Gbiv System Requirements Specification

Release 1.3

Dux D-zine

Dec 01, 2022

TABLE OF CONTENTS

CHAPTER

ONE

THE CONCEPT OF OPERATIONS (CONOPS)

1.1 Current System

There are generally two types of web applications that exist today with functionality similar to Gbiv. On one hand, there are sites such as Canva and Adobe Color Extractor which take user uploaded photos and generate a palette that matches the colors in the picture [*1*, *2*]. Alternatively, there are sites like Coolors and ColorSpace that allow the user to choose a color (by entering a hex code) and view color palettes that contain the hue [*3*, *4*].

Because color theory is a well known practice in design, there are many more sites that aim to help people choose and analyze colors and color palettes. Many of these applications have use cases adjacent to what Gbiv offers, but currently there is no website that allows users to extract the dominant color from an image and have palettes and related colors recommended.

1.2 Justification for a New System

Color design is everywhere from web development to fashion to interior design and architecture. We often have physical items such as paint, clothes, or furniture where we cannot simply check the hex code to see its color. Gbiv allows users to find the dominant color (in terms of hex) from any image and goes even further by recommending color combinations and palettes based on color theory.

There is a large need for such a system because of the ubiquity of color matching decisions. Of course the professional designers of the world often need such tools, but there are also hundreds of everyday situations where Gbiv would come in handy. Whether you are deciding what pants to wear with your new shirt, repainting the living room, or buying a new couch: color matters. With Gbiv, we will use the science of color to help users make informed design choices without years of color theory experience.

1.3 Operational Features of the Proposed System

The operational features of this proposed system will center around what happens after a user has uploaded an image. First a dominant color will be extracted from the photo and displayed to the user along with their uploaded photo. The site will then output several color palettes that contain this color which will be generated using principles of color theory. Simultaneously, Gbiv will suggest several individual related colors that may also be of use when making design decisions in the physical world.

Beyond this primary function, Gbiv will provide example palettes that can be sorted based on certain features of the colors such as whether the palettes are complementary, monochrome, analogous, etc. On other pages information about color theory that will help users to understand the science behind the application's palettes and suggested colors.

1.4 User Classes

Our application will have a single user class which will be designers of all skill levels. This could be people looking for specific color relationships to something they already have (e.g. a color that complements their green wall) or users who are looking for palettes to inspire overall color schemes. The beauty of our application is that it appeals to professional designers, as well as everyday consumers. Gbiv will introduce color theory and provide an intuitive user experience that will allow color experimentation and education for people of all backgrounds.

1.5 Modes of Operation

There will be a single mode of operation for all regular users of our application. This mode of operation is entered when they visit our site and end up on the main page where photos can be uploaded. In this state, users can navigate around the site and find all the information and functionality provided by Gbiv.

On top of this, our team as developers will have a mode of operation where we can add new palette generators and tweak the code for new releases. For now, only members of DUX D-Zine will be able to enter this mode of operation, but in the future Gbiv could be made open source and opened up for contribution from the community.

1.6 Operational Scenarios

1.6.1 Scenario 1

Uploading an Image

Brief Description:

This use case describes how a user would upload an image to Gbiv in order to trigger the color analysis that is the primary functionality of the website.

Actors:

A user

Preconditions:

1. The user must have a browser that satisfies the requirements necessary to access the Gbiv site.
2. The user must visit Gbiv's URL through their browser.

Steps to Complete the Task:

1. The user selects "Upload Image" on the gray rectangle at the top of the screen
2. A file system GUI will open up where they will be able to select a file to upload
3. The user will select a file that is properly formatted (**NOTE:** Gbiv will accept both .png and .jpg files, but no other image file types)
4. If the input is accepted, Gbiv's analysis and suggestion procedure will begin

Postconditions

The site will now display the dominant color that has been extracted from the image. Below the dominant color, several palettes will be displayed once the color processing has been completed. Once the palettes area is ready to be populated, related colors will also be displayed below the palette section.

CHAPTER

TWO

SPECIFIC REQUIREMENTS

2.1 External Interfaces (Inputs and Outputs)

2.1.1 User Image Uploads (Input)

The user will be able to upload an image to Gbiv in order to extract its dominant color. This is the most important input of the application because it allows users to access the main functionality which is to apply color theory to colors found in the physical world. The photo may be formatted as a .png or a .jpg file and will be restricted to a range of 0 to 99MB.

2.1.2 Generated Palettes (Output)

After uploading an image to be analyzed, our application will generate several palettes that contain the dominant color in the image. These palettes will be composed of 4 separate colors that are labeled with their individual hex codes. Along with the visual output of these four colors, the corresponding hex codes will also be available to end-users. The output will reach the user through our frontend by way of the website UI. Palettes will be displayed as rectangles divided into 4 horizontal bars each filled with one color from the generated palette. Users will be able to view the hex codes of the colors by hovering their mouse over the palette they are interested in.

