Color Buddy
Senior Design Project Fall 2019

Christina Bateman
Lukas Saul

Abstract

Imagine you're shopping at your favorite clothing store. You find an amazing shirt, but you don't know if it goes with anything you own. If it doesn't match anything, you will have wasted money on a shirt you'll never wear. What if there was a way to know without having to make any more wasteful purchases? Color Buddy is an Android app developed with Kotlin and Firebase which helps users curate the perfect wardrobe, interior design scheme, and more. It holds the color data for each item in the user's collection. When a user wants to ada new item to their collection, the user takes a picture of the object. Color Buddy forms a palette of the object's most prominent colors and, using color theory concepts, checks to see if a new item matches anything in the collection. Users are alerted in the new item matches or not, so they can make the most informed decisions.

1. Introduction

The goal of this project was an Android application that helps users determine if a new item would match items they already own (i.e. clothes, furniture, etc.). Our expected audience for this application is people who want help unifying their color schemes and potentially people who are color blind. We expect that people would develop a greater understanding of color matching, gain a more unified color scheme for whichever set of items they used the app for, and saved money by not buying things that don't match what they own.

1.1. Background

Color Theory:
The theory behind color matching is based on the color wheel. It typically describes four schemes by which colors are matched.
Complementary: These are colors that lie across from each other on the wheel.
Analagous: These are colors next to each other on the wheel
Triadic: These are three colors that are seperated by three other colors on the wheel. This scheme is always three primary colors, three secondary colors, or three tertiary colors.
Monochromatic: These are colors that are the same base color but are a shade(darker) or tint(lighter) of that color.
All of this information was translated to the unit circle to make it possible to do calculations over the pixel data obtained from the pictures. Initially, it was RGB (Red, Green, Blue) but this was converted to HSV (Hue, Saturation, Value) to allow for easier calculations. A color is then checked to see if its monochromatic with a color in the group, we check to see if its with 45 degrees of the color. This will be the basis for the rest of the schemes. When we look for another color that fits the scheme we add the appropriate amount for the scheme (180 degrees for Complementary, 30 or 60 degrees for Analagous, and 120 degrees for Triadic), then check if that new color is monochromatic with the one we're looking to match.

1.2. Challenges

Image size:
We cut off the outer edges of the photo (10% off each side) and took a sampling of the pixels from what was left.
Image and color quality:
We loosened up the definition of the monochromatic color scheme to account of variances in image quality, lighting, etc.
Calculation time:
A 12 MP camera has 12 Million pixels so looking at as many pixels as possible presents a potential challenge in the matching calculations.

2. Scope

The scope of this project is to develop an Android application that allows users to add items to a group and check to see if they match with one another.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Register	User	Easy	1
2	Login	User	Easy	1
3	Create Group	User	Easy	1
4	Add Items	User	Easy	1
5	Delete Items/Group	User	Easy	1

TABLE 1. USE CASES

2.1. Requirements

2.1.1. Functional.

- User needs to have private groups, these will be the items the user wants to potentially match and no one else should be able to see them.
- Users should have the ability
- You'll need more than 2 of these...

2.1.2. Non-Functional.

- Security – user credentials must be verified and stored within firebase.
- Usability – the application must be fairly simple so that most people could be expected to operate it without much trouble.

2.2. Use Cases

Use Case Number: 1
Use Case Name: Register
Description: A user has decided they want to use Color Buddy. They will click the 'Register' button. This will allow them to register.

1) User opens the app.
2) User clicks on the 'Register' button.
3) User enters the E-mail and password they would like to use.

Termination Outcome: The user is now registered with the application.

see Figure 2

Use Case Number: 2
Use Case Name: Login
Description: A user has registered with Color Buddy. They will input their credentials and click the 'Login' Button. This will take them to the main App.

1) User opens the application after registering.
2) User enters in their registered email and password.
3) User clicks the 'Login' button.
4) User is then taken to the application home screen.

Termination Outcome: The user is now logged in.

see Figure 1

Use Case Number: 3
Use Case Name: Create Group
Description: A user has decided that they need a group for a specific purpose. They will click the 'Groups' button, then the 'Add' button to start creating a group.

1) User clicks on the 'Groups' button. They are then taken to the groups page.
2) User then clicks the 'Add' button. They are taken to the group creation page.
3) User then enters in a name for the group and selects the kind of group it is via the toggle switch at the bottom of the screen.
4) User clicks the 'Add' button and is taken back to the groups page.

