Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Konzept einer Anwendung zur Farbanalyse auf Grundlage von Farbpaletten am Beispiel von Kleidung

Bachelorarbeit

Name des Studiengangs

Wirtschaftsinformatik

Fachbereich 4

vorgelegt von

Fabian Frank

Datum:

Berlin, 09.11.2020

Erstgutachter: Dr. Prof. Burkhard Messer

Zweitgutachter: Dr. Hermann Thiel

Vorwort / Abstract

Vor Ihnen liegt die Bachelorarbeit „Konzept einer Anwendung zur Farbanalyse auf Grundlage von Farbpaletten am Beispiel von Kleidung". Diese Bachelorarbeit habe ich als Abschlussarbeit meines Studium der Wirtschaftsinformatik an der Hochschule für Technik und Wirtschaft verfasst. Die Thematik ergab sich aus einer privaten Diskussion zwischen verschiedenen Parteien, welche Farbe mehr Spielraum für farbliche Kombinationen bietet. Im Laufe meiner Recherche ist mir aufgefallen, dass keine Anwendung diesen Problemfall abdeckt. Aus diesem Grund setzte ich mir das Ziel, ein Konzept für eine Anwendung zu schreiben, welche für das Zusammenstellen farblich passender Kleidung benutzt werden kann.

Ich wünsche Ihnen viel Freude beim Lesen dieser Bachelorarbeit.

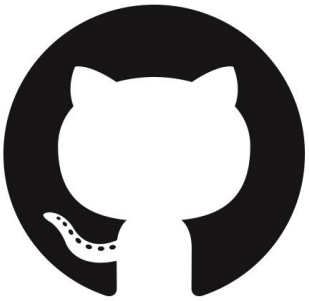
Fabian Frank

Berlin, 09.11.2020

Softwarequellen

Die dieser Arbeit zugrunde liegende Software-Idee wurde parallel ebenfalls vollständig implementiert. Sie finden diese im *Android Playstore* unter dem Namen *DressMeUp*[1].

Im Anhang finden Sie die Algorithmen als Quellcode, welcher auch als öffentliches *GitHub Projekt* einsehbar ist[2].



[1] https://play.google.com/store/apps/details?id=host.exp.dressMeUp
[2] https://github.com/FabianFra/dressMeUp

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Quellcodeverzeichnis

Abkürzungsverzeichnis

JSON	JavaScript Object Notation
HSV	Farbwert H, Sättigung S, Dunkelstufe V
RGB	Rot R, Grün G, Blau B
ECMA	European Computer Manufacturers Association
APK	Android Package
RFC	Request for Comments
UX	User-Experience (Benutzererfahrung)
UI	User-Interface (Benutzeroberfläche)
CMYK	Cyan C, Magenta M, Yellow Y, Key K

1. Einleitung

1.1 Themenfindung

Die Idee dieser Bachelorarbeit ergab sich aus einer Diskussion zwischen verschiedenen Parteien, welche Farbe mehr Kombinationen zulässt: Schwarz oder Braun. Nach einer anfänglichen Recherche und dem Nutzen vorhandener Algorithmen und Anwendungen konnten keine Lösungen gefunden werden, die das beschriebene Problem vollumfänglich abdecken. Da ich meine Bachelorarbeit über das Konzept eines Software-Entwicklungsprozesses am Beispiel einer Anwendung schreiben wollte, entschied ich mich einen Lösungsansatz für das Problem innerhalb dieser Arbeit zu konzipieren.

1.2 Abgrenzung zu bestehenden Algorithmen

Im Internet lassen sich einige Beispiele für Farbgeneratoren finden. Diese generieren durch Anwendung von Farbschemata, auf Basis einer oder mehrerer ausgewählten Farben, Farbpaletten. Als Beispiele dafür können *Adobe Color (Adobe Inc., 2019)*, *Coloors* (Bianchi, 2014) oder *Colormind* (Qiao, 2016) genannt werden. Die generierten Farbpaletten dieser Algorithmen enthalten harmonische Farben, welche durchaus auf die Auswahl von farblich passender Kleidung angewendet werden können. Die Problematik ist hierbei, dass die Zusammenstellung keine äußerlichen Merkmale oder Präferenzen des Anwenders berücksichtigt.

Sucht man nach Lösungen, welche einen größeren Fokus auf den Benutzer setzen, so stößt man auf digitale Farbberatungen wie Outfittery oder Zalon.

1.2.1 Outfittery und Zalon

Outfittery ist ein persönlicher Online-Einkaufsservice, welcher von Julia Bösch und Mitbegründerin Anna Alex gegründet und 2012 veröffentlicht wurde. Heute ziert der Slogan „Dein Stil. Dein Weg" die Startseite von Outfittery (Outfittery GmbH, 2012). Ein vergleichbares Konkurrenzprodukt mit passendem Slogan, „Outfits, die zu dir und deinem Leben passen", entwickelte Zalando und gab es den Namen *Zalon* (Zalando SE, 2015). Dieses wurde im Jahr 2015 in Deutschland, Österreich sowie der Schweiz veröffentlicht.

Vergleicht man beide Slogans, so kann man herauslesen worauf sich der Fokus beider Algorithmen richtet - auf dem Anwender. Die Nutzer bekommen gezielte Fragen gestellt zu Budget, äußerliche Eigenschaften, präferierten Styles, Schnittformen bei Ober- und Unterteilen, Marken, Material, Waschungen bei Jeans und so weiter. Daraufhin wird durch die Anwendung von künstlicher Intelligenz und unter Hinzuziehung von renommierten Modeberatern/-innen ein Vorschlag zu einzelnen Outfits zusammengestellt und dem Benutzer nach Hause geschickt, welche er/sie anprobieren und gegebenenfalls zurückschicken kann.

Die auswählbaren Farben sind jedoch bei beiden Anbietern begrenzt. Favorisiert der Nutzer beispielsweise ein helles Rot, so kann er bei Zalon nur ein Rot oder Bordeaux-Ton auswählen. Outfittery bietet wiederum keine direkte Auswahl der präferierten Farben an.

1.3 Zielsetzung und Abgrenzung

Die Idee des zu erstellenden Konzepts baut auf auf der Grundidee der oben erwähnten Anwendungen auf. Von Seiten der Farbzusammenstellung soll auf bewährte Theorien der Farbenlehre zurückgegriffen und mit Hilfe von *Farbschemata* (vgl. 2.7 Bedeutung und Anwendung der Farbschemata in der Mode), eine auf den Benutzer zugeschnittene Menge an passenden Farben generiert werden. Die betrachteten Parameter werden sich jedoch von denen von Zalon und Outfittery abgrenzen, da ausschließlich die Farben betrachtet werden. Weitere Eigenschaften wie beispielsweise Musterungen und Schnitte werden im Rahmen der Bachelorarbeit nicht betrachtet.

1.4 Erwartetes Ergebnis

Das erwartete Ergebnis dieser Arbeit ist ein Konzept für die Entwicklung einer Anwendung, welche von dem Benutzungsgefühl Zalon und Outfittery ähnelt. Anwender können nach einem Kleidungsstück, zum Beispiel einem T-Shirt oder einer Jacke suchen und bekommen am Ende eine Menge an passenden Farben dargestellt. Präferenzen und das äußerliche Erscheinungsbild des Anwenders sollen hierbei berücksichtigt werden, um die Ergebnismenge zu sortieren. Die Gewichtung der Sortierparameter soll möglichst real sein und zusammen mit Probanden erarbeitet werden.

Zur Veranschaulichung des Prozesses, soll die Anwendung parallel zu diesem Konzept implementiert und im *Google PlayStore* angeboten werden.

2. Theoretische Grundlagen der Farbenlehre

Fundament für das Zusammenstellen von Farben ist die Anwendung von Farbschemata, welche jeweils eine einzigartige visuelle Wirkung auf den Betrachter haben (vgl. 2.6 Farbschemata und ihre Wirkung). Im folgenden Abschnitt werden die theoretischen Grundlagen für die Farbenlehre betrachtet, die für das Konzept vorausgesetzt werden.

2.1 Was sind Farben

Auf die Frage „Was sind Farben?" gibt es nicht nur eine Antwort. Die Antwort hängt von der Sichtweise auf den Anwendungsbereich ab. Betrachtet man die Farbe im naturwissenschaftlichen Rahmen, so ist sie eine physikalische Größe, wie beispielsweise die Wellenlänge oder die Frequenz. Unterschiedlich ist die Betrachtung wie die Menschen die Farben wahrnehmen können, das heißt wie die Kommunikation zwischen dem Auge und dem Gehirn stattfindet.

Der ganzheitliche Ansatz hat jedoch den Menschen als Mittelpunkt. Farben wirken, so Karin Kunkel, immer ganzheitlich auf Körper, Psyche, Geist, Gemüt und Seele (Hunkel, 2011, 15). Farben sind im Allgemeinen wie das Licht, unsichtbar. Das was zu sehen ist, ist die Reflektion oder das Emittieren von Lichtstrahlen eines Objekts (Hunkel, 2011, 16). Warum beispielsweise ein Kästchen als rötlich wahrgenommen werden kann liegt an dem reflektierten roten Licht des Objekts. Dieses gelangt zum Auge und ermöglicht neben der eigentlichen Farbe auch Form und Kontur bildlich aufzubauen.

Die *physikalische Sichtweise* betrachtet Lichtwellen als elektromagnetische Wellen, welche durch zwei messbare Größen bestimmt werden. Dies sind zum Einen die Frequenz, in Hertz angegeben (Hz) und zum Anderen die Wellenlänge, gemessen in Nanometern (nm) (Hunkel, 2011, 16). Die Frequenz definiert wie schnell die Wiederholungen, im Fall der Lichtwellen die Schwingungsvorgänge, in einem periodischen Vorgang aufeinanderfolgen (Spektrum Akademischer Verlag, 2000). Die Wellenlänge beschreibt die Länge zwischen zwei benachbarten Wellenbergen. Jede Farbe besitzt einen definierten Wellenlängenbereich (Hunkel, 2011, 34). Die Augen des Menschen sehen im Normalfall in einem Wellenlängenbereich von 380 nm (Violett) bis 780 nm (Rot). Nicht jedes Lebewesen hat

jedoch den gleichen Wellenlängenbereich. Bienen können zum Beispiel kurzwellige Strahlung sehen (Ultraviolet), dafür aber kein rotes Licht (Mennerich, 2012).

Fällt Licht auf die Netzhaut der Augen, so werden ca. sechs Millionen Zapfen aktiviert, welche für das Sehen zuständig sind. Durch ein elektrisches Signal werden die Informationen an den Sehnerv übermittelt, welcher als Transmitter dem Gehirn das Erhaltene sendet. Dadurch können Farbe, Form, Tiefe und Bewegungen der Materie wahrgenommen werden (Hunkel, 2011, 18). Die Netzhaut des Menschen hat im Normalfall drei verschiedene Zapfentypen mit Fotopigmenten, wodurch die Lichtgrundfarben, Rot, Blau und Grün, ihre eigenständige Mischung und die durch Komplementärbildung entstehenden Farben erkennbar sind (Hunkel, 2011, 18-19).

2.2 Goethes Farbenlehre

Farbtheorien, welche über eine physikalische Definition hinausgehen, bauen mehrheitlich auf Goethes Werk auf (Hunkel, 2011, 26). Ursprung vieler neuzeitiger Farbräder ist das von Goethe entwickelte *Farbrad* (vgl. 5.3 Martian Color Wheel).