2.1.3 Related Colors (Output)

Similar to outputted palettes, after a user uploads an image Gbiv will calculate colors based on the extracted dominant color. These colors will be related to the user's color based on principles of color theory which will be explained further in the design specification as well as our site's "color theory" page.

These outputs will come from the backend in a similar format as the palettes, but will be displayed on the website in a different fashion. Instead of a single block with several stripes of color, the related colors will be displayed as squares that are filled entirely with a single color. These squares will be grouped in a logical fashion based on color theory fundamentals and like the palettes, will display their hex codes when hovered over.

2.2 Functions

2.2.1 Color Extractor

This function is the first function that will be called when the user uploads an image. It will be a fairly straightforward function that will take an image as an input (either .jpg or .png) and output the statistical mode pixel color in the picture (represented as a hex string). One key component of this function is that it must output in the appropriate format. There are several ways to digitally represent colors and it is vital that we have an output that can be easily translated between different color formats.

2.2.2 Generate Palettes

This function will act as a kind of "main" function that will be called in order to execute several palette creator functions. The generate palette function will pass the dominant color into these functions and save the outputted palettes. After several palettes have been created, the generate palettes function will return the set of sets to be displayed on the user interface.

2.2.2.1 Palette Creator

This is not a single function, but rather a class of functions that all serve a similar purpose. Each of these functions will take a single color as an input (which will be the color extracted from the user's image) and will output a palette of 4 colors–with one being the inputted color. We will need several of these functions in order to create several types of color-theory-based palettes such as monochromatic, complementary, analogous, and more.

2.2.3 Generate Related Colors

This function acts in a similar way to the "Generate Palette" function in that it will call several sub-functions and output their combined results in a convenient data format. The sub-functions here are the "Related Color Finder" functions which will take an input of a single color and output several related colors. The "Generate Related Colors" function will then pass the resulting colors to be properly displayed in the frontend.

2.2.3.1 Related Color Finder

This is a class of functions that will each take a single color as an input and output a set of colors that are related based on one particular tenet of color theory. There are a variety of defined color relationships in design, so these functions will be separated based on these relationships to maintain modularity and simplicity.

2.3 Usability Requirements

As an application built specifically for design, it is paramount that our site has a well-designed interface. The target users for this application include everyone who has a hand in any kind of design, so people visiting Gbiv will have a range of technical experience and knowledge. To address this, we will have an intuitive user interface that will be supplemented by thorough user documentation. We will also ensure that the site is easily navigable and sufficient information is provided regarding how the colors and palettes are created.

2.4 Performance Requirements

Our process of generating relevant palettes and related colors involves doing lots of math with numerical representations of colors. Complex mathematical equations can often be a bottle-neck for web applications, so we are aiming to optimize these functions as much as possible.

We plan to have performance of certain functions fall within defined bounds. Specifically: we plan to have the related colors and color palettes load within 20 seconds 95% percent of the time and we want all website pages to load within 5 seconds of the user clicking on them or visiting the URL.

2.5 Software System Attributes

Due to the diverse backgrounds of the user base we are targeting, two key system attributes we plan to build into Gbiv are usability and portability. We will achieve usability by having an intuitive user interface and thorough documentation that guides users through the app. Portability will be achieved by having the application web-hosted which will allow anyone with a web browser to use Gbiv. Furthermore, we will be building the frontend using Bootstrap which allows for the future possibility of a responsive site that will work on a variety of device sizes.

A secondary system attribute that will be kept in mind during the design phase is aesthetics. Because our site is providing a service related to design, having a well-designed site adds to our credibility. This system attribute will be secondary to having an operational and fluid site, but still is important to consider for this particular project.

CHAPTER

THREE

REFERENCES

1. Adobe. (n.d.). Extract Theme. Adobe Color. Retrieved November 6, 2022, from https://color.adobe.com/create/image
2. Canva. (n.d.). Color palette generator. Canva Colors. Retrieved November 6, 2022, from https://www.canva.com/colors/color-palette-generator/
3. Coolors. (n.d.). Coolors. Retrieved November 6, 2022, from https://coolors.co/
4. Lavicka, M. (2018). Color Palettes Generator and Color Gradient Tool. ColorSpace. Retrieved November 6, 2022, from https://mycolor.space/

CHAPTER

FOUR

ACKNOWLEDGMENTS

The format of this SRS document was originally created by Professor Juan Flores. Reference to a completed SRS document was provided by Ronny Fuentes, Kyra Novitzky, Jack Sanders, Stephanie Schofield, Callista West with their Fetch project SRS.