Termination Outcome: The user has now added a group.

see Figure 4

Use Case Number: 4
Use Case Name: Add Item
Description: A user has decided to add some items to a group. They will click the 'Info' button next to the group name, then 'Add'. This will allow them to add an item.

1) User navigates to the desired group.
2) User clicks the 'Add' button.
3) User then takes a picture of the item they want to add.
4) User enters in an item type and short description for the item.
5) User then clicks the 'Add' button.

Termination Outcome: The user has now added that item to the group.
see Figure 7

Use Case Number: 5
Use Case Name: Delete Item/Group
Description: A user has decided to delete items or a group entirely. They will click the 'Delete' button in the group. This will allow them to delete items or the group.

1) User navigates to the desired group.
2) User clicks on the 'Delete' button.
3) User then selects the items to delete then clicks the 'Delete' button.
4) User confirms they want to delete the items by answering yes to the popup.

Termination Outcome: The user has now deleted those items from their group.
Alternative: Group deletion.

1) User navigates to the desired group.
2) User clicks on the 'Delete' button.
3) User then clicks on the 'Delete Group' button.
4) User confirms they want to delete the group by answering yes to the popup.

Termination Outcome: The user has now deleted that group.
see Figure 8

2.3. Interface Mockups

3. Project Timeline

Phase One: September 30th 2019
Get the core of the Android application up and running.
Get the Firebase Realtime Database set up .
Begin developing the image processing algorithms.

Phase Two: October 31st 2019
Polish up the Android application.
Fully integrate the Database with the application.
Finish the image processing algorithms and add them to the application.

Phase Three: November 14th 2019
Bug fixes for the image processing.

4. Project Structure

Android/Kotlin:
We chose to implement this idea as an Android application as we both owned phones running Android. It was also important to us that the application was easy to work with on the go. It also allowed for easy camera integration into the application.
Firebase Realtime Database:
We chose to use the Firebase realtime database for its ease of use and excellent integration with android applications. We also wanted the data to be safe so that if the app was deleted, no data was lost.

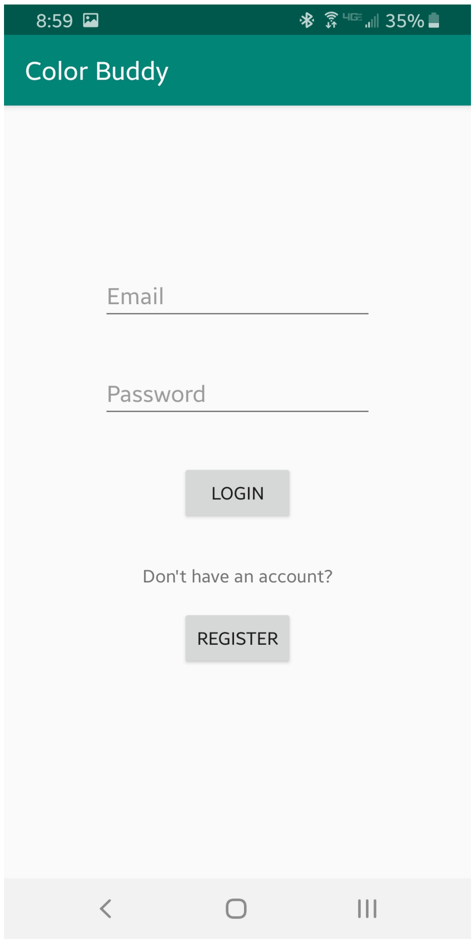

Figure 1. Login/Landing Page

4.1. UML Outline

Here is the basic structure of Color Buddy. 9

5. Results

At the end of the project we had a fully functioning application that did what we had in mind.. Users could create groups, add items to them, and could determine if a new item would match a group. We were not able to implement any stretch goals into our final product.

5.1. Future Work

Machine learning - this would allow the application to differentiate the item from the background in the image.
Improve calculation time - This could be done in a couple ways. It could be off loaded to a server or the phone's GPU could be utilized.
Outfit suggestion - Outfit suggestion could be implemented for wardrobes as the clothing type is stored along with the colors. This would also take the current weather into account.

