

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Gerlin Vainomäe 193877IADB

Mobiilirakenduse loomine harmoneeruvate värvide leidmiseks värviteooria alusel

Bakalaureusetöö

Juhendaja: Meelis Antoi
Magistrikraad

Tallinn 2022

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Gerlin Vainomäe

16.05.2022

Annotatsioon

Bakalaureusetöö eesmärk on luua mobiilirakendus, mis aitaks kasutajal tuvastada värvide nimetusi ja anda informatsiooni kahe valitud värvi kokkusobivuse kohta lähtudes värviteooria reeglitest. Loodava rakenduse peamine sihtgrupp on värvipimedad inimesed, kellel esineb raskusi värvide eristamisel, mis muudab keeruliseks ka värvide kooskõla tuvastamise.

Töö koosneb probleemi tausta uurimisest, rakenduse nõuete ja olulisemate funktsionaalsuste paika seadmisest, loodava rakenduse jaoks parima tehnoloogia valimisest, arenduse tulemuste kirjeldusest, testimisest ja võimalikest edasiarendustest.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 34 leheküljel, 7 peatükki, 13 joonist, 4 tabelit.

Abstract

Creation of a Mobile Application for Finding Harmonious Colors Based on Color Theory

The goal of this thesis is to create a mobile application that can identify colors in pictures selected by the user and find how harmonious are two chosen colors based on color theory. The target audience for this application is color-blind people who have difficulties distinguishing colors and, thus, find it difficult to identify color compatibilities.

This project includes researching the problem's background, describing the application's most important functionalities, finding the best technology for the project, analyzing the results, testing and suggesting ideas for future enhancements.

The created application includes the following functionalities: finding information about selected colors, saving colors for later viewing, finding color harmonies, and saving found harmonies.

The thesis is in Estonian and contains 34 pages of text, 7 chapters, 13 figures, 4 tables.

Lühendite ja mõistete sõnastik

Andmebaasi mootor	Programm, mille abiga opereeritakse andmebaasiga
Android	Google'i poolt arendatud mobiilsete seadmete operatsioonisüsteem
API	<i>Application Programming Interface</i> – rakendusliides
CMYK	<i>Cyan, Magenta, Yellow, Key</i> – värvimudel, mis kasutab värvide saamiseks tsüaani, fuksiinpunase, kollase ja võtmevärvi kombineerimist
CSV	<i>Comma Separated Values</i> – failitüüp, kus hoitakse andmeid komadega eraldatuna
DELETE	Kustutuspäring, mis saadab API serverisse korralduse kustutada valitud kirjed
GET	Päring, millega küsitakse API serverist üks või mitu kirjet
<i>Happy path</i>	Kasutusjuhtude täitumine vigadeta
HEX	Kuue sümboliga kood, mis kujutab värvi
HSL	<i>Hue, saturation, lightness</i> – värvimudel, mis kasutab värvide saamiseks värvitooni, küllastuse ja heleduse kombineerimist
HSV	<i>Hue, saturation, value</i> - värvimudel, mis kasutab värvide saamiseks värvitooni, küllastuse ja heleduse kombineerimist
HTTP	<i>Hypertext Transfer Protocol</i> – protokoll teabe edastamiseks ja vahetamiseks arvutivõrkudes
IDE	<i>Integrated Development Environment</i> – integreeritud arenduskeskkond koodi kirjutamiseks
iOS	<i>iPhone Operating System</i> – ettevõtte Apple poolt arendatud mobiilsete seadmete operatsioonisüsteem
JSON	<i>Javascript Object Notation</i> – Javascript'il põhinev andmevahetusvorming
Komplementaarne	Täiendav. Komplementaarvärvid ehk täiendvärvid
L*a*b*	Värvimudel, mis kasutab värvide saamiseks heleduse, punase-rohelise skaala ja kollase-sinise skaala kombineerimist

LCh	<i>Luminance, Chroma, Hue</i> – värvimudel, mis kasutab värvide saamiseks heleduse, kromaatilise väärtuse ja värvitooni kombineerimist
Monokromaatiline	Ühevärviline
ORM	<i>Object-Relational Mapping</i> – tehnika, mis võimaldab objekt orienteeritud programmeerimiskeeles teisendada andmeid ühest vormist teise
POST	Päring, millega saadetakse API serverisse andmeid kirje lisamiseks või uuendamiseks
REST	<i>Representational State Transfer</i> – veebiteenuste tarkvaraarhitektuuri liidese stiil
RGB	<i>Red, green, blue</i> – värvimudel, mis kasutab värvide saamiseks punase, rohelise ja sinise kombineerimist
SQL	<i>Structured Query Language</i> – struktuurpäringukeel andmebaasidega suhtlemiseks
<i>Unit testing</i>	Ühiktestimine, mille eesmärk on testida programmi väiksemaid osi
Värvipimedus	Haigus, mis tekitab raskusi ühe või mitme värvi eristamisel