Goethe nahm an, dass die Farben ein Zusammenspiel von Licht und Finsternis sind. Betrachtet man den Himmel, so hat er in der Nacht einen Indigo- bis Schwarzton. Durch Aufhellen des Tons durch die Sonnenstrahlen, erhält der Himmel sein helles Blau (Hunkel, 2011, 27). Licht und Finsternis waren für Goethe die Grundfarben Gelb und Blau (Wäger, 2017, 186). Zusätzlich betrachtete er die Nachbildungen des *Sukzessivkontrast* (vgl. 9.1 Begriffsdefinitionen) und die farbigen Schatten. Dies hatte er mit einem Experiment realisiert. Hierbei schaut man für ungefähr zwanzig Sekunden auf einen roten Punkt und danach auf ein weißes Blatt Papier. Im Idealfall ist dann die Komplementärfarbe von Rot, das Grün, zu sehen (vgl. Abb. 1)

Auf Rot reagiert das Auge mit → Grün = Blau plus Gelb

Auf Gelb reagiert das Auge mit → Violet = Rot plus Blau

Auf Blau reagiert das Auge mit → Orange = Rot plus Gelb

Abb. 1: Goethes Farbkreis

Die Polarität der Nachbildungen veranlasste Goethe das Purpurrot (Magenta) als Komplementärfarbe von Grün hinzuzufügen (Hunkel, 2011, 25). Mit diesen Erkenntnissen schuf Goethe in seiner Farbenlehre die drei *physischen* Grundfarben Rot (Magenta), Gelb und (Cyan-) Blau. Durch Mischen dieser, können alle weiteren Farben mit Nuancierungen gebildet werden. Die finale Version des Farbkreises besteht aus den Farben Gelb, Blau, Purpurrot und den drei Mischfarben Grün, Orange und Violett (Goethe, 1809).

2.3 Additive und Subtraktive Farben

Additive Farben, auch Lichtfarben, sind Farben, welche immer von einer Lichtquelle emittiert werden. Mischt man die Grundfarben des additiven Farbkreises, Rot, Grün und Blau, so entsteht ein Weißton. Merkmal der additiven Farben ist, dass durch Mischen der Primärfarben der resultierende Ton heller erscheint. Kombiniert man die additiven Grundfarben miteinander, so entstehen die subtraktiven Grundfarben Magenta, Cyan und Gelb (Hunkel, 2011, 32).

Subtraktive Farben werden auch als physische Farben bezeichnet und haben die drei von Goethe definierten Grundfarben: Rot (Purpur bei Goethe, heute Magenta), Cyan-Blau und Gelb. Das Mischen aller subtraktiven Farben resultiert in einem Schwarz- oder einen schmutzigen Grauton. Diese Farben werden als *subtraktiv* bezeichnet, da beim Mischen den Farben Licht entnommen wird. Dadurch wird die Farbmischung als *dunkler* wahrgenommen (Hunkel, 2011, 31). Mischt man die subtraktiven Primärfarben, so entstehen die Sekundärfarben Rot, Violett und Grün (vgl. 2.8 Etablierte Farbräume).

2.4 Farbwirkung auf das Befinden

Heutzutage werden Farben gezielt eingesetzt, um Gefühle beim Betrachtenden auszulösen. Diese können von negativer- oder positiver Natur sein, welche auf unseren Urinstinkt, unsere Kultur als auch auf die Assoziation mit der Farbe zurückzuführen sind. Betrachtet man beispielsweise *die Farbe Rot*, so steht sie in vielen Kulturen für Leidenschaft, Energie und Glück. Der Farbton steht jedoch auch für Gefahr und Hass. Der Farbton wird auch als *Signalton* bezeichnet, da er mit Gefahren in der Natur wie Blut oder Feuer assoziiert wird.

Nimmt man wiederum *die Farbe Grün*, so steht sie für Erholung, Natürlichkeit und Hoffnung und hat eine entspannende Wirkung auf den Körper.

Das folgende Beispiel zeigt den gezielten Einsatz von Farben und ihrer Wirkung im Bereich der Wandgestaltung.

Unter der Anleitung von einem Farbforscher wurden an der Helios Klinik in Wuppertal die Farbgestaltung der Wände überarbeitet. Einige Patienten hatten nach dem Erwachen aus der Narkose Orientierungsprobleme, welche zum Teil in Panikattacken resultierten. Das lag daran, dass die Wände in einem sterilen Weiß gehalten wurden und dazu ein grelles weißes Licht das Sichtfeld noch monotoner hat wirken lassen. Des weiteren waren die Ergebnisse einer Umfrage nach dem Wohlbefinden in der Klinik von Arbeitern und Patienten als Mangelhaft bewertet worden.

Zur Orientierung innerhalb der Flure gestaltete man die Wände in einem Oca-Ton. Der Empfang und die Zugänge wurden umrandet mit einer Nuancierung der gewählten Wandfarbe, um diese abzusetzen. Die Intensivzimmer der Uniklinik wurden in beruhigenden Pastelltönen gestaltet und in den Büros der Mitarbeiter entschied man sich für ein helles Blau, welches die Konzentration steigern sollte. Um die Erholung innerhalb der Pausen zu erhöhen wurden die Wände der entsprechenden Räume in einem frischen Grün gestrichen. Fazit: Die Umgestaltung erhielt eine positive Resonanz und der prozentuelle Einsatz von Beruhigungsmitteln, so die Uniklinik, sank drastisch (BR Fernsehen, 2020).

Betrachtet man den Bereich des Webdesign, so spielen Farben für die *Corporate-Identity* (vgl. 9.1 Begriffsdefinitionen) eine wichtige Rolle und werden verwendet um beispielsweise Blicke von Anwendern zu lenken (Evernine Insights, 2019).

2.5 Bedeutung der Ablehnung von Farbe

Ablehnung einer Farbe ist nicht in einer Ursache begründet. Es ist vielmehr ein psychologisches Phänomen, welches sich von Mensch zu Mensch unterscheidet. Definiert wird dieses, durch die Erfahrungen und die dadurch entstehenden Assoziationen (Hunkel, 2011, 104). Sabine Gimm schreibt in ihrem Blogeintrag, dass sie eine Ablehnung gegen die

Farbe Braun besitzt. Diese Ablehnung begründet sie anhand von persönlichen Erfahrungen und einer Abneigung gegen Rechtsextremismus[3] (Gimm, 2019).

2.6 Farbschemata und ihre Wirkung

Ein Farbschema definiert eine Beziehung zwischen verschiedenen Farben. Das Arbeiten mit diesen spielt vor allem in den Bereich des Grafikdesigns und bei Benutzer-Schnittstellen von Anwendungen eine essentielle Rolle (vgl. 6.7 SearchForTop). Farbschemata unterscheiden sich u.a. in der Anzahl von den gewählten Basisfarben und des angewandten Kontrastes.

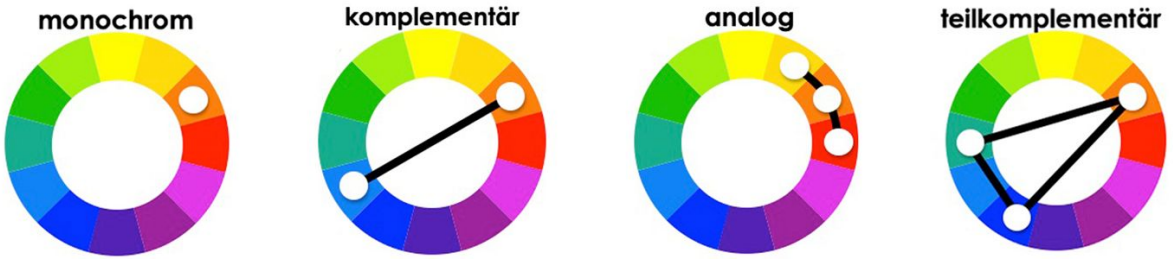


Abb. 2: Beispielhafte Farbschemata

2.6.1 Monochromatisches Farbschema

Um eine *monochromatische* Beziehung zu kreieren werden die Nuancen der Basisfarbe ausgewählt, das heißt die Farben der Farbfamilie. Das Schema wirkt allgemein harmonisch und ruhig. Die Grundwirkung der Beziehung wird jedoch definiert von der ausgewählten Basisfarbe.

2.6.2 Komplementäres Farbschema

Möchte man eine kontrastreiche Farbbeziehung erzielen, so benutzt man den *Komplementärkontrast*. Das Schema ist ein Zweiklang, in welchen sich die Farben auf dem Farbkreis gegenüberliegen. Die Kombination wirkt akzentuiert, laut und spannungsgeladen (vgl. 6.3 Programmatischer Lösungsansatz für die Farbtonverschiebung).

[3] Die Farbe braun wurde 1925 als Farbe des Nationalsozialismus erklärt und stand damals für Verbundenheit mit Heimat und Land (Focus Online, 2013).

2.6.3 Analoges Farbschema

Das *analoge Farbschema* besitzt drei Basisfarben, welche im Farbkreis nebeneinander liegen. Die Kombination derer wirkt meist einheitlich, harmonisch und ruhig. Ein beispielhaftes analoges Farbschema ist die Kombination aus Rot, Orange und ein dunkles Gelb (vgl. Abb. 2).

2.6.4 Teilkomplementäres Farbschema

Das *teilkomplementäre* Farbschema besitzt, wie das analoge Farbschema, drei Basisfarben. Dieses setzt sich aus einer Farbe und den analogen Farben der Komplementärfarbe zusammen. Es hat eine ähnliche, aber nicht so intensive Wirkung wie das komplementäre Farbschema.

2.7 Bedeutung und Anwendung der Farbschemata in der Mode

Mit der gezielten Auswahl an Farben kann im Rahmen der Garderobe eine gewollte Wirkung beim Betrachter hervorgerufen werden. Sie werden verwendet, um Körperstellen zu retuschieren oder den Fokus auf sie zu setzen. Mit kontrastreichen Harmonien lassen sich Kleidungsstücke farblich voneinander trennen (vgl. 2.6.2 Komplementäres Farbschema oder 2.6.4 Teilkomplementäres Farbschema). Sollen diese sich ergänzen, so werden kontrastarme Farbharmonien angewendet (vgl. 2.6.1 Monochromatisches Farbschema oder 2.6.3 Analoges Farbschema).

2.8 Etablierte Farbräume

In den einzelnen Anwendungsgebieten wird mit unterschiedlichen Farbräumen gearbeitet. In diesem Abschnitt sollen die einzelnen Farbräume betrachtet werden, um ein Grundverständnis für die unterschiedlichen Anwendungszwecke aufzubauen.

Gängige Farbmodelle unterscheiden sich in ihrer Beschreibung der Farbe und in ihrem Nutzungsbereich. Farbmodelle werden in zwei Hauptmodelle unterteilt - die technisch-physikalischen und die wahrnehmungs-orientierten Modelle.

T*echnisch-physikalische Modelle* erzeugen die Farbwahrnehmung aus realen oder idealisierten farbgebenden Stoffen (vgl. 2.8.1 RGB-Farbraum und 2.8.2 CYMK-Farbraum). W*ahrnehmungs-orientierte Modelle* beschreiben wiederum die Farbe durch Helligkeit, Sättigung und Farbton (vgl. 2.8.3 HSV-Farbraum).

2.8.1 RGB-Farbraum

Das RGB-Farbmodell ist ein technisch-physikalisches Modell, welches auf den drei *additiven* Grundfarben, Rot, Grün, Blau aufbaut. Durch Mischen der Farben wird die Farbwahrnehmung erzeugt. Dieses Farbmodell findet ihren Nutzen hauptsächlich in den Bereichen der Bildwiedergabe und Bildaufnahme (Rödiger, 2018).

2.8.2 CYMK-Farbraum

Der CMYK-Farbraum ist, wie der RGB-Farbraum, ein technisch-physikalisches Modell, welches jedoch auf die *subtraktiven* Grundfarben, Cyan, Magenta und Gelb aufbaut. Zusätzlich wird in dem Modell auch der Schwarzanteil bei der Mischung hinzugezogen. Anwendungsbereiche sind die Drucktechnik, Computertechnik und Lichttechnik (Rödiger, 2018).

2.8.3 HSV-Farbraum

Der HSV-Farbraum zählt zu den *wahrnehmungsorientierten* Farbmodellen, welche die Farbe anhand des Farbwertes, der Farbsättigung und des Hellwerts beschreibt. Der Farbwert, repräsentiert den Farbwinkel (0° für Rot, 120° für Grün, 240° für Blau). Der Sättigungswert des Farbtons wird mit einer Prozentzahl von 0% bis 100% beschrieben. Ein Farbton mit einem Sättigungswert von 0% gilt als Neutralgrau (vgl. 9.1 Begriffsdefinitionen). Besitzt die Farbe einen S-Wert von 50%, so ist diese wenig gesättigt. Liegt der Wert wiederum bei

100%, so ist die Farbe gesättigt. Der Hellwert wird wie der Sättigungswert in einer Prozentzahl zwischen 0% und 100% angegeben. Hat die Farbe beispielsweise eine Hellwertstufe von 0%, so hat diese keine Helligkeit. Liegt der Wert wiederum bei 100%, so besitzt die Farbe ihre voll Helligkeit (Eichner, 2015).

