Ricardo Rigodon
Assignment 4 OSS

Project Testing

Testing

We have to make our system testing as extensive as possible. If our system has issues, it won't work the way it was intended to. We must test every level of our system to make sure it is operating at peak efficiency. One strategy we could use is performance testing. For my addition to this app, I want to tell the user as quick as possible if the outfit is a match or not. Testing will include setting benchmarks on how long it takes to process the colors to see if they are a match or not. I want my app to be convenient and not a hassle to use. Android offers some tools to help with testing so I will use these to make sure my app is the best it can be.

Espresso for testing UI framework. Android JUnitRunner for test cases.

Also smoke testing and testing as often as possible will help me weed out bugs in the amount of time that we have to do this project.

Error/Defect collection and analysis

Unfortunately creating applications, we are going to run errors and it is very rare the first time we create an application that it runs perfectly the first time. We want to find errors in the development phase as early as possible rather than later so we can fix them. It is easier to fix these errors because early in the development process there is less functionality to be worried about so fixing bugs and such is easier. All it takes is one bug or error to take down a whole system and cause a lot of headaches.

Security Testing

My application does not request any sensitive information so I think security is not as big of an issue. However, the last thing I want is my phone to be a gateway to someone's personal information so making sure the system is safe enough to use is important. And also requesting access to camera, there could be a security flaw allowing a hacker to access location data or photos so it is in my best interest to make sure my app is safe as possible.

Android has the following tool called Traceview that allows you to track the time of method calls which can help our app run faster. We can see how long each method call takes and see if we can make changes to make it faster. One testing method we could try implementing is the ability of the algorithm to guess the correct answers given a list of colors. If a set of colors is a match

and the algorithm returns false, we know something is wrong and should go back and try to figure out where it went bad.

Functionality Tested	**Inputs**	**Expected Output**	**Actual Output**
Color grab	Pixels from camera	Hex code of color	
Color save button	Hex code of color	Save to list	
Color delete button	Position of selected color	Deletes color from list	
Share button	Values of color and image of color shared	Color and stats sent in message	

Use Case Descriptions

Use-Case for seeing the color of an item
Primary actor(s): User
Goal in context: See the color of an item.
Preconditions: The user must be able to open the app.
Trigger: The user is curious about the color of an item and wants to find out.
Scenario:

1. User opens the app on their phone.
2. User clicks the dropper icon on the bottom right to “grab” a color.
3. The camera opens and the user focuses the circle in the center on the color they want to grab.
4. User then clicks on the circle to snap the color and see what color it is.
5. The color is displayed with its corresponding hex code value.

Exceptions:

1. The user closes the app before grabbing a color.

Use-Case of saving a color
Primary actor(s): User
Goal in context: User wants to save a color

Preconditions: The user must be able to open the app.
Trigger: The user sees a color they like and wants to save for a later time.
Scenario:

1. User opens the app on their phone.
2. User clicks the dropper icon on the bottom right to “grab” a color.
3. The camera opens and the user focuses the circle in the center on the color they want to grab.
4. User then clicks on the circle to snap the color and see what color it is.
5. The color is displayed with its corresponding hex code value.
6. The user clicks the save icon (floppy disk) right of the hex value.
7. The user clicks back to see a list of the colors the user grabbed.

Exceptions:

1. The user closes the app before clicking the save icon.

Use-Case of deleting a color
Primary actor(s): User
Goal in context: User wants to delete a color.
Preconditions: The user must have a color saved.
Trigger: The user sees a color they don’t like or want to remove from their saved colors.
Scenario:

1. User opens up app with their saved colors.
2. User clicks the color that they want to delete.
3. Color screen pops up with color, hex value, rgb string, and hsv values.
4. User clicks trash can icon.
5. Dialog pops up asking for user to be sure if they want to delete color. User confirms.
6. Color is deleted.

Exceptions:

1. The user has no colors to delete.

Use-Case of editing a color
Primary actor(s): User
Goal in context: User wants to delete a color.
Preconditions: The user must have a color saved.
Trigger: The user sees a color they might like and want to give it a name.
Scenario:

1. User opens up app with their saved colors.
2. User clicks the color that they want to edit.
3. Color screen pops up with color, hex value, rgb string, and hsv values.
4. User clicks pencil icon.
5. Dialog pops up asking user for the name they want to give color.
6. User types name and clicks edit.
7. Color now has that name.

Exceptions:

1. The user has no colors to edit.

Use-Case of sharing a color
Primary actor(s): User
Goal in context: User wants to share a color.
Preconditions: The user must have a color saved.
Trigger: The user sees a color they
Scenario:

1. User opens up app with their saved colors.
2. User clicks the color that they want to share.
3. Color screen pops up with color, hex value, rgb string, and hsv values.
4. User clicks share icon.
5. Dialog pops up asking user where they want to share the color they found.
6. User chooses the sharing option.
7. User shares the color they found.

Exceptions:

1. The user has no colors to share.

Use-Case of finding if their outfit is a match or not.
Primary actor(s): User
Goal in context: User wants to find out if their outfit matches or not.
Preconditions: The user must have an outfit to match.
Trigger: The user is trying to find out if their outfit's colors match or not.
Scenario:

1. User opens the app on their phone.
2. User clicks the dropper icon on the bottom right to "grab" a color.
3. The camera opens and the user focuses the circle in the center on the color they want to grab.
4. User then clicks on the circle to snap the color and see what color it is.
5. The color is displayed with its corresponding hex code value.
6. The user clicks the save icon (floppy disk) right of the hex value.
7. Repeat steps 4-6 until all desired colors are saved.
7. The user clicks back to see a list of the colors that were grabbed.
8. The user clicks the match button to find out if the colors stored are a match or not.

Exceptions:

1. The user closes the app before clicking the save icon.

2. The user does not save another colors before clicking match icon.