Sisukord

1 Sissejuhatus	11
2 Projekti eesmärk	12
3 Taust	14
3.1 Sihtgrupp	14
3.2 Värviteooria	15
3.3 Värvimudelid	16
3.4 Olemasolevad lahendused	18
4 Loodava mobiilirakenduse analüüs	20
4.1 Nõuded.....	20
4.1.1 Funktsionaalsed nõuded	20
4.1.2 Mittefunktsionaalsed nõuded.....	21
4.2 Funktsionaalsuse disain	21
4.2.1 Voodiagramm	21
4.2.2 Andmebaas	22
4.2.3 Värviharmoniate arvutamine	23
4.3 Tehnoloogia valik	25
4.3.1 Programmeerimiskeele valik	25
4.3.2 Arenduskeskkonna valik	26
4.3.3 Tagarakenduse raamistik	27
4.3.4 Andmebaasi valik	28
4.3.5 Kliendipoolne tehnoloogia	29
5 Mobiilirakenduse arendus	31
5.1 Tagarakenduse arendus.....	31
5.1.1 Päringud.....	31
5.1.2 Algoritmid	33
5.2 Eesrakenduse arendus.....	34
5.2.1 Kasutajaliidese arhitektuur	34
5.2.2 Rakenduse vaated	35
5.2.3 Tagarakendusega suhtlus.....	39

6 Tulemused	40
6.1 Testimine	40
6.2 Realiseeritud funktsionaalsus	42
6.3 Edasiarendus	43
7 Kokkuvõte	44
Kasutatud kirjandus	45
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	49
Lisa 2 – Voodiagramm	50
Lisa 3 – Δ ECMC.....	51
Lisa 4 – Δ E94	52
Lisa 5 – Δ E2000.....	53
Lisa 6 – Mobiilirakenduse kättesaadavus.....	55

Jooniste loetelu

Joonis 1. HSL- ja HSV-värvimudelid [10].	18
Joonis 2. Vaadete vahel tagasi navigeerimise voodiagrammi näide.	22
Joonis 3. Andmebaasiskeem.	23
Joonis 4. Põhivärvitoonide vahemikud HSV-värvimudelis.	33
Joonis 5. Mobiilirakenduse avavaade.	35
Joonis 6. Mobiilirakenduse salvestatud värvide vaade nimekirjas (vasakul) ja avatuna (paremal).	36
Joonis 7. Mobiilirakenduse salvestatud värviharmoniate vaade.	36
Joonis 8. Mobiilirakenduse pildi valimise vaade enne (vasakul) ja pärast (paremal) esimese värvi valimist.	37
Joonis 9. Mobiilirakenduse pildi vaade enne (vasakul) ja pärast (paremal) esimese värvi valimist.	38
Joonis 10. Mobiilirakenduse harmoneeruvate värvide hüpikaknaga (vasakul) ja tavaline (paremal) vaade.	39
Joonis 11. Värvide salvestamise vaatesse lisatud kommentaariväli.	40
Joonis 12. Täiendatud avaekraani menüü (vasakul) ja esimene kasutusjuhendi vaade (paremal).	41
Joonis 13. Täiendatud pildi valimise menüü.	42

Tabelite loetelu

Tabel 1. Programmeerimiskeelte võrdlus.....	26
Tabel 2. Tagarakenduse raamistike võrdlus.	28
Tabel 3. Klientrakenduse raamistike võrdlus.	30
Tabel 4. Mobiilirakenduse API päringud.	32

1 Sissejuhatus

Värvipimedus on haigus, mis põhjustab raskusi värvide mõistmisel. Sellel esineb erinevaid kergemaid ja raskemaid alamvorme, kuid laias perspektiivis on kõigil värvipimedatel inimestel ühine probleem – ei suudeta eristada ega tuvastada värve. Sellest tulenevalt võib muutuda keeruliseks igapäevases vestluses, tööülesannete täitmisel ja mitmetes teistes situatsioonides konteksti mõistmine.

Kuna värvipimedus tekitab raskusi värvide eristamisel, ei osata ka alati leida inimese silmale esteetilisi värvikombinatsioone. Sobivate harmoneeruvate värvide leidmine osutub oluliseks näiteks disainimisel, kodu jaoks sobiva mööbli leidmisel kui ka endale meeldiva riietuse valikul.

Käesoleva lõputöö eesmärk on analüüsi teel leida kõige parem lahendus mobiilirakendusele, mis võimaldab rakenduse kasutajal tuvastada valitud piltidelt värve ning leida värviharmoniaid.

Eesmärgi saavutamiseks uuritakse teemaga seonduvaid teoreetilisi allikaid ja olemasolevaid lahendusi, vormistatakse nõuded, võrreldakse erinevaid rakenduse loomiseks võimalikke tehnoloogiaid, luuakse mobiilirakendus ja tehakse järelalus tulemusest.

2 Projekti eesmärk

Projekti eesmärk on luua mobiilirakendus, mis tuvastab valitud pildilt värvinime ning leiab kui hästi kaks valitud värvi omavahel harmoneeruvad lähtuvalt värviteooria reeglitest.