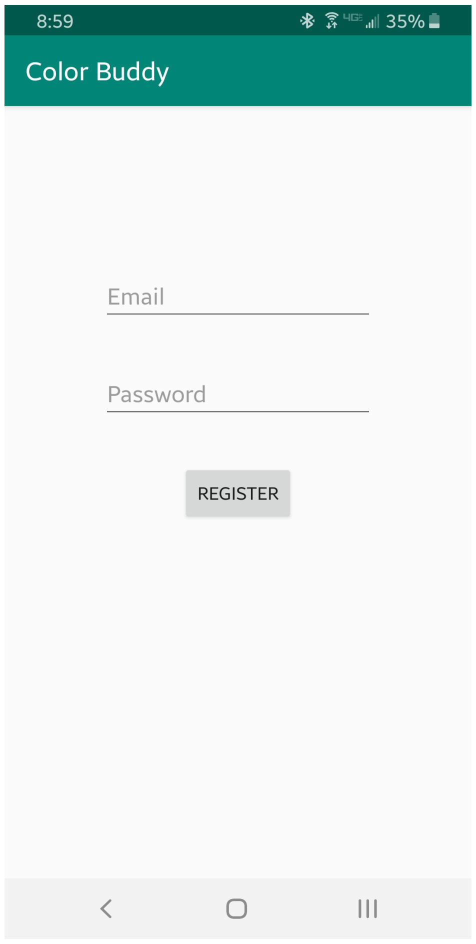

Figure 2. Account Registration

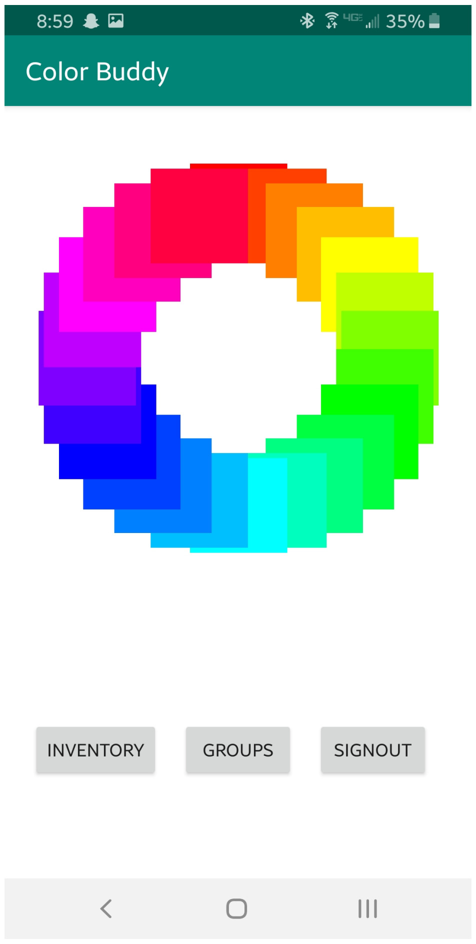

Figure 3. Home

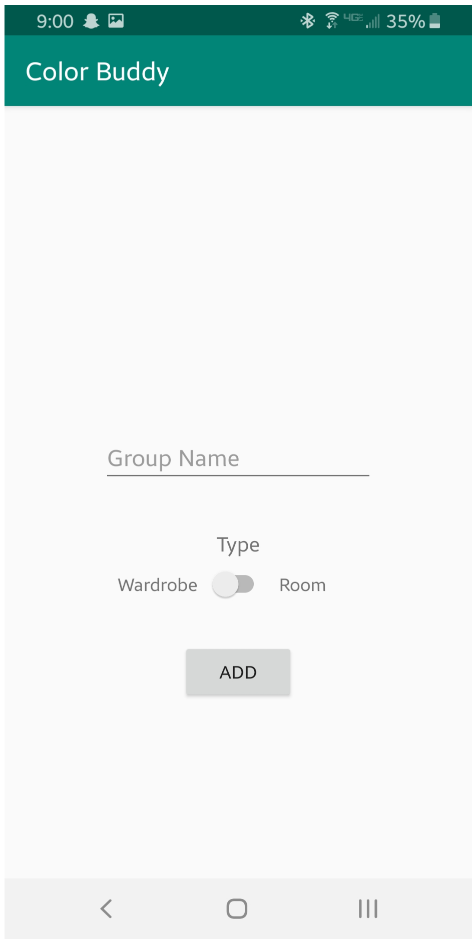

Figure 4. Group Creation