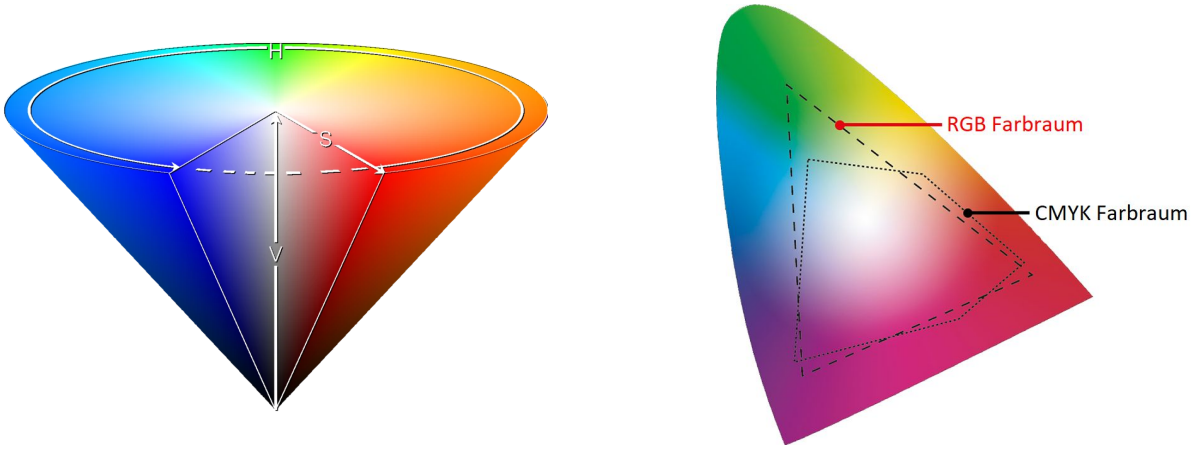


Abb. 3: Gegenüberstellung wahrnehmungs-orientierter und technisch-orientierter Farbräume (l: HSV, r: RGB/CMYK)

3. Konzeptioneller Ansatz der Anwendung

Innerhalb eines konzeptionellen Ansatzes gilt es, wichtige Funktionalitäten zu definieren, sowie die Nutzung und die Benutzergruppen abzugrenzen. Der folgende Abschnitt beschäftigt sich im Rahmen des Konzepts mit dem definierten theoretischen Ansatz der Anwendung.

3.1 Abgrenzung des Nutzens des Anwendung

Die mobile Applikation die in dieser Arbeit beispielhaft konzeptioniert und implementiert wird, soll als Werkzeug für die Unterstützung des Anwenders bei der Zusammenstellung farblich passender Kleidung dienen. Sie wird verwendet, um Entscheidungen der farblichen Zusammenstellung zu erleichtern und das Auge des Benutzer darin zu schulen, Farbharmonien und Disharmonien zu identifizieren.

3.2 Abgrenzung der Benutzergruppen

Die fokussierte Benutzergruppe sind Menschen, welche sich für Mode und ihr äußeres Erscheinungsbild interessieren. Die anvisierte Altersgruppe der Anwender ist ab 14 Jahren unabhängig von Geschlecht oder Orientierung.

3.3 Logischer Aufbau

Eine der Grundvoraussetzungen der Business-Idee und ihrer Implementierung ist die Erstellung eines Anwenderprofils. Der Benutzer bekommt in dem Prozess gezielte Fragen gestellt, um Augenfarbe, Haarfarbe und Hautton herauszufinden. Anhand der zusammengestellten Daten wird dann der *Farbtyp* des Anwenders identifiziert (vgl. 5.4 Das Vier-Farbtyp-System). Die Daten sollten in einer Datenstruktur gespeichert werden, welche es ermöglicht die Benutzerdaten zu einem späteren Zeitpunkt einzusehen oder zu ändern. Falls der Anwender die Profilerstellung erfolgreich abgeschlossen hat wird er zum Hauptmenü geführt. Dort soll er die Möglichkeit besitzen nach Kleidungsstücken, wie beispielsweise T-Shirt oder Jacke, zu suchen. Sucht der Anwender jetzt zum Beispiel nach einem passenden T-Shirt, so werden ihm Fragen zu Präferenzen und zu Farben seiner Garderobe gestellt. Am

Ende soll der Anwender eine Menge an passenden Farben vorgeschlagen bekommen, welche auf Grundlage seiner Antworten sortiert sind.

3.4 Essentielle Funktionalitäten

Im folgenden Abschnitt sind die essentiellen Funktionalitäten aufgeführt, welche die Endanwendung abdecken muss.

1. Der Anwender erstellt beim ersten Benutzen sein Profil, in welchem er Angaben zum eigenen Äußerlichen und zu seinen Präferenzen angibt.
2. Der Anwender kann sein Profil zurückzusetzen.
3. Der Anwender kann Antworten von gestellten Fragen korrigieren.
4. Der Anwender kann aus einer Menge von benannten Farben auswählen.
5. Der Anwender kann nach einem T-Shirt suchen, welches zu seinem zusammengestellten Outfit farblich passt.
6. Der Anwender bekommt nach der Suche eine sortierte Übersicht von farblich passenden Vorschlägen zum gesuchten Kleidungsstück.
7. Die Sortierung der am Ende angezeigten Farben beruht auf den Antworten des Anwenders

Der Fokus der Anwendung liegt auf dem Anwender. Es gilt gezielte Fragen zu stellen, um am Ende eine auf ihn zugeschnittene Farbauswahl vorzuschlagen. Hierbei müssen die einzelnen Parameter in einer Art gewichtet sein, dass bei der dargestellten Ergebnismenge die Präferenzen des Benutzers als auch die Erkenntnisse der Farbtheorie in angebrachten Maß berücksichtigt werden. Die Benutzerführung muss leicht verständlich sein.

4. Lösungsansatz für die Entwicklung

Für die Umsetzung der Konzept-Idee als Anwendung sind folgende Entscheidungen notwendig, die nachfolgend exemplarisch beschrieben werden.

4.1 Definition der Plattform

Zuerst muss definiert werden, in welcher Skript- oder Programmiersprache der Algorithmus implementiert werden soll. Dafür müssen die Vor- und Nachteile der jeweiligen Plattform betrachtet und gegeneinander abgewogen werden. Es ist wichtig das Anwendungsgebiet und das gewünschte Betriebssystem zu definieren, da diese Faktoren einen Einfluss auf die *Plattform-Entscheidung* haben. Für den Rahmen des Konzepts wird die Skriptsprache JavaScript (ECMA 6) benutzt.

4.2 Frameworkauswahl

Auf Basis der Plattform-Entscheidung wird die Anwendung i.d.R. auf bereits existierenden Bibliotheken aufgebaut. Diese sind ein Gerüst aus anwendungsunabhängigen *Basisfunktionalitäten*. Bei der Wahl der Bibliotheken gilt es, analog der Plattform-Definition, Vor- und Nachteile zu betrachten und diese abzuwägen. Für den Realisierungsprozess des Konzepts wurden folgende Bibliotheken ausgewählt.

4.2.1 TinyColor

TinyColor ist es eine ressourcenarme JavaScript-Bibliothek für das Arbeiten mit Farben. Der Konstruktor braucht lediglich ein valides Darstellungsformat, wie HSV, HSL, RGB (vgl. 2.8 Etablierte Farbräume) oder Farbnamen, woraus ein Javascript Objekt instanziiert wird. Auf diesen lassen sich mehrere farbspezifische Hilfunktionalitäten ausführen, wie Farbkonvertierungen (vgl. 9.1 Begriffsdefinitionen) und Farbmanipulationen (Cranston, 2014). TinyColor wird in der Realisierung benutzt, um Farben zu identifizieren und diese nach Helligkeit zu sortieren, die Lesbarkeit von Texten zu garantieren und Farbkonvertierungen auszuführen.

4.2.2 React Native

React Native ist ein Open-Source Framework, mit welchem sich native Applikationen für Android und iOS entwickeln lassen. Aus geschriebenen JavaScript und React Code wird nativer Quelltext generiert, welcher dann von dem Zielbetriebssystem interpretiert werden kann. Das Paradigma von React beschreibt, dass sowohl das Rendering als auch andere UI-Logiken grundsätzlich zusammenhängen (FaceBook Inc., 2015). React Native wird im Rahmen der Realisierung des Konzepts benutzt, um eine beispielhafte Integration in eine native mobile Applikation aufzuzeigen. Zusätzlich wird die syntaktische Erweiterung JSX benutzt, um React-Elemente erzeugen zu lassen. Die Sprache React ist jedoch nicht auf JSX angewiesen. Im Laufe des Realisierungsprozess hat sich herausgestellt, dass es ein hilfreiches Werkzeug zum Arbeiten mit UI-Elementen und Logik darstellt.

4.2.3 Async Storage

AsyncStorage ist ein unverschlüsseltes, asynchrones, persistierendes, Schlüssel-Wert Speichersystem für React Native Applikationen, welche nicht auf einen Server bzw. eine externe Datenbank angewiesen sind (React Native Community, 2020). In der Realisierung des Konzepts wird das Framework für die Speicherung von Anwenderdaten eingesetzt.

4.2.4 React Navigation

In *React Native* werden für das Wechseln zwischen Ansichten verschiedene Möglichkeiten angeboten. Neben der allbekannten Tab-Navigation wird auch die Drawer-Navigation und Stack Navigation unterstützt. Für die Realisierung der Anwendung wurde sich für die *Stack Navigation* entschieden (vgl. Abb. 4). Diese findet ihren Nutzen im Transitieren zwischen verschiedenen Ansichten und in der Verwaltung der Navigations-Historien. Die Ansichten werden auf einen sogenannten Stack postiert und über push- und pop-Befehle verwaltet (Expo & Softwaremansion, 2020).

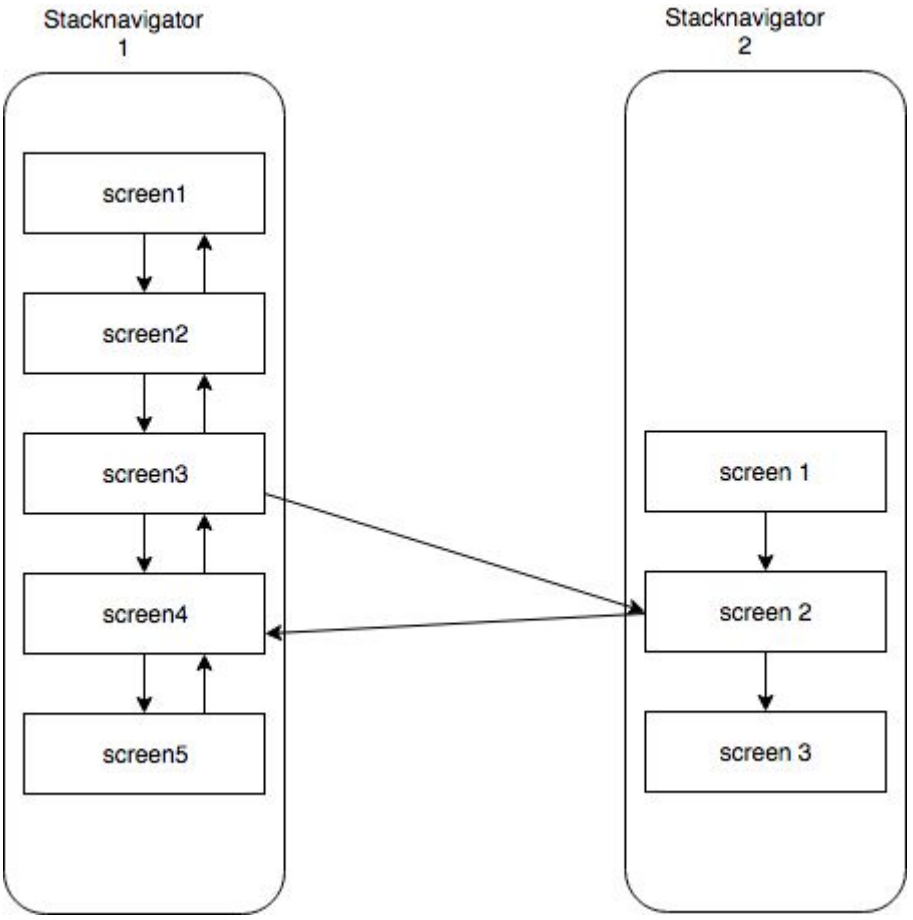


Abb. 4: Stack Navigation in React Native

4.2.5 Expo

Expo ist eine Open-Source Plattform für das Entwickeln von Android-, iOS- und Web Applikation mit JavaScript und React. Es bietet ein Grundgerüst aus Werkzeugen, welche das Entwickeln, Testen und Veröffentlichen von React Native Anwendungen vereinfacht (*Expo*, 2020). Im Rahmen der Realisierung des Konzepts unterstützte Expo den Entwicklungsprozess und wurde genutzt, um die *APK* für die veröffentlichte Applikation zu generieren.