Mobiilirakendus aitab ennekõike värvipimedusega inimesi, kel esineb igapäevaselt raskusi värvide tuvastamisel ja kokkusobivate värvide leidmisel.

Lahendus luuakse mobiiliseadmetele, sest autor leiab, et telefoni pildigaleriist on kasutajal kõige mugavam pilte valida ning mobiiltelefoniga esineb ka võimalus pildistada. Pildistamise võimalus tähendab, et kasutajal saab kohe päriselus nähtava värvi kohta kiiresti informatsiooni.

Lõputöö loodava rakenduse skooopi kuuluvad järgnevad punktid:

- Kasutajal on võimalus valida oma mobiilseadme pildigaleriist pilt.
- Kasutajal on võimalus rakenduses pildistada ja loodud pilti rakenduses kasutada.
- Peale pildi valimist ja pildil konkreetse punkti valimist kuvatakse ekraanil värvinimetused, värvitoon ja RGB-kood.
- Peale värvipunkti valimist avaneb võimalus korrata samme punktides 2 ja 3, et valida teine värv.
- Peale kahe värvi valimist on kasutajal võimalik teada saada kui hästi värvid harmoneeruvad.
- Valitud värve on kasutajal võimalik salvestada, hiljem vaadata, kasutada ja kustutada.
- Kasutajal on võimalus salvestada info kahe värvi värviharmoniaate kohta ja seda hiljem vaadata ning kustutada.

Kuna erinevaid esteetilisi värvikoosluseid ja -skeeme on palju ning neid lisandub pidevalt juurde, lähtub autor harmoneeruvate värvide leidmisel ainult põhilistest ja tuntumatest värviteooria tavadest.

Leitud värvi valimisel näidatakse RGB-koodi teiste võimalike värvikoodide asemel, sest RGB-kood on kõige universaalsemalt tuntud värvivorming.

3 Taust

Lõputöös käsitletava probleemi tausta uurimine loob aluse mõistmaks loodava rakenduse vajaminevatest funktsionaalsustest ja kasutajaliidesest. Rakenduses soovitakse teha erinevaid arvutusi seoses värvidega, mis tähendab, et eelnevalt on vaja omada arusaama värviteooriast ja värvimudelitest.

3.1 Sihtgrupp

Antud lõputöö eesmärk on luua eelkõige abivahend värvipimedatele inimestele, mistõttu on oluline tutvuda lähemalt erinevate värvipimeduse vormidega ja sellest tulenevate raskustega.

Värvipimeduseks nimetatakse haigust, mille korral ei suuda inimene eristada teatud värve, sagedamini rohelisi ning punaseid toone. Enamus juhtudel on värvipimedus kaasasündinud ning teadaolevalt kaasasündinud värvipimedust ravida ei ole võimalik. Kaasasündinud värvipimeduse korral on see enimlevinud meestel, sest värvipimedus kandub edasi X-kromosoomi pidi. Naistel on kaks X-kromosoomi, mis tähendab, et kui üks neist on terve ja teine haigusega, siis haigust ei avaldu. [1]

Kuigi ülemaailmselt on värvipimedus ligikaudu 8% meestest ja 0,5% naistest, siis näiteks Skandinaaviamaades on mehi, kel esineb värvipimedus, ligikaudu 10-11%. See tähendab, et värvipimedust esineb kõige rohkem Kaukaasia rahvastel. [2]

Värvipimedad isikud saavad värve eristada ainult ereduse ja intensiivsuse järgi [3].

Värvuste tajumise häiretel on erinevaid raskusastmeid ning need jagunevad järgmiselt:

- Anomaalne trikromaasia – isikud näevad kõiki kolme põhivärvust, kuid värviaistingu kandjate töö on alatalituslik. Anomaalne trikromaasia jaguneb protanomaaliaks ehk vähenenud tundlikkus punase valguse suhtes, deuteranomaaliaks ehk vähenenud tundlikkus rohelise valguse suhtes ja tritanomaaliaks ehk vähenenud tundlikkus sinise valguse suhtes. [2]
- Dikromaatiline nägemine – isikud näevad ainult kahte põhivärvi. Dikromaatia jaguneb vastavalt protanooopiaks ehk punapimedus, deuteranooopiaks ehk rohepimedus ja tritanooopiaks ehk sinipimedus. [4, lk 12]

- Monokromaasia ehk akromatopsia – täielik värvipimedus ehk isikud ei näe üldse värvusi. Seda liiki värvipimedust esineb alla 0,01% rahvastikust. [3], [4, lk 12]

Kõige rohkem esineb deuteranomaaliat - seda on 5% meestest. Deuteranopiat, protanopiat ja protanomaaliat on kõiki vastavalt 1%. Ligikaudu pooltel värvipimedatel inimestel on kerge anomaalne vaegus ning ülejäänutel mõõdukas või raskekujuline. Tritanopia, tritanomaalia ja akromaasia esineb umbes ühel inimesel 30 000 kuni 50 000 inimese kohta. [2]

3.2 Värviteooria

Käesoleva projekti raames on otsustatud leida värvide harmooniaid ning luua selle jaoks algoritm, mistõttu on esmalt oluline tutvuda värviteooria põhimõtetega.

