow for the full range of color. If a user pointed it at any shade of green it would just identify a basic color like “Green” or “Red” instead of something more specific like “Lime” or “Scarlet”. Along with that, the UI only worsened the usability of those apps, since navigation was not intuitive. We had no idea how to get back to using the camera for Color Name AR. Mantis improved on both of those parts. With the API that we used to identify color, it retrieved the hex, RGB, and CMYK codes for the color that was chosen and the color palette generation; this allowed us to display the exact color that was chosen on your device. Also, our planned UI was very intuitive and flexible. A first-time user could easily navigate the app and understand how to use it. Our Color Library implementation also helped users keep track of their favorite colors and palettes so they would be able to use it in the future.

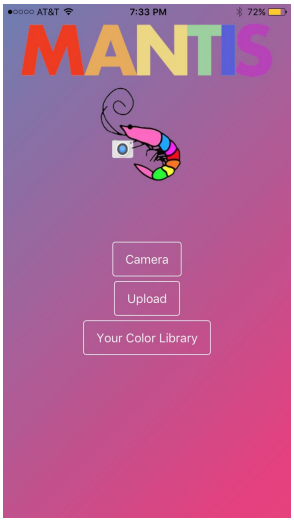
Conceptual Model

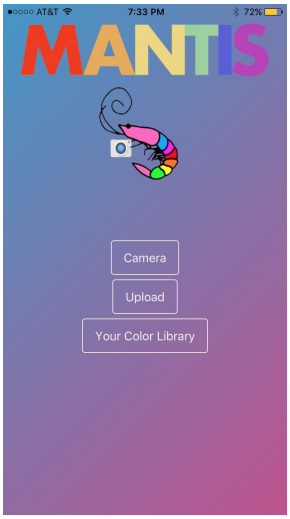
The conceptual design of our application was extremely simplistic and consisted of three main components which were the photo camera, upload, and Color Library. By combining these three pieces, we created a unique user experience that solved the problem of being able to analyze, identify, and save colors into palettes. The application allows for user input from the user via the camera or upload option and from there transitions to an intermediary color selector page. There a picker can individually select a color or an entire palette can be created from the dominant

colors of the image allowing for a variety of options whereas most other applications do not fully implement user choice. Lastly, the user selection can be saved and accessed within the color library allowing for as many color palettes as the user would like to stored and re-accessible.

Prototype

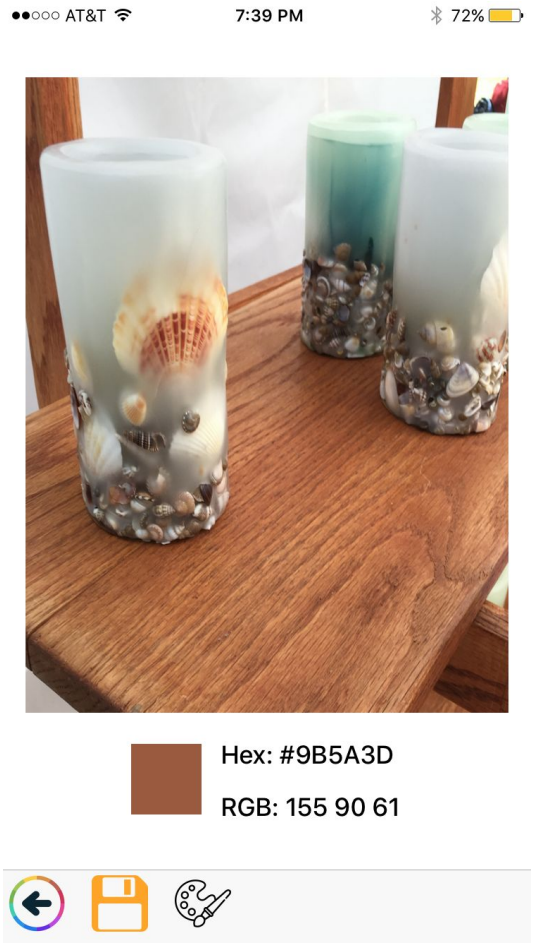
To create the Mantis app, we decided to use the Ionic Framework which is an open-source SDK for hybrid mobile app development, built on AngularJS and Apache Cordova. We had a total of five main pages within our app. The first page consisted of the home page which gave the user an three options: taking a picture, uploading a picture, and navigating to the Color Library.



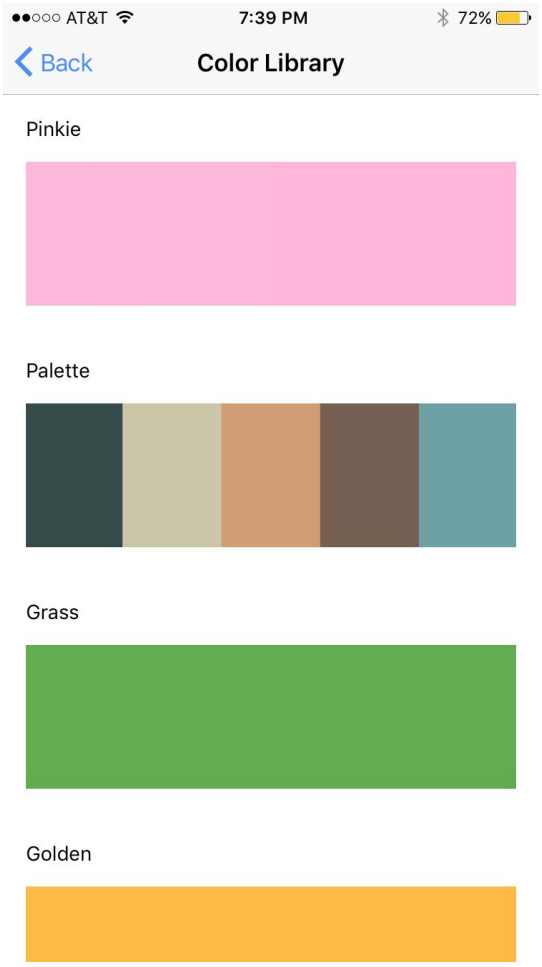


We wanted our whole app to be colorful because it was an app that related to the notion of colors, so we chose Mantis as our name because it is based on Mantis Shrimp which has twelve color-receptive cones whereas humans have only three. This means that Mantis Shrimp can see colors our brains are not even capable of processing. Thus, the home page's title is in rainbow and the background is a moving gradient as seen above.