4.3 JavaScript Object Notation

Die *JavaScript Object Notation* (JSON) ist eine strukturierte, plattformunabhängige Notation zur Darstellung von Daten, die den Datenaustausch zwischen verschiedenen Anwendungen ermöglicht. Die Notation wird seit 2017 von zwei verschiedenen, inhaltlich jedoch gleichen Standards spezifiziert. Das sind zum einen das *RFC 8258* (Internet Engineering Task

Force, 2017) und zum Anderen das *ECMA-404* (Ecma International, 2017). Heutzutage bieten viele Programmiersprachen eine Unterstützung des JSON-Formats, sie benötigen lediglich einen Generator (Serialisierung) und einen Parser (Deserialisierung).

Die Syntax der Notation beschreibt Objekte als *Key-Value-Paare*. Hierbei ist der Key eine Zeichenkette und das Value kann vom Typ *null*, *boolean*, *integer*, *string*, *array* oder *object* sein. Values können dadurch auch Objekte oder Mengen aus Objekten enthalten, wodurch eine Verschachtelung möglich ist. Es wird vorausgesetzt, dass das JSON mit einem Objekt oder einem Array beginnt.

```
{
	"name": "Grey",
	"hsv": "hsv(0, 100%, 50%)",
	"hex": "#808080",
	"specifications": [...]
}
```

Quellcode-Beispiel 1: JSON-Beispiel

In *Quellcode-Beispiel 1* stellt einen Ausschnitt eines achromatischen Farbobjekts aus dem Datenbestand der Anwendung dar. Der *name-, hsv-* und *hex-Wert* referenzieren string-Werte. Der Wert des Schlüssels *specifications* besitzt eine Menge aus Farbobjekten, die Graustufen repräsentieren.

5. Fachliche Anforderungen des Anwendungskonzeptes

Neben den programmiertechnischen Anforderungen müssen auch die *Fachanforderungen* der Anwendung berücksichtigt werden. Nachfolgend wird der Umgang am Beispiel der Farbapplikation gezeigt.

5.1 Business-Konzept

Anwendungen im kommerziellen Kontext basieren in der Regel auf einer *Business-Idee*, die den eigentlichen Wert der Anwendung und ein Alleinstellungsmerkmal darstellt. Die Idee im gewählten Beispiel kann typischerweise mit verschiedenen Plattformen (vgl. 4.1 Definition der Plattform) gleichwertig implementiert werden. Die Business-Idee dieser Arbeit ist die Kombination zweier Theorien.

5.1.1 One-Color-Trick

Der *One-Color-Trick* ist eine etablierte Methode in der Zusammenstellung von Kleidung, welche auf der Kombination von nicht neutralen- und neutralen Farben basiert. In der Theorie lassen sich ohne große Bedenken neutrale Farbe miteinander kombinieren. Besteht das zusammengestellte Outfit ausschließlich aus neutralen Farben, so kann dazu eine beliebige Farbe getragen werden. Ist die farbliche Zusammenstellung des Outfits wiederum eine Kombination aus neutralen Farben und maximal einer nicht neutralen Farbe, so kann eine Farbe der gleichen Farbfamilie gewählt werden (Vantongeren, 2018).

5.1.2 FabscheAlgorithmus

Der *FabscheAlgorithmus* kombiniert die Theorien der Farbenlehre (vgl. 2. Theoretische Grundlagen der Farbenlehre) und der Stilberatung. Das erwartete Ergebnis ist eine Menge aus passenden Farben, welche nach den Präferenzen des Anwenders sortiert sind. Es gilt ein Fundament zu errichten für die Suchalgorithmen der einzelnen Kleidungsstücke. Für den Rahmen des Konzepts wurde die Suche nach einem passenden Oberteil beispielhaft realisiert (vgl. 6.7 SearchForTop).

Für das Zusammenstellen sollten verschiedene Parameter betrachtet werden (vgl. Tab. 1). Zum Einen sollte das Oberteil im Kontrast zur Hautfarbe stehen beziehungsweise nicht zu nah an den Farbton der Haut dran sein. Besitzt der Anwender beispielsweise eine hellere Haut, so sollte dieser eher einen dunklen Farbton für das Oberteil wählen. Für eine dunkle Haut gilt das Entgegengesetzte. Menschen mit oliven Hautton können alle Töne tragen, solange sie auf ausreichenden Kontrast achten (Creative Ventures Inc., 2013).

Zu Anderen ist das Zusammenspiel der einzelnen Kleidungsstücke im Aufbau eines vollständig farblich harmonisierenden Outfits ein leitender Faktor. Aus diesem Grund ist die Betrachtung der Farben des bestehenden Outfits ein essentieller Bestandteil des Algorithmus.

In der Theorie sollte die Farbe des Oberteils in Kontrast zur Hosenfarbe stehen und ihr nicht gleichen. Es gibt jedoch Ausnahmen wie zum Beispiel ein schwarz auf schwarz. Niedriger gewichtet ist die Farbe der Schuhe, welche im Optimalfall die Farbe des Oberteils unterstützt beziehungsweise ähnelt. Trägt der Suchende eine Jacke, so gilt es einen Kontrast zwischen Jacke und Oberteil zu gewährleisten.

Mit den gezielten Einsatz von Farben können gewünschte Wahrnehmungen im Betrachter erzeugt werden (vgl. 2.4 Farbwirkung auf das Befinden). Möchte der Anwender zum Beispiel die Blicke auf seine Augen lenken, so kann er ein kontrastreiches Farbschema anwenden. Das monochromatische- oder analoge Farbschema kann wiederum benutzt werden, um den Augen zu schmeicheln (Creative Ventures Inc., 2013).

5.2 Fachliche Grundlagen des Business-Konzepts

Das Business-Konzept basiert in der Regel auf etablierten öffentlich zugänglichen Theorien. Im Beispiel sind Farben das Herz der Anwendung, da diese die auswählbaren Farben für den Anwender sind. Daher ist es wichtig eine ausreichend große Menge an benannten Farben dem Benutzer zur Verfügung zu stellen. Es gibt zahlreiche Farbräder, wie Isaac Newtons von 1704, Goethes Farbkreis von 1810 (vgl. 2.2 Goethes Farbenlehre) oder Ittens Modell von 1961, welche jedoch eine unzureichende Menge an Farben inkludieren. Für die beispielhafte Realisierung des Konzepts wurde sich für das Martian Color Wheel, von Warren Mars entschieden. Die Entscheidung ist begründet durch eine Analyse von bestehenden Farbräder und anschließender Bewertung dieser.

5.3 Martian Color Wheel

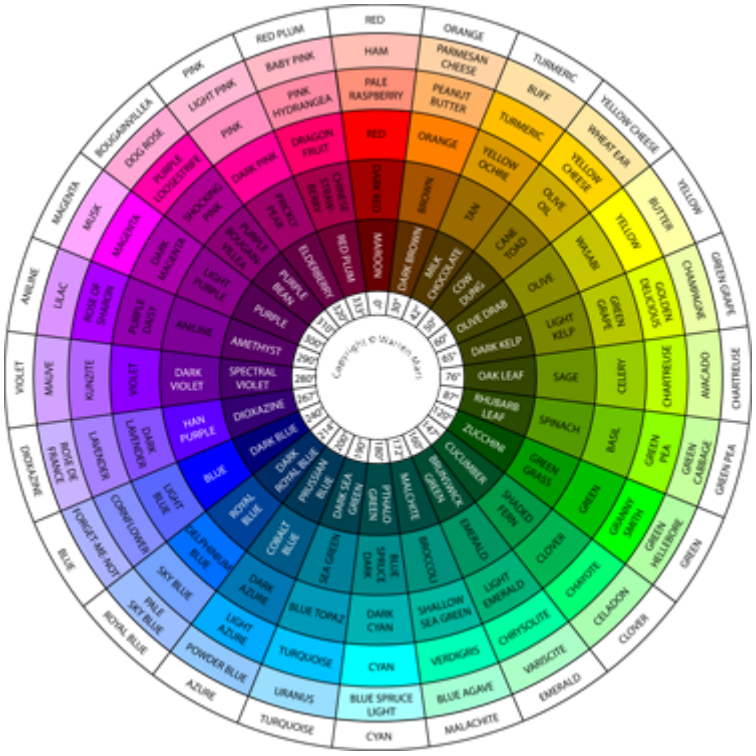


Abb. 5 - Martian Color Wheel (Mars, 2014)

Das *Martian Color Wheel* (Mars, 2014) ist ein von Warren Mars entwickelter Farbkreis (vgl. Abb. 5), welchen er 2014 in seinem Blog veröffentlichte. Dort schreibt er über seine Motivation und den Entstehungsprozess des Farbrads. Für ihn sind die Mengen der Farben in vorhandenen Farbkreisen zu gering und die Benennung derer war entweder nicht vorhanden, inkorrekt oder nichtssagend. Zusätzlich bemängelt er, dass in den meisten Modellen nur die Hauptschatten des Farbtons inkludiert wurden, ohne weitere Farbtönungen der Farbe zu betrachten.

Das Martian Color Wheel basiert auf dem HSV-Farbmodell (vgl. 2.8.3 HSV-Farbraum) und umfasst 24 Farbtöne, mit jeweils vier Farbabstufungen mit mehr Weiß- oder Schwarzanteil. Summiert man die genannten Zahlen, so umfasst das Farbrad eine Menge von 120 Farben, wobei jede Einzelne einen Namen basierend auf einem, in der realen Welt existierenden Beispiel erhalten hat. Die einzelnen Farbbereiche sind in ihrer Helligkeit, Farbverzerrung der Primärfarben und dem Abney Effect korrigiert. Der mittlere Ring beschreibt die Repräsentative des Farbtons, welcher in seiner Sättigung und Helligkeit unverändert ist. Geht man von dem mittleren Ring zum Zentrum, so findet man dort die Shades des Farbtons,

welche im V-Wert absteigend sind. Betrachtet man wiederum die Farben am Rand des Farbkreises so haben diese einen geringeren Sättigungswert.

5.4 Das Vier-Farbtyp-System

Im gewählten Beispiel sind Haar- und Augenfarbe, den Hautton und weiterer Spezifikationen essentielle Rahmenbedingungen. Diese werden auch als *Typ* wahrgenommen. Dieser Typ wird hauptsächlich in der Farbberatung benutzt, um eine Zuordnung von Mengen an Farben für Kleidung, Make-up und Wohnungsgestaltung zusammenzustellen. Grundsätzlich gibt es vier Farbtypen, welche in kalte und warme Typen eingeteilt werden können. Unter den *kalten Farbtypen* befindet sich der Sommer- und der Wintertyp. Farbpaletten für diesen Typ fehlt der gelbe Unterton und einige, aber keineswegs alle, besitzen einen blauen Unterton.