Andrus Rinde, Tallinna Ülikooli multimeediumi lektor, on kirjutanud, et „kunstis on värviteooria värvide segamise ja visuaalse mõjuga seotud praktiliste soovitude ja reeglite kogum“ [5].

Kuigi esimesed värviteooria printsiibid on kirjeldatud 15. sajandil, lähtutakse klassikaliselt värviteooria traditsioonidest 18. sajandi tekstidest, mille autoriks on Isaac Newton, kes koostas ka esimese värviringi. Algselt lähtuti kolmest primaarsest värvist: punane, kollane ja sinine, ning arvati, et neist on võimalik segada kokku kõik teised värvid. 19. sajandi lõpus aga jõuti järeldusele, et värvide tajumist saab kõige paremini kirjeldada punase, roheline ja sinise primaarvärvi abil. Nende kolme värvi akronüüm on tuntud kui RGB (*red, green, blue*). Neid värve peetakse ka tänapäeval primaarvärvideks. [5]

Värviringidest kõige tuntum ja teoreetilise alusena kõige sagedamini kasutatavam on Johannes Itteni 12 osaline värviring, mis on arendatud kollasest, punasest ja sinisest [4, lk 32].

Marje Tammert defineerib värviharmooniat järgmiselt: „Värvikombinatsioon on kahe või enama värvi kõrvutamisel tekkinud uus kvaliteet, mille puhul tavatsetakse rääkida värvide harmoonilisest sobivusest – värviharmooniast“. [4, lk 55]

Klassikaliseim viis värviharmoniaate leidmiseks on kujutada värviringil geomeetrilisi kujundeid, mille tipud tähistavad kokkusobivaid värve. Kõige levinumad reeglid on järgmised:

- Monokromaatiline harmoonia – kõik harmoneeruvad värvid pärinevad ühest värvitoonist (näiteks punane). Skeemi värvid erinevad teineteisest ainult mõne heledus-tumedusastme võrra (näiteks helepunane ja tumepunane). [6]
- Analooovärvid – värviringis lähestikku asetsevad värvid. Üldiselt peetakse analooovärvideks 60 kraadise nurga poolt hõlmatavaid värvitoonide üleminekuid. [6]
- Komplementaarne harmoonia – värviringis üksteise vastas paiknevad värvid reeglina harmoneeruvad (näiteks punane ja roheline). [7]
- Kolmkõlad – kolm värvi, mis paiknevad üksteisest värviringis võrdsel kaugusel (näiteks punane, kollane, sinine). [7]
- Nelikõlad – neli värvi, mis paiknevad üksteisest värviringis võrdsel kaugusel (näiteks kollane, lilla, roheline, sinine). [7]

Monokromaatilise harmoonia, analooovärvide ja komplementaarse harmoonia põhjal on võimalik ka moodustada suuremaid värviskeeme. [7]

Eelmainitud värvikombinatsiooni leidmise reeglite erandiks võib pidada akromaatilisi värve ehk must, valge ja hall, mis sobivad kokku paljude erinevate värvidega. [8]

3.3 Värvimudelid

Värvimudel on abstraktne matemaatiline mudel, millele värviruum on ülesehitatud ning mis kirjeldab erinevaid värve numbriliste järjenditena. Värviruum on seadme värviulatuse geomeetriline esitus, milles kõik värvid on esitatud punktidenä mingis värvimudelis. [5], [9]

Peatükis on toodud välja enimtuntud värvimudelid, mida kasutatakse ka projektis arvutuste tegemiseks.

Kõige tuntum on RGB-värvimudel, mis on kasutusel otsese valguse korral. Otsest valgust kiirgavad näiteks päike ja arvutiekraan. Kõik värvid saadakse kolme primaarvärvi kombineerimisel. Kolmeks primaarvärviks on punane (R - *red*), roheline (G - *green*) ja sinine (B - *blue*). RGB on aditiivne ehk liitev mudel, mis tähendab, et must tähistab valguse puudumist (RGB:0,0,0) ja valge saadakse primaarvärvide maksimaalse koguse segamisel (RGB:255,255,255). [9]

