Gbiv System Requirements Specification

Release 1.0

Dux D-zine

Nov 09, 2022

TABLE OF CONTENTS

CHAPTER

ONE

THE CONCEPT OF OPERATIONS (CONOPS)

1.1 Current System

There are generally two types of websites that exist today with functionality similar to our application. On one hand, there are sites such as Canva and Adobe Color Extractor which take user uploaded photos and generates a palette that matches the colors in the picture. Alternatively, there are sites like Coolors and MyColorSpace that allow the user to choose a hex color and view color palettes that have that hex color in them.

1.2 Justification for a New System

Design and color theory are everywhere from web development to fashion to interior design and architecture. No existing site has the functionality of our application which provides a powerful tool for applying color theory to the real world. We often have physical items such as paint, clothes, or furniture where we cannot simply check the hex code of the dominant color. Our application combines this analysis of the color in terms of hex along with recommendations for colors to combine it with based on color theory.

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. In addition to this, the site will generate several colors that are related to the dominant color by way of color theory. The final output will be a collection of palettes that have the dominant color in it or are in some way adjacent to the color.

1.4 User Classes

Our application will have a single user class which will be designers of all skill levels who are looking to apply color theory in their design choices. This could be users 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 by providing a crucial service in a convenient way, as well as the designer in the everyday person by introducing color theory and providing a fun way to experiment with colors.

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. On top of this, our team as developers will have a mode of operation where we can upload new palettes and tweak the code for new releases.

1.6 Operational Scenarios

1.6.1 Scenario 1

Uploading an Image

The user who wishes to extract colors from something tangible in the world, such as a piece of clothing, a building, nature, etc. would want to utilize the Image Upload feature. Upon uploading their selected image, our program will compute a single hex color that occurs most in the image, and captures the general essence of the image. Using this color, our program will then compile a list of color palettes, and another list of related palettes that might not necessarily use the initial color, that the user can view.

1.6.2 Scenario 2

Selecting Related Colors and/or Palettes

Any user who wishes to browse through our page of palettes can view a main list of palettes as an archive, or they may enter or select a specific color, and view palettes that work with that color, as well as related palettes.

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 primary 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 image and will be restricted to a range of 0 to [**N**]MB.

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 in the form of *4* colors [**TENTATIVE?**] and will come from our backend as a list structure with *4* strings representing the hexadecimal value of each color in the palette. The output will reach the user through our frontend by way of the website UI. Palettes will be displayed as blocks divided into *4 horizontal* bars each filled with one color from the generated palette.

2.1.3 Related Colors (Output)

Similarly to the palettes that will be generated, after a user uploads an image Gbiv will calculate related 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 in the frontend in a different way. 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. Furthermore, these squares will be grouped in a logical fashion based on color theory fundamentals.

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 and output the most prominent color in the picture. One key component of this function is that it must output the color extracted in the appropriate formats. There are several ways to digitally represent colors and it is vital that we have all the formats that we need at hand for future functions.

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 in an acceptable format.

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 that contains 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.

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 our functions will be divided 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 our site 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 as a guide if users get lost.

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 aim 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 the entire 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 using the app. We will achieve portability 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 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 ourselves 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

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.