Betrachtet man wiederum die *warmen Farbtypen*, so zählt man die Farbtypen des Frühlings und des Herbsts dazu. Die Farbtabellen dieser Typen besitzen einen starken gelben Unterton. Im Laufe der Zeit haben sich, aufbauend auf dem Vier-Farbtyp-System, weitere Systeme entwickelt, welche die einzelnen Typen weiter voneinander trennen durch Mischtypen.

5.5 Charakteristika der Farbtypen

Die unten aufgeführten Charakteristika sind nur grobe Maßstäbe für die Identifikation des Farbtyps und garantieren keinen einhundert prozentigen Erfolg. Nach Karin Kunkel besteht nur eine 80 prozentige Wahrscheinlichkeit und die Beschreibungen beziehen sich auf „[...]-im Erwachsenenalter befindliche- mitteleuropäische Bevölkerung.” (Hunkel, 2011, 254).

Frühlingstyp

Der cremefarbene- bis goldene Hautton ist charakteristisch für den Frühlingstyp. Ihre Augen bieten die größte farbliche Bandbreite unter den Farbtypen. Diese können goldgrün, bernsteinfarben sein oder ein warmes blau oder braun besitzen. Ihre Haare zeichnen sich durch einen warmen Grundton mit einen goldenen Glanz aus. Frühlingstypen sind in den Wintermonaten meist blass und die Haut bräunt meist schnell (Hunkel, 2011, 266-268).

Sommertyp

Der Sommertyp hat einen rosigen Farbton und errötet leicht. Ihre Augen können Farben wie ein klares blau, aquamarin, haselnuss-oder dunkelbraun, blaugrün oder blaugrau besitzen. Die meist verbreiteste Haarfarbe innerhalb des Typs sind aschblond oder rotbraun (Hunkel, 2011, 285-287).

Herbsttyp

Der Herbsttyp besitzt auch eine helle Haut, welche einem Rosa ähnelt. Menschen dieses Typs haben meist Sommersprossen und bräunen kaum. Ihre Augen können ein goldenes Bernstein, Haselnussbraun, helles oder dunkles braun, Rotbraun oder ein Braun, das fast schwarz wirkt, sein. Einige Menschen des Typs besitzen auch grüne Auge in Richtung eines oliv- oder avocado grüns. Ihre Haare sind meist feuerrot, welches oft einen aschton besitzt. In Ausnahmefällen ist ihre natürliche Haarfarbe ein dunkelbraun (Hunkel, 2011, 273-275).

Wintertyp

Betrachtet man den Wintertyp, so besitzt er eine helle bis dunkle olive Haut, welche schnell braun wird. Ihre Bandbreite der Augenfarbe besitzt ein gold- haselnuss-, dunkel- und schwarzbraun und ein klares- oder dunkles Blau bis zu einem Indigo-Ton (Hunkel, 2011, 293-295).

5.6. Integration und Aufbereitung der Farbdaten

Bei der Definition des Datenmodells müssen die zuvor getroffenen Entscheidung für technische Plattformen (vgl. 4.1 Definition der Plattform) mit den fachlichen Anforderungen (vgl. 5.1 Business-Konzept) in Einklang gebracht werden.

Zuerst wurden die Farbdaten für das gewählte Datenformat (vgl. 4.3 JSON) aufbereitet. Hierfür wurden, zusätzlich zu den Farben des Martian Color Wheels, die *achromatischen Farben* (vgl. 9.1 Begriffsdefinitionen) - Schwarz, Weiß, Grau, Dunkel- und Hellgrau betrachtet. Das Hinzuziehen der achromatischen Farben ist begründet in der nicht Vorhandenheit im Martian Color Wheel und diese oft in der Garderobe zu finden sind.

Herausforderung im Datenmodell ist in diesem Beispiel die Integration von den achromatischen Farben in den Farbraum des Martian-Color-Wheels (vgl. 2.8.3 HSV-Farbraum). A*chromatische Farben* sind linear als Farbobjekte definiert, welche einen Namen, den HSV-Wert und den HEX-Wert als *String* realisieren. Definitionen des *Martian-Color-Modells* besitzen Objekte, welche die gesamte Farbfamilie darstellen. Diese beinhalten neben dem Namen der Farbfamilie und einer Kategorisierung auch Tuple aus Farbobjekten für die repräsentative Farbe, sowie die Tints und Shades der Farbfamilie. Quellcode-*Beispiel 2* zeigt dies an dem Beispiel von der Farbfamilie von Rot.

```
{
	"identifier": "Additive Primary Color",
	"colorFamily": "Red",
	"representative": {
		"name": "Red",
		"hsv": "hsv(0, 100%, 100%)",
		"hex": "#ff0000"
	},
	"shades": [...],
	"tints": [...]
}
```

Quellcode-Beispiel 2 - Farbobjekt Rot

Der Nutzen *des Farbtyps* (vgl. 5.4 Das Vier-Farbtyp-System) sind zugehörigen Farbpaletten, welche zu den äußerlichen Merkmalen des Anwenders passen. Die dafür benötigten Daten wurden vollumfänglich in einer Gesamt-JSON-Struktur definiert. Die Struktur besitzt eine Menge aus Objekten, welche jeweils eine Id, einen Namen und eine Menge aus Farben besitzen, welche zu dem jeweiligen Typ passen.

5.7 Datenaufbereitung am Beispiel von Farbkonvertierung

Fachliche Anforderungen passen nicht immer zu den Schnittstellen der verwendeten technischen Plattformen und müssen für die konkrete Umgebung angepasst werden. Im gewählten Beispiel trifft dies beispielsweise auf das Darstellungsformat der Farben zu.

Durch die Farbbeschreibung von Warren Mars sind die HSV-Werte der Farbtöne gegeben, jedoch unterstützt React-Native dieses Darstellungsformat nicht (FaceBook Inc., 2020). Aus diesem Grund wurde mit Hilfe von TinyColor (vgl. 4.2.1 TinyColor) die HSV-Werte in Hex-Werte formatiert. *Quellcode-Beispiel 3* zeigt die Konvertierung einer Farbe von dem RGB-Format zum Hex-Format.

```
function rgbToHex(r, g, b, allow3Char) {
   var hex = [
       pad2(mathRound(r).toString(16)),
       pad2(mathRound(g).toString(16)),
       pad2(mathRound(b).toString(16))
   ];
   // Return a 3 character hex if possible
   if (allow3Char && hex[0].charAt(0) == hex[0].charAt(1) &&
                     hex[1].charAt(0) == hex[1].charAt(1) &&
                     hex[2].charAt(0) == hex[2].charAt(1)) {
      return hex[0].charAt(0) + hex[1].charAt(0) + hex[2].charAt(0);
   }

   return hex.join("");
}
```

Quellcode-Beispiel 3 - TinyColor, RGB zu Hex

6. Entwicklungsprozess

Bei der Implementierung der Anwendung wird zunächst das Architekturmodell mit fachlichen Abhängigkeiten definiert. Die in *Abb. 6* dargestellte Klassenhierarchie dient zur Veranschaulichung eines beispielhaften Architekturmodells im Rahmen der Realisierung des Konzepts. In folgenden Abschnitt werden die einzelnen Klassen vorgestellt und in ihren Einsatzgebiet und Nutzen beschrieben.

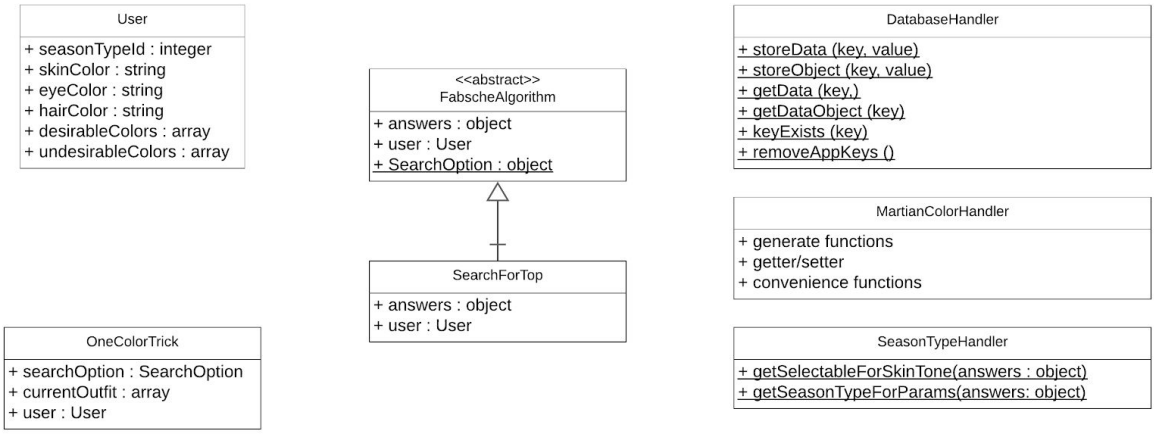


Abb. 6 - Beispielhafte Architektur

6.1 DatabaseHandler

Die Klasse *DatabaseHandler* ist die Schnittstelle für das Arbeiten mit dem Framework AsyncStorage (vgl. 4.2.3 Async Storage). Es besitzt Hilfsfunktionen, um Daten beziehungsweise Datenobjekte zu speichern und auszulesen. Des weiteren können Datensätze auf ihre Existenz überprüft werden und gegebenenfalls gelöscht werden.

6.2 MartianColorHandler

Der *MartianColorHandler* dient grundlegend dazu, um mit den eingespeisten Farbobjekten zu arbeiten. Innerhalb der Klasse werden die fachlichen Anforderungen mit den Möglichkeiten der gewählten Plattform realisiert (vgl. 5.2 Fachliche Grundlagen des Business-Konzepts)

Essentielle Funktion ist *findColorByProperty*, welche die Farbe anhand des übergebenen *property*- und *value* Parameters zurückgibt. Zusätzlich können durch die Hilfsfunktionen *isDarkColor* und *isLightColor* die Farben nach ihrer Helligkeit unterschieden werden. Des weiteren gibt es Funktionen, um die warmen- und kalten Farben zu identifizieren, Farben anhand ihrer Bezeichnung zu filtern, den Farbwert, die Sättigung oder die Helligkeit der Farbe zu extrahieren und Farbobjekte zu vergleichen.

Hauptaufgabe der Klasse ist die Implementierung der Farbschemata. Nachfolgend wird die beispielhafte Realisierung des Komplementär Kontrastes betrachte. Laut der Definition liegt die Komplementärfarbe auf der gegenüberliegenden Seite des Farbkreises (vgl. Abb. 2).

Vorbedingung: $H \in [0°, 360°]$

$$(H2) := \begin{cases} H + 180^\circ & \text{, falls } H + 180^\circ \leq 360^\circ \\ H - 180^\circ & \text{, falls } H + 180^\circ > 360^\circ \end{cases}$$

Nachbedingung: $H2 \in [0^\circ, 360^\circ]$

Die Formel lässt sich jedoch nicht unmittelbar auf das Martian-Color-Wheel anwenden, da die H-Werte nicht gleichmäßig auf dem Farbrad verteilt sind. Diese Farbtonverschiebung hat ihren Ursprung in der Korrigierung der Farbverzerrung. Aus diesem Grund wurde eine Lösung realisiert, welche auf den Indizes der Farbtöne arbeitet. Folgende Formel ergibt sich aus dem Ansatz, mit der Vorbedingung, dass der H-Wert in der Menge *hues* existiert. Die Menge *hues* inkludiert alle H-Werte der Farbtöne (innerster Ring).

Vorbedingung: $I \in [0, 23]$

$$(I2) := \begin{cases} I + 12 & \text{, falls } I + 12 \leq 23 \\ I - 12 & \text{, falls } I + 12 > 23 \end{cases}$$

Nachbedingung: $I2 \in [0, 23]$

6.3 Programmatischer Lösungsansatz für die Farbtonverschiebung

Der Lösungsansatz für die Farbtonverschiebung findet ihren Einsatz in den Methoden für die Anwendung der Farbschemata.