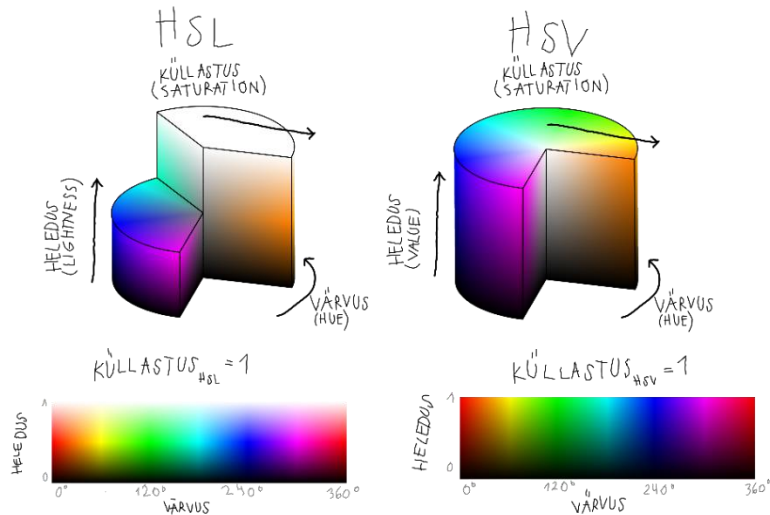
CMY-värvimudel on kasutusel valguse peegeldumise korral. Sel juhul on kolm primaarvärvi tsüaan (C - *cyan*), magenta (M - *magenta*) ja kollane (Y - *yellow*). CMY-värvimudel on subtraktiivne ehk lahutav, mistõttu must saadakse primaarvärvide seguna (CMY:100,100,100) ja valge värvide puudumisel (CMY:0,0,0). CMY-värvimudelis jäävad aga tumedad värvid tuhmiks, mistõttu on lisatud puhas musta värvi jaoks võtmevärv (K - *key*) ehk värvimudeli uus lühend on CMYK. CMYK-mudelis musta värvi esitusviis on (CMYK:0,0,0,100). [9]

Eelnimetatud mudelid on head värvide kujutlemiseks arvuti jaoks, kuid mitte inimese, mistõttu on loodud intuiitsemaid variante lähtudes inimese nägemisest. Tuntumad nendest on HSL- ja HSV-värvimudelid. Värvide defineeritakse sel juhul värvuse ehk tooni (H - *hue*), küllastuse (S - *saturation*) ja heleduse (L/V - *lightness/value*) kaudu. Nii HSL kui ka HSV on silindrilise ülesehitusega, kus iga punkt on defineeritud ühe nurga ja kahe koordinaadiga. Silindri iga sektor tähistab erineva värvusega värvi. Silindri nurk määrab värvuse, kaugus keskteljest küllastuse ja kaugus põhjasuunast heleduse. [10]

HSL-värvimudelit peetakse HSV-mudelist paremaks, sest see kujutab paremini intuiitset arusaamist küllastusest ja heledusest [5].

Kui HSV-mudelis segatakse puhast värvi valgega ehk toodetakse heledusastmeid, siis küllastus väheneb, kuid mustaga segamine ehk tumedusastmete tootmine jätab küllastuse muutumatuks. HSL-värvimudeli puhul esinevad nii heledus- kui tumedusastmed täisküllastuses. [9]

HSL- ja HSV-mudelid on kujutatud joonisel 1.



Joonis 1. HSL- ja HSV-värvimudelid [10].

CIELAB, $L^*a^*b^*$ või CIE Lab on 1976. aastal Rahvusvahelise Valgustuskomisjoni poolt kasutusele võetud värvimudel. Antud mudel proovib värve kirjeldada nii nagu seda inimsilm tajub ja eemaldada sõltuvus erinevatest seadmetest, kuna erinevatel seadmetel ja monitoridel näivad värvid erinevad. $L^*a^*b^*$ -värvimudeli põhikarakteristikud on heledus (L - *luminance*), värvuskomponent rohelisest punaseni (a) ja värvuskomponent sinisest kollaseni (b). L^* väärtus on vahemikus 0 kuni 100, kus 0 on must ja 100 valge. Komponentid a ja b varieeruvad +127'st kuni -128'ni. Selle värvimudeli ulatus on suurem RGB- ja CMYK-mudelitest. [5], [9]

$L^*a^*b^*$ 'iga samas värviruumis on LCh, mis koosneb heledusest (L - *luminance*), kromaatilisusest (C - *chroma*) ja värvitooni (h - *hue*) väärtusest. $L^*a^*b^*$ kasutab ristkoordinaadistikku, kuid LCh polaarkoordinaadistikku. Kromaatilisus on joone pikkus nullist punktini (a, b) ja värvitoon on selle joone nurk koordinaadistiku suhtes. [11]

3.4 Olemasolevad lahendused

Lõputöö autori otsingute põhjal ei ole olemas täpselt samasuguse põhimõttega mobiilirakendus, mida soovitakse projekti raames luua. Siiski on olemas rakendusi, mis võimaldavad valida pildilt värvi ning kuvada värvinimetusi ning rakendusi, mis võimaldavad valida värvi ja leida selle põhjal harmoneeruva värviskeemi. Järgnevalt on uuritud populaarsemaid eelmainitud funktsionaalsusega mobiilirakendusi.