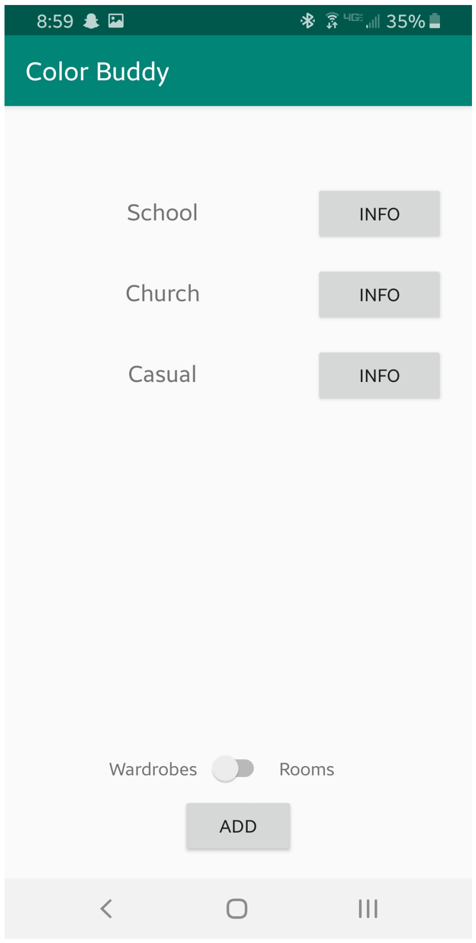

Figure 5. Groups List

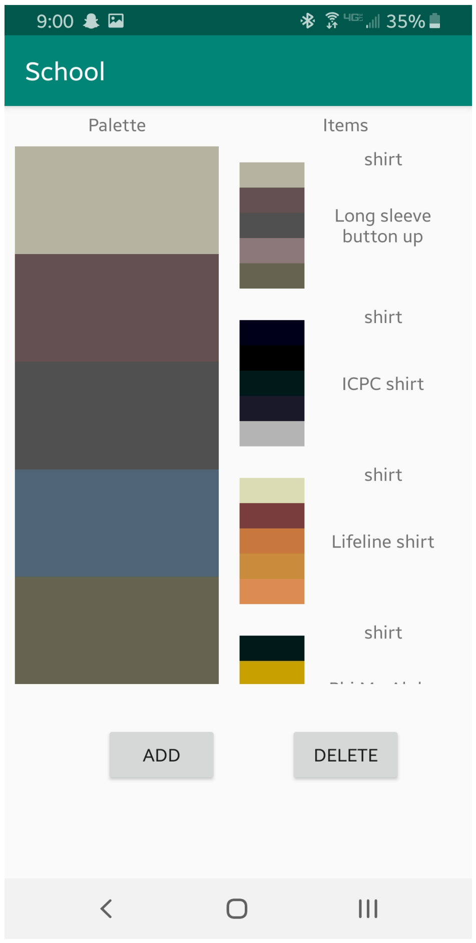

Figure 6. The items within a group

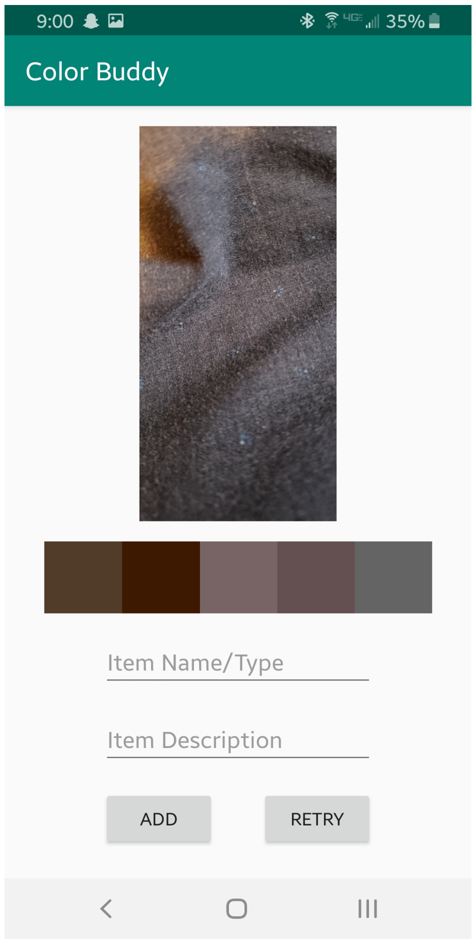