Quellcode-Beispiel 4 zeigt eine beispielhafte Realisierung des Komplementärkontrastes (vgl. 2.6.2 Komplementäres Farbschema). Zuerst wird die derzeitige Position/Index der übergebenen Farbe identifiziert. Darauffolgend wird das Farbobjekt der Komplementärfarbe herausgefiltert, um im letzten Schritt, die passende Farbe zu finden.

```
getComplementColor(color) {
    let complement = undefined;
    let wheelPosition = this.getColorWheelPosition(color);
    let complementColorObj = this.getNextColorObjBySteps(wheelPosition, 12);

    if (typeof complementColorObj !== "undefined") {
        complement = this.getMatchingColorInColorObj(color, complementColorObj)
    }

    return complement;
}
```

Quellcode-Beispiel 4 - Komplementärkontrast

6.3 SeasonTypeHandler

Im Rahmen der Identifizierung des Typs werden dem Anwender Fragen zu äußerlichen Merkmalen und Präferenzen gestellt. Der *SeasonTypeHandler* ermittelt anhand der gegebenen Daten den Typ des Anwenders. Grundlage ist der von Zalando implementierte Algorithmus, welcher auf der Theorie des *Vier-Farbtyp-Systems* basiert. Die Grundlagenentscheidung ist begründet in einer anfänglichen Analyse der gewählten Parameter und deren Gewichtung. Zusätzlich wurde die Veränderung der Endergebnismenge bei unterschiedlichen Eingabereihenfolgen betrachtet.

In dem Ansatz der Realisierung hält sich die Fragestellung und Benutzerführung an Zalandos entwickelten Algorithmus (vgl. Tab. 2). Die Entscheidung des Typs ist grundlegend abhängig von der Antwort der letzten Frage. Die Antwortmöglichkeiten dieser variieren durch die Eingabe von spezielle Kombinationen zwischen Haar- und Augenfarbe (vgl. Tab. 3)

6.4 User

Objekte der Klasse *User* halten die repräsentativen Nutzerdaten. Sie beinhalten die äußerlichen Merkmale des Benutzers, die präferierten und nicht präferierten Farben als auch den Typ des Anwenders (vgl. Quellcode-Beispiel 5).

```
constructor(seasonTypeId, skinColor, eyeColor, hairColor, desirableColors,
            undesirableColors) {
   this.skinColor = skinColor;
   this.eyeColor = eyeColor;
   this.hairColor = hairColor;
   this.desirableColors = [].concat(desirableColors);
   this.undesirableColors = [].concat(undesirableColors);
   this._assignSeasonType(seasonTypeId)
   this._finalizeInitialization();
}
```

Quellcode-Beispiel 5 - Konstruktor des Benutzerobjekts

6.5 Implementierung des One-Color-Trick

In diesem Abschnitt wird mit Hilfe von *Abb. 7* ein beispielhafter Prozess für die Realisierung der Business-Idee, *One-Color-Trick* (vgl. 5.1.1 One-Color-Trick), aufgezeigt.

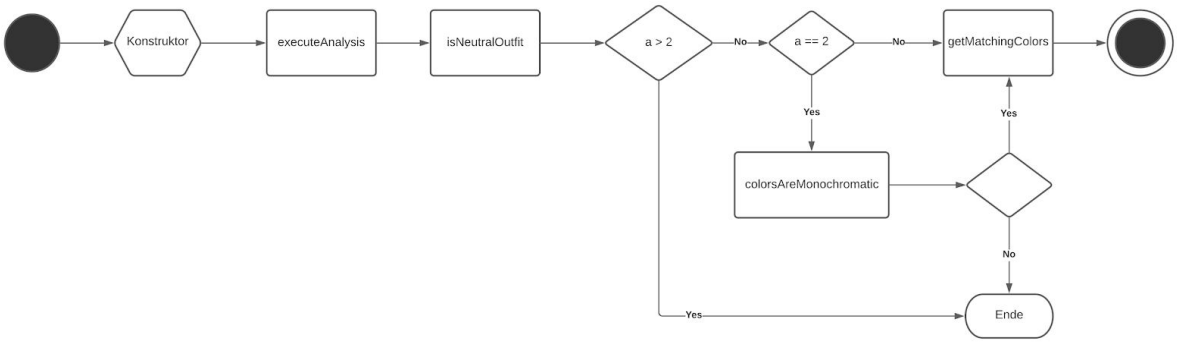


Abb. 7 - Prozessdiagramm von OneColorTrick

Während der Datenvorbereitung des Prozesses werden die Suchoption, die Farben des derzeitigen Garderobe und das Benutzerobjekt erzeugt. Die Suchoption dient lediglich zu Fallunterscheidungen bei den einzelnen Prozessschritten.

Der Prozess *executeAnalysis* umfasst die Voranalyse des derzeitigen Outfits. Ziel ist es die Farben des Outfits in nicht neutrale- und neutrale Farben zu unterteilen und diese für Folgeschritte aufzubereiten.

Innerhalb des Prozesses *isNeutralOutfit* wird anhand der Analyseergebnisse entschieden, ob der Algorithmus angewendet werden kann. Diese Bedingungen beziehen sich auf die Business-Idee des One-Color-Tricks.

Beispielhaft wird folglich der Fall betrachtet, dass das Outfit aus zwei nicht neutralen Farben besteht. Der Prozessschritt *colorsAreMonochromatic* überprüft, ob die Farben aus einer Farbfamilie abstammen. Anhand des Ergebnisses wird der Fortgang des Prozesses entschieden.

Kann der *One-Color-Trick* angewendet werden, so wird der Prozessschritt *getMatchingColors* angestoßen. Dessen Aufgabe ist es, eine Menge aus farblich passenden Farben zusammenzustellen. Grundlegend werden alle neutralen Farben der Ergebnismenge hinzugefügt, falls diese nicht von dem Anwender benachteiligt werden.

Das Ergebnis aus *analysisOutfit* nimmt Einfluss auf die Ergebnismenge von *getMatchingColor*. Grundlegend werden alle monochromatischen Farben der Ergebnismenge hinzugefügt, falls der Anwender eine nicht neutrale Farbe trägt. Trägt der Benutzer jedoch nur neutrale Farben, so kann mehr auf die Präferenzen eingegangen werden. Hierbei werden präferierte Farben, die nicht präferierten Farben und die Farbpalette des Typs betrachtet. Am Ende des Prozessschrittes wird sichergestellt, dass die Ergebnismenge keine Duplikate besitzt und die Reihenfolge der Sortierung beibehalten bleibt.

Das Ende des Prozesses beschreibt die Bereitstellung der zusammengestellten Farben für aufrufende Prozesse.

6.6 FabscheAlgorithm

Allgemeingültige Vorgaben und Verhaltensweisen sind innerhalb der Realisierung festgeschrieben. Diese Klasse ist eine Blaupause für erbende Klassen, die einen Suchalgorithmus für Kleidungsstücke anhand der Business-Idee *FabscheAlgorithmus* realisieren (vgl. 5.1.2 FabscheAlgorithmus).

Innerhalb der Realisierung wird zum Beispiel festgeschrieben, dass jede Subklasse die Funktion *getOutfitColors* definiert, um die Farben des derzeitigen Outfits zurückzugeben.

6.7 SearchForTop

Der folgende Abschnitt beschreibt eine beispielhafte Realisierung eines Suchalgorithmus nach der Idee des *FabscheAlgorithmus*. Die Klasse realisiert den in *Abb. 8* beschriebenen Prozess für das Suchen nach einem farblich passenden Oberteil.

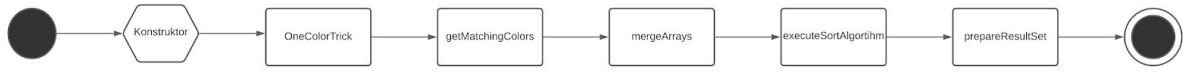


Abb. 8 - Prozessdiagramm von SearchForTop

Bevor der Prozess angestoßen wird bekommt der Anwender gezielte Fragen gestellt. Es gilt die Präferenzen des Anwenders und die selektierten Farben der Garderobe herauszufinden. Zusätzlich zu den Antworten braucht die Vorbereitung das Benutzerobjekt des Anwenders. Die Vorbereitung ist abgeschlossen nach dem Sammeln und Aufbereiten der geforderten Daten.

Zu Beginn wird der Prozess *OneColorTrick* (vgl. 6.5 Implementierung des One-Color-Trick) ausgeführt. Ziel des Prozessschrittes *getMatchingColors* ist das Sammeln von Farben. Grundlegend werden entweder alle hellen Farben oder alle dunklen Farben des Datenbestands herausgefiltert. Die Entscheidung wird anhand der präferierten Helligkeit, der Jackenfarbe oder der Hautfarbe entschieden. Fallunterscheidungen sind in der Business-Idee des *FabscheAlgorithmus* (vgl. 5.1.2 FabscheAlgorithmus) aufgeführt. Um eine beispielhafte Realisierung für das Selektieren der hellen Farben aufzuzeigen wird das Quellcode-Beispiel 6 beschrieben.

Im ersten Schritt werden die Funktionen *getTints* und *getLightAchromatics* aufgerufen, um alle hellen Farben des Datenbestands herauszufiltern. Die Methode *getTints* selektiert die repräsentative Farben und die Tints des Martian-Color-Wheels. Die zwei Ergebnismengen werden konkateniert und im nächsten Schritt gefiltert. Bei jeder Farbe wird überprüft, ob der Nutzer diese benachteiligt, die Farbe noch nicht selektiert wurde und die Farbe nicht der Hose

gleicht. Zusätzlich wird die Gleichheit mit der Jackenfarbe überprüft, falls der Anwender eine Jacke trägt.

```
getLightColorsForUser(wearsJacket) {
    let matchingColors = [];
    let tints = this.martianColorHandler.getTints(null, true,  "hex");
    let lightAchromaticColors = this.martianColorHandler
                                    .getLightAchromaticColors("hex");
    tints.concat(lightAchromaticColors).forEach(color => {
        if(!this.user.postponesColor(color)) {
            if(!HelperTool.isInArray(color, matchingColors) &&
                color !== this.answers.trousersColor) {
                    if(wearsJacket && color !== this.answers.jacketColor) {
                                matchingColors.push(color);
                    } else { matchingColors.push(color); }}}})

    return matchingColors;
}
```

Quellcode-Beispiel 6: Selektion aller hellen Farben

Der darauffolgende Prozessschritt *mergeArray* konkateniert die Ergebnismenge von den Prozessschritten *OneColorTrick* und *getMatchingColor*. Hierbei werden Duplikate exkludiert und darauf geachtet, dass die Reihenfolge der Elemente beibehalten wird.

Sortieralgorithmus

Im Rahmen von *executeSortAlgortihm* wird die konkatenierte Ergebnismenge nach verschiedenen Parametern sortiert. Die JavaScript-Funktion *Array.prototype.sort()* wandelt die einzelnen Elemente der Ergebnismenge in String und sortiert diese dann anhand ihrer UTF-16 Codepoints. Es gibt jedoch die Möglichkeit eine eigene Sort-Funktion zu schreiben, indem man als Parameter eine Funktion übergibt. Für das Erstellen einer beispielhaften Sortierfunktion beziehe ich mich folglich auf das Quellcode-Beispiel 7.

```
function sortAlgorithm(array) {
  let sortedArray = [];

  if(array.length > 0) {
      sortedArray = array.sort(function(a, b) {
          return a - b;
      })}

  return sortedArray;
}
```

Quellcode-Beispiel 7: Custom Sortierfunktion

Die Funktion benötigt zwei Parameter, welche jeweils ein Element repräsentieren. Innerhalb der Funktion wird dann an gewünschten Kriterien überprüft, wie sich die Reihenfolge der zu vergleichenden Elemente ändert. Besitzt das Endergebnis einen negative Wert so steht Element *a* vor Element b. Bei einem positiven Ergebnis trifft das Entgegengesetzt zu. Die Reihenfolge der Elemente bleibt erhalten, falls das Endergebnis einen Wert von 0 besitzt.