Color harmony – rakendus on mõeldud peamiselt värvipalettide moodustamiseks. Värvide valimiseks on võimalus kaameraga valida reaalsest elust mõni värv või hoopis sisestada enda soovitud RGB-kood. Värvikooslusi saab moodustada valides värvid ise või lastes rakendusel valida värve värviteooria reeglite abil, mis on kirjeldatud ka peatükis „Värviteooria”. Puudub aga valik teada saada kas kasutaja poolt loodud värviskeem on harmooniline.

Color blind pal – rakenduse idee on kuvada kasutajale ekraani keskpunktis olev värv, mis leitakse vaadates telefoni kaameraga enda ümbrust. Lisaks saab lisada erinevaid filtreid, mis aitavad värvipimedatel isikudel paremini värve mõista (näiteks tuues esile konkreetsed värvitoonid mustrite abil) või hoopis simuleerivad erinevaid värvipimeduse vorme. Rakenduses on hästi läbi mõeldud värvide kuvamine. Kaameraga on võimalik küll hetke pilti näha ja peatada, kuid galeriist ei saa pilte valida.

Color grab – rakendus toimib sarnaselt rakendusega Color harmony, kuid rakenduse fookus on värvide valimisel, mitte palettide loomisel. Värvide saab valida nii vaadates telefonikaameraga ümbrust kui ka valides galeriist pildi. Valitud värvide kohta on saadaval palju informatsiooni, näiteks värv erinevates värvimudelites, värvi heledustase ja üldine värvitoon. Palettide moodustamisel on samuti arvestatud värviteooria reeglitega.

4 Loodava mobiilirakenduse analüüs

Selles peatükis määratakse loodava mobiilirakenduse põhilised funktsionaalsed ja mittefunktsionaalsed nõuded, kirjeldatakse lahti olulisemate funktsionaalsuste ülddisain ning valitakse tehnoloogiad.

4.1 Nõuded

Mobiilirakenduse loomiseks on välja toodud kasutuslugudel põhinevad funktsionaalsed ja mittefunktsionaalsed nõuded, mis annavad ülevaate plaanitavas rakenduses tehtavatest tegevustest ja selle üldisest funktsionaalsest käitumisest.

Nõuded määratakse ainult tavakasutaja perspektiivist, sest loodav mobiilirakendus on kasutajatele personaalne ning projekti skoopi ei kuulu kasutajate süsteemi loomine.

4.1.1 Funktsionaalsed nõuded

Rakenduse kasutaja funktsionaalsed nõuded on järgmised:

- Kasutajana tahan valida telefoni pildigaleriis paiknevate piltide seast pildi.
- Kasutajana tahan luua telefoni kaameraga rakenduses pildi.
- Kasutajana tahan, et kaameraga pildistatud pilt oleks rakenduses kohe nähtav ja kasutatav.
- Kasutajana tahan pildil suvalisele punktile vajutades näha punkti juures paikneva värvinimetust ja selle värvi RGB-koodi.
- Kasutajana tahan tuvastatud värvi salvestada.
- Kasutajana tahan näha nimekirja salvestatud värvidest ja nimekirjast värve kustutada.
- Kasutajana tahan salvestatud värvi kasutada harmoneeruvate värvide leidmiseks.
- Kasutajana tahan peale esimese pildi valimist, pildistamist või värvi valimist, valida või pildistada uue pildi.

- Kasutajana tahan, et peale värvide valimist oleks võimalik näha kuidas ja milliste reeglite põhjal valitud värvid omavahel harmoneeruvad.
- Kasutajana tahan kahe värvi harmoneerumise andmeid salvestada, neid hiljem vaadata ja soovi korral kustutada.

4.1.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded on järgnevad:

- Rakendus on kasutatav mobiilsetel seadmetel.
- Rakenduse sisuline pool on mõistetav inglise keele kõnelejatele.
- Kasutajaliideses eristub tekst ja muu vajalik informatsioon selgelt taustast.
- Värvide kirjeldustes on piisavalt informatsiooni loomaks kasutajale hea ülevaate värvide olemusest.

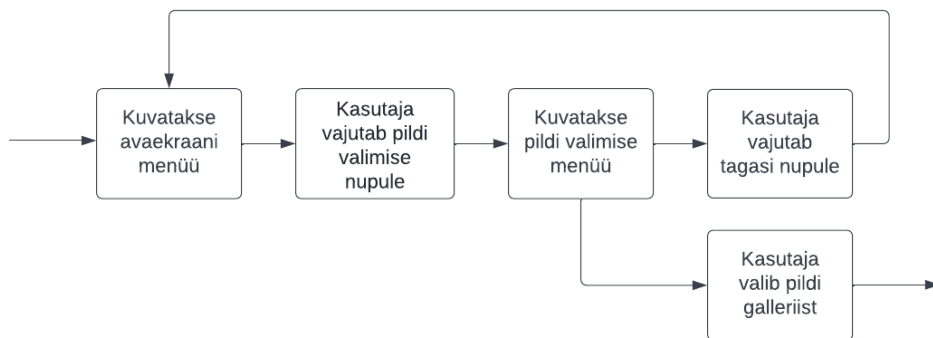
4.2 Funktsionaalsuse disain

Peatükis kirjeldatakse detailsemalt mobiilirakenduse loodavate funktsionaalsuste tausta – kuidas on võimalik mobiilirakenduses erinevate vaadete vahel navigeerida ning mis on erinevate tegevuste tulemuseks, milline on andmebaasiskeem ja kuidas hakatakse programmikoodis värviharmoniate leidmist arvutama.