Figure 7. Adding an item to a group

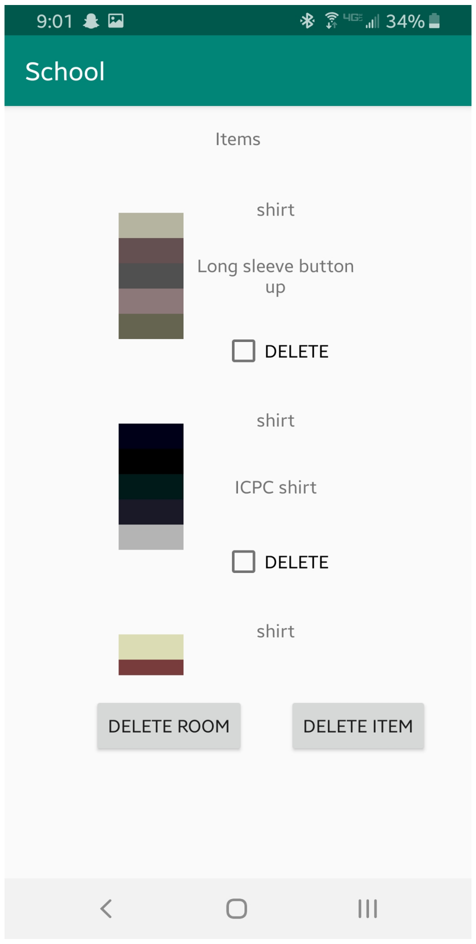

Figure 8. Deleting items or the group

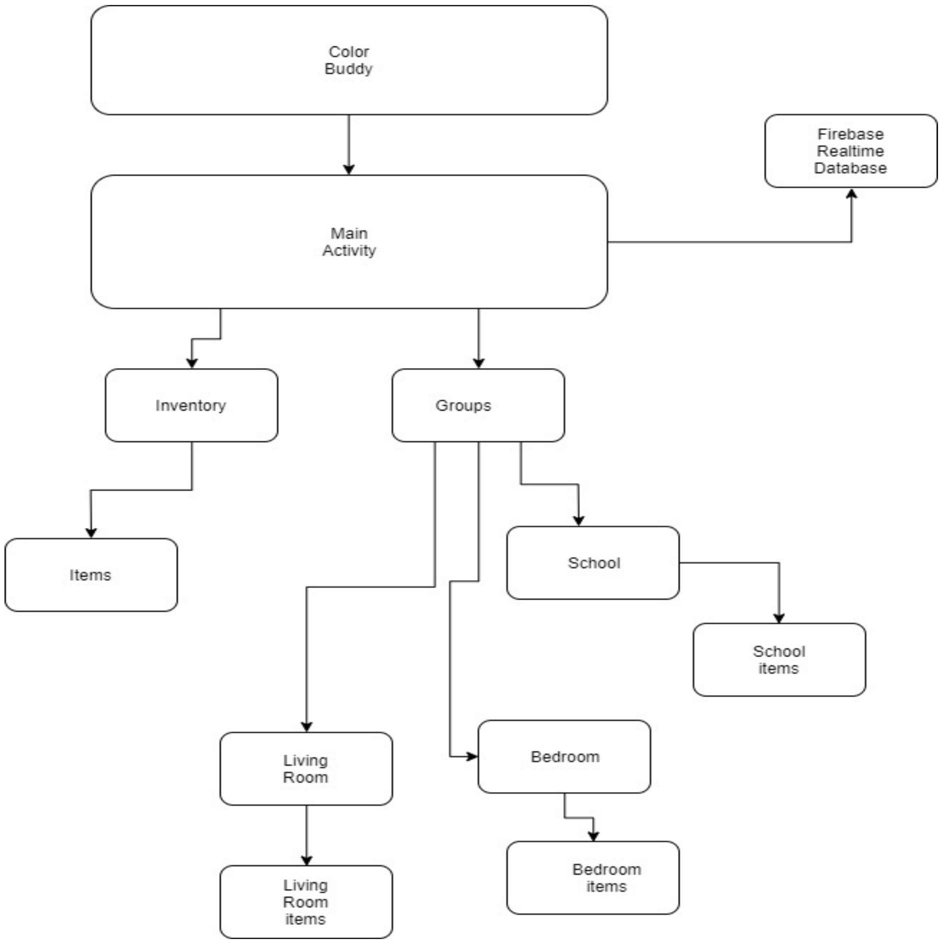

Figure 9. Basic Structure of the Application