Im Rahmen der Realisierung des maßgeschneiderten Sortieralgorithmus werden die in Tab. 1 dargestellten Parameter betrachtet. Diese sind nach ihrer Gewichtung absteigend sortiert und werden in dem Prozess *executeSortAlgorithm* chronologisch überprüft. Die aktuelle Gewichtung der Parameter ist durch den Einfluss der Theorie und durch die Erkenntnisse der *Testphase* (vgl. 7.2 Testablauf) resultiert.

7. Realisierung des Testmanagements

Neben der Entwicklung der Anwendung ist das Etablieren eines effizienten Testmanagements essentiell. Am Anfang jeder Entwicklung sollte fest definiert werden, welche Testarten eingesetzt werden. Dieser Abschnitt behandelt verschiedene Testvarianten und deren Integration in den Softwareentwicklungsprozess dieser Anwendung.

7.1 Testvarianten

7.1.1 Applikationsbezogen

Applikationstests dienen dem Zweck die geschriebene Anwendung auf funktionaler Ebene zu testen. Anhand von User Stories und daraus resultierenden automatisierten Tests können beispielsweise reale Interaktionsverläufe getestet werden. *Functional Unit Test*, *Smoke Test*, *Integrationstests* sind einige Beispiele der applikationsbezogenen Tests.

7.1.2 Infrastrukturbezogen

Die Infrastruktur einer Anwendung hat einen großen Einfluss auf das Anwender-Erlebnis (engl. User Experience, UX). Es wird unter anderem betrachtet wie sich die Applikation im Umfeld einer hohen Nutzlast verhält. Einige Beispiele für infrastrukturbezogene Tests sind *Performance/Last-/Massentest*, *Failover Test*, *Spezialtest* und *Penetration Test*

7.1.3 Nutzerbezogen

In *nutzerbezogenen Tests* geht es um das manuelle Testen der Anwendung durch potentielle Anwender. Ein großes Problem der Entwicklung ist der eingeschränkte Blick wie die Anwendung am Ende tatsächlich benutzt wird. Durch die Integration von realen Nutzern/Testern können Intuitivität, Haptik, Benutzungsgefühl und Logikfehler der Anwendung identifiziert werden.

7.2 Testablauf

Für den Rahmen der Bachelorarbeit wurde ausschließlich die nutzerbezogenen Tests durchgeführt, da diese für die Realisierung der Business-Idee die wichtigsten Rolle spielen (vgl. 5.1 Business-Konzept). Die Ergebnisse der Tests unterstützten die Entwicklung des Sortieralgorithmus und verbesserten die Haptik und die User-Erfahrung. Zusätzlich konnten durch das Einbeziehen der Anwender/Tester Logikfehler und Designfehler identifiziert werden, die auf Grund fehlender Unit-Test zunächst unbemerkt blieben.

Zum Start der einzelnen Testzyklen bekamen die Probanden die jeweils neueste Version der Applikation auf ihre mobilen Geräte installiert. Im ersten Schritt wurde überprüft wie intuitiv die Anwendung ist. Die Probanden wurden gebeten Unklarheiten und Anpassungsvorschläge zu erwähnen, welche in Testprotokollen erfasst wurden. Im Laufe der Tests ergaben sich hierdurch Design- und Funktionsvorschläge. Zusammen mit den einzelnen Teilnehmern wurden mehrere Testdurchläufe durchgeführt und die Gewichtung der Sortierparameter angepasst.

7.3 Mehrwert für das Endprodukt

Das Einbeziehen der potentiellen Anwender/Tester war im Rahmen der Realisierung essentiell und hatte für das Endprodukt einen spürbaren Mehrwert. Es konnten Fehler identifiziert werden, da sich das Nutzungsverhalten der Endanwender von dem der Entwickler unterscheidet. Zusätzlich wurden die Wünsche berücksichtigt und resultierend eine Anwendung geschaffen, welche im Allgemeinen Anklang findet.

8. Fazit

Der folgende Abschnitt diskutiert in wie fern die eingangs gestellte Zielsetzung erfüllt werden konnte. Es werden der derzeitige Stand des Konzepts, sowie die Anwendung und während der Implementierung aufgetretene, als auch zukünftige Herausforderungen betrachtet. Zusätzlich wird eine Zukunftsvision für das Konzept beschrieben, welche mögliche Integrationen und Weiterentwicklungen aufführt.

8.1 Ist-Stand

Der aktuelle Stand des Konzepts deckt alle definierten fachlichen Anforderungen ab (vgl. 5. Fachliche Anforderungen des Anwendungskonzeptes). Durch die Realisierung konnte das Konzept erprobt und verbessert werden. Zusätzlich unterlief es einer Testphase in welcher, dank der Hilfe von Probanden, die Business-Idee getestet und optimiert werden konnte.

8.2 Herausforderungen und Ausblick

In der Theorie erstellte Vorgehensweisen können oft nicht vollumfänglich praktisch umgesetzt werden. Mit der Auswahl der Plattform und der verwendeten Frameworks (vgl. 4.2 Frameworkauswahl) schränken sich die Möglichkeiten der Umsetzung ein. Es gilt Lösungen zu finden, um die daraus resultierenden Herausforderungen zu meistern. Im Rahmen der Realisierung kam es sowohl zu Problemen in der Datendarstellung (vgl. 5.6. Integration und Aufbereitung der Farbdaten) als auch zu Logikfehlern in den einzelnen Prozessschritten beziehungsweise im Konzept selbst.

In der Welt der Mode gibt es keine scharf definierten Regeln an die man sich halten kann. Es gibt lediglich theoretische Ansätze und Vorschläge für die Anwendung von Farbschemata. Zusätzlich sind die Geschmäcker jedes Einzelnen unterschiedlich, wodurch eine allgemeingültige Lösung schwer zu entwickeln ist. Durch die Integration der Testprobanden konnten deren Vorschläge zwar berücksichtigt und umgesetzt werden, jedoch decken diese nicht die gesamte Masse an potentiellen Anwendern ab. Ein Lösungsansatz hierfür wäre das Sammeln von Anwender-Präferenzen. Es gilt so viele Informationen wie möglich von jedem

Anwender zu sammeln, um eine maßgeschneidertes Ergebnis in Form eines Vorschlages anbieten zu können. Bei einer Weiterentwicklung müssten jedoch noch viele weitere Umfragen und Tests konzipiert und durchgeführt werden. Zusätzlich sollten renommierte Mode- und Typberater in dem Entwicklungsprozess integriert werden.

8.3 Zukunftsvision

Ich möchte das in dieser Bachelorarbeit entwickelte Konzept weiter ausbauen und mit weiteren mobilen Anwendungen verknüpfen. Der Algorithmus kann sowohl als Fundament darauf aufbauender Anwendungen, als auch zur Erweiterung bestehender Implementierungen verwendet werden. Es könnte beispielsweise als Erweiterung in die Algorithmen von Zalando (Zalon) und Outfittery (vgl. 1.2.1 Outfittery und Zalon) integriert werden. Durch die differenzierte Betrachtung der Farbwünsche der Kunden könnte hierdurch ein Mehrwert für diese Anwendungen entstehen.

Die im Rahmen der Bachelorarbeit entstandene Anwendung soll zukünftig verbessert und erweitert werden. Grundidee ist es, dass Anwender ihren Kleiderschrank in die mobile Applikation einpflegen. Anknüpfend daran kann das entwickelte Konzept benutzt werden, um farblich passende Garderoben beziehungsweise Kleidungsstücke aus den eingepflegten Kleiderschränken vorzuschlagen.

Parallel dazu existiert der Plan eine Community zu schaffen. Innerhalb dieser können dann Teile oder auch der gesamte Kleiderschrank mit seinen Freunden und Familienmitgliedern geteilt werden. Denkbar ist auch eine erweiterte Suche für den konzipierten Algorithmus (vgl. 5.1.2 FabscheAlgorithmus). Zusätzlich sind Konzepte wie Ebay und Verleihdienstleistungen möglich oder vergleichbare Algorithmen wie *Shop-The-Look* (Zalando).

9. Glossar

9.1 Begriffsdefinitionen

Abney Effect	Der Abney Effect beschreibt die Farbtonverschiebung einer monochromatischen Farbe durch Hinzufügen von weißen Licht, wodurch die Sättigung der Farbe verändert wird.
Tints	Tints sind Farbabstufungen eines Farbtons durch Hinzufügen von Weiß.
Shades	Shades sind Farbabstufungen eines Farbtons durch Hinzufügen von Schwarz.
Sekundärfarbe	Sekundärfarben sind Farben, welche durch Mischen von zwei Primärfarben entstehen.
Tertiärfarbe	Tertiärfarben sind Farben, welche durch Mischen von zwei Sekundärfarben entstehen.
Achromatische Farben	Achromatische Farben, auch unbunte Farben genannt, sind Farben wie Schwarz, Weiß und die dazwischenliegenden Graustufen.
Neutrale Farben	Neutrale Farben sind achromatische Farben und dezente Farben wie Beige, ein helles Gelb, Khaki, Navi.
Farbkonvertierung	Eine Farbkonvertierung ist die Konvertierung des Darstellungsformat einer Farbe zu einer einem anderen Format (vgl. 9.6 Umrechnung von HSV zu RGB Werten).
Corporate-Identity	Corporate-Identity beschreibt die Gesamtheit der Merkmale eines Unternehmens. Diese sind kenzeichnende- oder zu anderen Unternehmen unterscheidende Merkmale.

Android Package (APK)	APK, auch Android Package, ist ein Paketdateiformat, das vom Android-Betriebssystem oder auf Android basierenden Betriebssystem genutzt wird, um Applikationen zu installieren.
Neutralgrau	Eine Farbe ist Neutralgrau, falls sie alle Farben des Spektrums im gleichen Anteil enthält. Im RGB-Farbraum ist beispielsweise folgende Farbe Neutralgrau → Rot: 46, Grün: 46, Blau: 46
Sukzessivkontrast	Sind Kontraste, welche nach Wegnahme des Sinnesreizes, durch bspw. das Schließen der Augen, ein Nachbild hervorrufen.

9.2 Umrechnung von HSV zu RGB Werten

Vorbedingung: $H \in [0°, 360°]$, $S, V \in [0, 1]$

$$h_i := \left\lfloor \frac{H}{60°} \right\rfloor; \quad f := \left(\frac{H}{60°} - h_i \right)$$

$$p := V \cdot (1 - S); \quad q := V \cdot (1 - S \cdot f); \quad t := V \cdot (1 - S \cdot (1 - f))$$

$$(R, G, B) := \begin{cases} (V, t, p), & \text{falls } h_i \in \{0, 6\} \\ (q, V, p), & \text{falls } h_i = 1 \\ (p, V, t), & \text{falls } h_i = 2 \\ (p, q, V), & \text{falls } h_i = 3 \\ (t, p, V), & \text{falls } h_i = 4 \\ (V, p, q), & \text{falls } h_i = 5 \end{cases}$$

Nachbedingung: $R, G, B \in [0, 1]$

Falls der S-Wert den Wert 0 besitzt, so kann die Formel wie folgt vereinfacht werden: $R = G = B = V$ *(Wikipedia, 2020)*.

9.3 Beispielhafte Parameter für die Suche nach Oberteilen

Parameter	Erklärung
Farbe gleicht der Hosenfarbe	In diesem Schritt wird überprüft, ob die Farben der Hosenfarbe gleicht.
Farbe gleicht der Schuhfarbe	Es wird überprüft, ob die Farben der Schuhfarbe gleichen.
Sortierung nach der Helligkeit	Die Farben werden anhand der Präferenz der Helligkeit sortiert.
Sortierung nach gewünschten Farben	Es wird überprüft, ob die Farben präferiert beziehungsweise nicht präferiert werden.
Sortierung nach neutralen Farben	Es wird überprüft, ob die Farben neutral sind.
Sortierung nach Farbpaletten des Farbtyps	Es wird überprüft, ob die Farben Bestandteil der Farbpalette des zugewiesenen Farbtyps sind.
Sortierung nach der Hosenfarbe	Es wird anhand der Hosenfarbe sortiert (vgl. 5.1.2 FabscheAlgorithmus).
Sortierung nach der Schuhfarbe	Es wird anhand der Schuhfarbe sortiert (vgl. 5.1.2 FabscheAlgorithmus).
Sortierung nach der Jackenfarbe	Es wird anhand der JackenFarbe sortiert (vgl. 5.1.2 FabscheAlgorithmus).