4.2.1 Voodiagramm

Loodava rakenduse kasutajakogemuse paremaks mõistmiseks on loodud voodiagramm, mis näitab rakenduse ideaalset teekonda ehk *happy path*. Ideaalne teekond kirjeldab ideaalset juhtu, milles ei esine vigu ega erisuguseid alternatiivseid võimalusi.

Rakenduses on kasutajal alati võimalik pöörduda tagasi eelmisesse vaatesse, kuid diagrammi lihtsuse huvides ei ole märgitud tagasi minemise võimalust. Joonisel 2 on demonstreeritud kuidas kujutatakse diagrammil ühe vaate juurest teise juurde tagasi minemist. Joonis kujutab ainult väikest osa rakenduse funktsionaalsusest, mistõttu ei ole joonisel näidatud konkreetset algus- ega lõpp-punkti, vaid algus ja lõpp on tähistatud diagrammist siseneva ja sealt väljuva noolega.



Joonis 2. Vaadete vahel tagasi navigeerimise voodiagrammi näide.

Kasutajakogemuse voodiagramm on kujutatud peatükis Lisa 2. Diagramm on jaotatud kolmeks osaks, kus esimene osa pealkirjaga „Kasutaja“ tähistab kasutaja poolt tehtavaid tegevusi, „Süsteem“ klientrakenduse tegevusi ning „Tagarakendus“ serveripoolseid tegevusi. Juhul kui skeemil on sõna „värv“ juures kuvatud tärniga mäрге, kehtib sama funktsionaalsus värviharmoniate kohta.

4.2.2 Andmebaas

Andmebaasi eesmärk on hoiustada rakenduse töö toimimise jaoks vajaminevaid andmeid.

Rakenduses plaanitakse luua nii värvide kui ka värviharmoniate salvestamine, mistõttu koosneb andmebaasi skeem kolmest tabelist. Kaks peamist tabelit on „värv“ ja „värvipalett“. Kahe tabeli vahel on mitu-mitmele seos, mis tähendab, et tabelis „värvipalett“ võib olla mitu kirjet tabelist „värv“ ning ka vastupidi, mistõttu on olemas ka kolmas tabel, mis ühendab eelmainitud kaks tabelit.

Tabel „värv“ hoiab informatsiooni antud värvi identifitseerimisnumbri, nimetuse, HEX-koodi, värvitooni ja ajatempli kohta. Identifitseerimisnumber on ühe värvi unikaalne number, mis lisatakse tabeli kirjesse automaatselt. Selle abil on võimalik värve andmebaasist kustutada. Ka ajatempel määratakse programmis automaatselt võttes kasutusele lisamise hetkeaja. Lisaks on tabelis ka rida, mis määrab kas salvestatud värvide tabelis peab kuvama käesoleva värvi või mitte. See on oluline, sest värv võib olla küll andmebaasi tabelis, sest sellega on seotud värvipalett, kuid kasutaja ise ei ole manuaalselt värvi salvestanud ning seetõttu ei soovi värvi ka tabelis näha.

Tabel „värvipalett“ hoiab informatsiooni värvipaleti identifitseerimisnumbri, monokromaatilise harmoonia, analoogvärvide, komplementaarse harmoonia, neutraalsete värvide harmoonia ja värvipaleti loomisaja kohta.

Tabelid ning tabelite vaheline seos on kujutatud joonisel 3.



Joonis 3. Andmebaasiskeem.

4.2.3 Värviharmoniate arvutamine

Rakenduses soovitakse kasutada algoritmi, mis arvutaks kui hästi kaks kasutaja poolt valitud värvi omavahel harmoneeruvad. Loetavuse huvides tähistatakse valitud värvid tähtedega X ja Y. Värviharmoniate leidmiseks on plaanitud esialgu leida värviga X kokkusobivad värvid kasutades värviteooria reegleid. Kui on teada millised värvid sobivad kokku värviga X, saab leitud värve võrrelda värviga Y. Nii saab teha järelduse, et mida lähedasem on värv Y värvi X harmoneeruvatele värvidele, seda paremini sobivad kokku värv X ja Y.