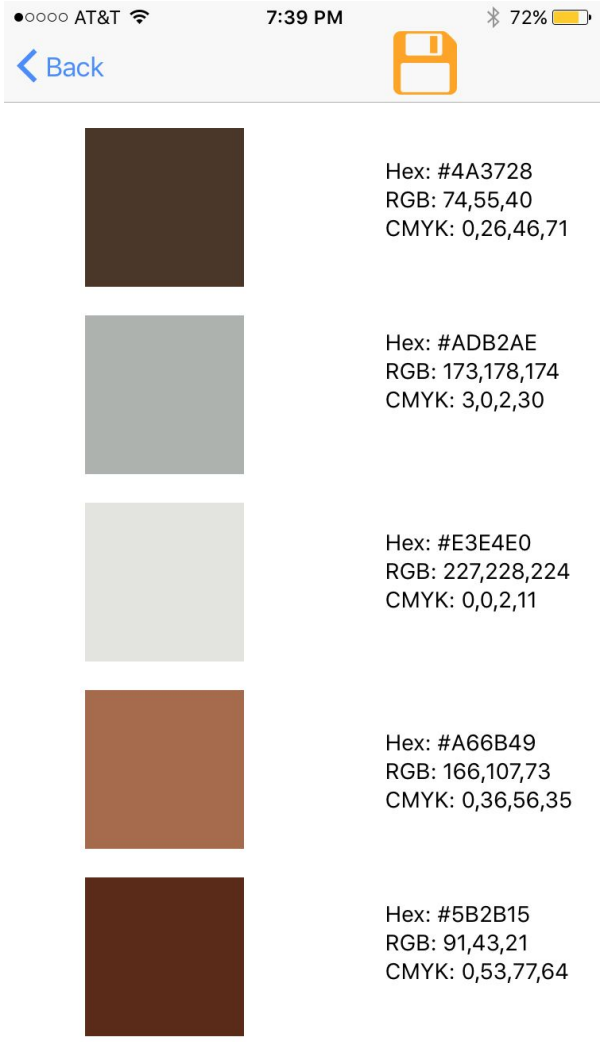
Our next page displays the picture that the user chooses, and wherever the user touches on the picture, the color, hex code, and RGB codes will displayed below. This is done with an API built off of AngularJS that specifically spits out the exact color value. This page also includes navigation in the bottom bar which allows the user to navigate to the previous page, save the identified color, and generate a color palette that compliments the identified color. The color palette generator is implemented with the ColorMind REST API, which uses deep learning to learn color styles from photographs, movies, and popular art in order to create aesthetically pleasing color schemes.



The next main page is the Color Library page. This page displays all the colors and palettes that the user has saved, chronologically. We figured that the user would like to access the most recent save so we made sure the page displayed most recent to least recent. We implemented this by using Ionic's built in Storage plugin, which allows storage of key value pairs, so we can keep track of color or palette information.



Then, within each color or palette listed in the Color Library, the user would be able to click on each listing to see more information. For example, the picture below shows more information of a color, which will have hex, RGB, and CMYK codes. A palette page would display each of the colors in the palette with their respective color codes as well.

Color Information	Palette Information
	

Discussion

As you could imagine, trying to implement something like this was not easy and we ended up running into a lot of issues. First off, our camera could not actually upload a picture that we took right there. The problem we found was that the app would push to the next page immediately

after pressing the camera button, so it would not push the picture we just took until the next time that we hit the camera button. Along with that, our eyedropper API would not implement as cohesively as we wanted it to. The eyedropper API would allow users to zoom in and really select the color they wanted, so our app did not have the accuracy that we wanted while selecting colors. Making Mantis, we learned that mobile apps are hard to make. You cannot just take APIs and fit them like puzzle pieces to complete something. There is a lot of steps that have to be taken into consideration before you actually have something functioning. Along with that, we learned that a lot of thought goes into UI. Taking this class we learned a lot more about actual design, so as we went on our ideas on UI were advancing. This ended up leading to a fair amount of time being spent on making sure that our app followed all the standards.

Future Work

With an app like this, the possibilities are limitless for an extension. If we were to continue working on this project, we would want to clean it up and add more technical features. We want the user to be able to zoom, crop, and manipulate their pictures if they want the API to create color palette for just one portion of the screen. Along with that we would also like to add the ability to make accounts. Your account would allow you to save your data and access it on all devices. We also had the idea of using these accounts to create a social aspect for the app. The way the social aspect would work is that you can add friends, and friends can put colors on their wish-list with a description of what they are looking for. For example, if your friend scans a color that is close to the one on your wishlist, they will get an alert in the corner that says “This color is close to the one on Tim’s wish list with description ‘I want this color paint for my room’. Send message”. This will really help with the design aspect of this app because it will help users find objects that they really want in a certain color. Also, users would be able to upload colors that they find and other users could vote on them to choose a top five colors of the week so people can share the cool colors that they find. The social aspect of this app has not been touched on at all by similar works and we feel like we can expand greatly in that area.