Tab. 1 - Beispielhafte Parameter für die Suche nach Oberteilen

9.4 Beispielhafte Fragen für die Farbtypidentifikation

Nr.	Frage	Antworten
1	Selektiere deine Augenfarbe	Blau, Grün, Braun
2	Selektiere den *-Ton deiner Augenfarbe	Blau: 1. Reines Blau 2. Blaugrün 3. Blaugrau Grün: 1. Reines Grün

		2. Grünbraun 3. Grüngrau Braun: 1. Reines Braun 2. Braungrün 3. Schwarzbraun
3	Selektiere deine Haarfarbe	Blond, Rot, Braun, Schwarz
4	Selektiere den *-Ton deiner Haare	Blond: 1. Hellblond 2. Aschblond 3. Goldblond 4. Rotblond Rot: 1. Reines Rot 2. Rotblond 3. Kupfer Braun: 1. Goldbraun 2. Hellbraun 3. Rotbraun 4. Schwarzbraun Schwarz: 1. Schwarz 2. Blauschwarz 3. Braunschwarz
5	Selektiere deine Hautfarbe	eher Rosig, eher Beige
6	Selektiere deine Hautfarbe (Antwortmöglichkeiten unterscheiden sich bei Sonderfällen in Kombination von Haar- und Augenfarbe (vgl.9.8 Farbtypberatung Sonderfälle))	eher Rosig: 1. Haut wird braun mit Sonnenbrand 2. Haut wird braun ohne Sonnenbrand eher Beige: 1. Schnell braun, mit goldgelber Bräunung 2. Bräunt fast nicht, neigt zu Sommersprossen

Tab. 2 - Zalandos Farbberatung (Zalando SE, 2016)

9.5 Farbtypberatung Sonderfälle

Antwort: Augenfarbe	Antwort: Haarfarbe	Antwortmöglichkeiten bei der letzten Frage
Reines Blau, Blaugrau, Grüngrau, Scharzbraun	Hellblond, Aschblond, Hellbraun, Schwarzbraun, Schwarz, Blauschwarz	1. Haut wird braun mit Sonnenbrand 2. Haut wird braun ohne Sonnenbrand
Blaugrün, Reines Grün, Grünbraun, Reines Braun, Braungrün	Goldblond, Rotblond, Reines Rot, Rotblond, Kupfer, Goldbraun, Rotbraun, Braunschwarz	1. Schnell braun mit goldgelber Bräunung 2. Schnell braun ohne goldgelbe Bräunung

Tab. 3 - Sonderfälle bei Zalandos Farbberatung (Zalando SE, 2016)

Literaturverzeichnis

Adobe Inc. (2019, 10 30). *Farbrad, ein Farbpaletten-Generator*. Farbrad, ein Farbpaletten-Generator. Zugriff 11 16, 2020, von https://color.adobe.com/de/create/color-wheel

Baumgart, G., Müller, A., & Zeugner, G. (1996). *Farbgestaltung: Baudekor, Schrift, Zeichnen* (1st ed.). Cornelsen Verlag. 3-464-43401-X

Bianchi, F. (2014, 10). *Coolors - The super fast color schemes generator!* Coolors - The super fast color schemes generator! Zugriff 11 16, 2020, von https://coolors.co

Bibliographisches Institut AG. (1973). *Meyers Enzyklopädisches Lexikon* (9th ed., Vol. 8). Bibliographisches Institut AG.

BR Fernsehen. (2020, 02 01). *Farbenlehre*. Farbenlehre. Wie Farben im Krankenhaus wirken. Zugriff 12 09, 2020, von https://www.br.de/mediathek/video/farbenlehre-wie-farben-im-krankenhaus-wirken-av:5e344091f6e8060012b40767

Coleman, A. M. (2009). *A Dictionary of Psychology* (3rd ed.). Oxford University Press. 9780199534067

Cranston, B. (2014, 07 06). *TinyColor*. TinyColor. Fast, small color manipulation and conversion for JavaScript. Zugriff 12 09, 2020, von https://bgrins.github.io/TinyColor/

Creative Ventures Inc. (2013, 03 18). *How to Know What Colors Work (Men's Guide to T-Shirts)*. How to Know What Colors Work (Men's Guide to T-Shirts) | One Hour Tees. Zugriff 12 12, 2020, von https://www.onehourtees.com/blog/how-to-know-what-colors-work/

Ecma International. (2017, 12). *Standard ECMA-404*. Standard ECMA-404. Zugriff 12 16, 2020, von https://www.ecma-international.org/publications/standards/Ecma-404.htm

Eichner, H. (2015, 05 05). *206 - Wahrnehmungsorientierte Farbmodelle*. 206 - Wahrnehmungsorientierte Farbmodelle. Zugriff 12 09, 2020, von http://ias.uni-klu.ac.at/projects/greybox/m01/206_PC_2.html

Evernine Insights. (2019, 10 30). *Diese Rolle spielen Farben im B2B Marketing*. Diese Rolle spielen Farben im B2B Marketing. Zugriff 12 12, 2020, von https://evernine-group.de/wie-sie-im-b2b-umfeld-modern-mit-farben-aktivieren-koennen/

Expo. (2020, 07 18). Expo. Zugriff 12 16, 2020, von https://expo.io/

Expo & Softwaremansion. (2020, 11 10). *React Navigation | React Navigation*. React Navigation | React Navigation. Zugriff 12 09, 2020, von https://reactnavigation.org/

FaceBook Inc. (2015, 03 26). *React Native. A framework for building native apps using React*. React Native. A framework for building native apps using React. Zugriff 12 09, 2020, von https://reactnative.dev/

FaceBook Inc. (2020, 10 28). *Color Reference. React Native*. Color Reference. React Native. Zugriff 12 09, 2020, von https://reactnative.dev/docs/colors

Focus Online. (2013, 11 16). *Extremismus. Braun war Farbe der Nazis in der NS-Zeit.* Extremismus. Braun war Farbe der Nazis in der NS-Zeit. Zugriff 12 09, 2020, von https://www.focus.de/politik/deutschland/extremismus-braun-war-farbe-der-nazis-in-der-ns-zeit_aid_685411.html

Gimm, S. (2019, 10 24). *Braun ist nicht meine Farbe.* Braun ist nicht meine Farbe. Zugriff 12 09, 2020, von https://www.blingblingover50.de/braun-ist-nicht-meine-farbe

Goethe, J. W. (1809). *Farbenkreis zur Symbolisierung des menschlichen Geistes- und Seelenlebens*.

Goethe, J. W. (1982). *Farbenlehre*.

Hunkel, K. (2011). *Ganzheitliche Farbberatung. Ein Ratgeber zur richtigen Farbentscheidung* (8th ed.). Schirner Verlag. 978-3-8434-4470-5

Internet Engineering Task Force. (2017, 10). *RFC 8258, Generalized SCSI*. RFC 8258, Generalized SCSI. Zugriff 12 16, 2020, von https://tools.ietf.org/rfc/rfc8258.txt

Mars, W. (n.d.). *The Evolution Of The Martian Color Wheel.* The Evolution Of The Martian Color Wheel. Zugriff 12 09, 2020, von http://warrenmars.com/visual_art/theory/colour_wheel/evolution/evolution.htm

Mars, W. (n.d.). *The Martian Color Wheel*. The Martian Color Wheel. Zugriff 12 09, 2020, von http://warrenmars.com/visual_art/theory/colour_wheel/colour_wheel.htm

Mars, W. (2014). *Colours Of The Martian Colour Wheel*. Colours Of The Martian Colour Wheel. Zugriff 12 12, 2020, von http://warrenmars.com/visual_art/theory/colour_wheel/wheel_colours.htm

Mars, W. (2014). *The Evolution Of The Martian Color Wheel*. The Evolution Of The Martian Color Wheel. Zugriff 12 09, 2020, von http://warrenmars.com/visual_art/theory/colour_wheel/evolution/evolution.htm

Mars, W. (2014). *The Martian Colour Wheel*. The Martian Colour Wheel. Zugriff 12 09, 2020, von http://warrenmars.com/visual_art/theory/colour_wheel/colour_wheel.htm

Mennerich, I. (2012, 06). *Bienen, Licht & Farbe*. Bienen, Licht & Farbe. Zugriff 12 12, 2020, von http://www.schulbiologiezentrum.info/Bienen/Bienen%20Licht%20und%20Farbe.pdf

Outfittery GmbH. (2012). *OUTFITTERY. Dein Stil. Dein Weg*. OUTFITTERY. Dein Stil. Dein Weg. Zugriff 11 16, 2020, von https://www.outfittery.de

Oza, H. (2017, 10 17). *Importance Of UI/UX Design In The Development Of Mobile Apps.* Importance Of UI/UX Design In The Development Of Mobile Apps. Zugriff 12 09,

2020, von https://www.hyperlinkinfosystem.com/blog/importance-of-uiux-design-in-the-development-of-mobile-apps#:~:text=A%20very%20good%20and%20efficient,generate%20more%20traffic%20and%20revenue

Qiao, J. (2016). *Colormind color palette generator.* Colormind color palette generator. Zugriff 11 16, 2020, von http://colormind.io

React Native Community. (2020, 11 02). *Async Storage. Data storage system for React Native*. Async Storage. Data storage system for React Native. Zugriff 12 09, 2020, von https://react-native-async-storage.github.io/async-storage/

Rödiger, J. (2018, 08 16). *RGB und CMYK - die richtige Anwendung für das perfekte Design.* RGB und CMYK - die richtige Anwendung für das perfekte Design. Zugriff 12 09, 2020, von https://www.media-company.eu/blog/allgemein/unterschied-rgb-und-cmyk/

Spektrum Akademischer Verlag. (2000). Frequenz. In *Lexikon der Physik* (Vol. 2, p. 2862). Spektrum Akademischer Verlag. 9783827414625

Vantongeren, R. (2018, 09 29). *How to Match Clothes When You're Clueless About Color-Matching*. How to Match Clothes When You're Clueless About Color-Matching. Zugriff 12 12, 2020, von https://restartyourstyle.com/740/how-to-match-clothes-when-clueless-about-color-matching

Wäger, M. (2017). *Das ABC der Farbe. Theorie und Praxis für Grafiker und Fotografen.* (1st ed.). Rheinwerk Verlag GmbH. 978-3-8362-4501-2

Wikipedia. (2020, 11 13). *HSV-Farbraum*. HSV-Farbraum. Zugriff 01 02, 2020, von https://de.wikipedia.org/wiki/HSV-Farbraum

Zalando SE. (2015). *Persönliche Stilberatung online | Zalon by Zalando DE*. Persönliche Stilberatung online | Zalon by Zalando DE. Zugriff 12 08, 2020, von https://www.zalon.de

Zalando SE. (2016). *Finden Sie heraus welcher Farbtyp Sie sind | Farbberatung Zalando.* Finden Sie heraus welcher Farbtyp Sie sind | Farbberatung Zalando. Zugriff 12 08, 2020, von https://www.zalando.de/farbberatung

Danksagung

Im folgenden Abschnitt möchte ich mich für die Unterstützung der folglich genannten Personen bedanken. Sie haben mich während der Erarbeitung der Bachelorarbeit sowie beim Testen der Anwendung unterstützt.

- *D. Albrecht*
- *S. Albrecht*
- *C. Ebel*
- *S. Ebel*
- *E. Felski*
- *M. Frank*
- *M. Heyer*
- *Prof. Dr. B. Messer*
- *T. Ostermaier*
- *E. Schmeltzer*
- *T. Stark*
- *Dr. H. Thiel*
- *D. Weigel*
- *S. Weigel*

Ein besonderer Dank gebührt *Warren Mars* für das Bereitstellen des von ihm geschaffenen Farbrads (vgl. 5.3 Martian Color Wheel).

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe angefertigt habe. Textpassagen, die wörtlich oder dem Sinn nach auf Publikationen oder Vorträgen anderer Autoren beruhen, sind als solche kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Berlin, 28.12.2020

_____

Fabian Frank