Värvide sarnasust on aga keeruline hinnata, sest inimesed näevad värve erinevalt [12]. Lisaks ei taju inimsilm kõiki värvi eripärasid võrdselt. Kõige paremini eristatakse värvitooni erinevusi, seejärel värvilisust ehk küllastust ning kõige viimasena heledust. [13]

ΔE on kahe värvi visuaalse tajumise muutuse mõõt, mis aitab luua arusaama kuidas inimesed tajuvad värvide erinevust. Mida väiksem on ΔE , seda sarnasemad on värvid. ΔE väärtus on tavaliselt vahemikus 0-100. Kui väärtus on alla 1, siis inimsilm erinevust ei taju, vahemikus 1-10 on erinevus väike, 11-49 on erinevus märgata, kuid värvid on pigem sarnased kui erinevad ning väärtuse 100 juures on värvid vastandid. [14]

Kõige levinum ja lihtsam viis värvide sarnasuse leidmiseks on kasutada Eukleidese kaugust. Seda meetodit tutvustas 1976. aastal Rahvusvaheline Valgustuskomisjon (CIE). Valemi tähis on ΔE_{ab} . [15]

$$\Delta E_{ab} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2}$$

$L_1 = L^*a^*b^*$ värviformaadis oleva esimese värvi L^* väärtus.

$a_1 = L^*a^*b^*$ värviformaadis oleva esimese värvi a^* väärtus.

$b_1 = L^*a^*b^*$ värviformaadis oleva esimese värvi b^* väärtus.

$L_2 = L^*a^*b^*$ värviformaadis oleva teise värvi L^* väärtus.

$a_2 = L^*a^*b^*$ värviformaadis oleva teise värvi a^* väärtus.

$b_2 = L^*a^*b^*$ värviformaadis oleva teise värvi b^* väärtus.

1984. aastal täiendas CMC (Colour Measurement Committee of the Society of Dyers and Colourists) meetodit lisades valemile juurde võimaluse mitmekordistada heleduse ja küllastuse osakaalu (tähis ΔE_{CMC}). Samuti hakati teoreemis $L^*a^*b^*$ -värvimudeli asemel kasutama LCh-värvimudelit. Need kaks mudelit erinevad selle poolest, et LCh kujutab värvitooni nurgana, mitte lõpmatu arvu värvipunktidenä. See võimaldab hõlpsamini sooritada tõrkeotsingut ja arvutada värvitoone. Antud valemis parameetrid l ja c võimaldavad kasutajal määrata vastavalt vajadusele heleduse (l - *lightness*) ja kromaatilisuse (c - *chroma*) kaalud. Parameetrid S_L , S_C ja S_H korrigeerivad vastavalt heledust, kromaatilisust ja värvust, et värvide erinevus oleks määratletud sarnaselt sellele kuidas inimsilm tajub värvide erinevusi. [15]

Valemi ΔE_{CMC} arvutuskäik on kirjeldatud peatükis Lisa 3 [16].

$$\Delta E_{CMC} = \sqrt{\left(\frac{\Delta L}{lS_L}\right)^2 + \left(\frac{\Delta C}{cS_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2}$$

1995. aastal (tähis ΔE_{94}) ja 2000. aastal (tähis ΔE_{2000}) lisati valemile veel koefitsiente juurde, mille töötasid välja spetsialistid analüüsid erinevaid võimalikke tulemusi. Valem ei ole küll täiuslik, kuid 2000. aastal täiendatud valemit peetakse tänapäeval üheks kõige paremaks viisiks värvi erinevuste arvutamiseks, sest see on kõige lähedasem

sellele, kuidas inimesed tajuvad värve. Parameetrid k_L , k_C ja k_H on lisatud, et võtta arvesse eksperimentaalsete vaatamistingimuste mõju. Parameeter R_T on valemile juurde pandud, et korrigeerida saadud tulemust olukorras, kus ühe värvi toon satub sinise tooni piirkonda. [15]

Valem ΔE_{94} on lahti kirjutatud peatükis Lisa 4 ja valem ΔE_{2000} peatükis Lisa 5 [16].

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L}{K_L S_L}\right)^2 + \left(\frac{\Delta C}{K_C S_C}\right)^2 + \left(\frac{\Delta H}{K_H S_H}\right)^2}$$

$$\Delta E_{2000} = \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right) \left(\frac{\Delta H'}{K_H S_H}\right)}$$

4.3 Tehnoloogia valik

Rakenduste arendamiseks on loodud palju erinevaid tehnoloogiaid, mis määravad arenduskäigu sujuvuse ning üldised koodi kirjutamise piirangud. Käesolevas peatükis analüüsitakse millised tehnoloogiad on parimad plaanitava mobiilirakenduse loomiseks. Valimisel lähtutakse tehnoloogiate tuntusest ja kasutajaskondade suurusest kui ka autori isiklikest kogemustest.

4.3.1 Programmeerimiskeele valik

Projekti tagarakenduse programmeerimiskeele valikul lähtutakse soovist kasutada OpenCV raamistikku, mis on ettenähtud arvutinägemise, masinõppe ja pilditötluse jaoks [17]. Raamistik toetab mitmeid programmeerimiskeeli, täpsemalt Python, C++, Java ja MATLAB [18]. Eelnimetatud programmeerimiskeelte võrdlus on toodud tabelis 1.

Tabel 1. Programmeerimiskeelte võrdlus.

Keel	Plussid	Miinused
C++	Hea OpenCV tugi laia kasutajaskonnaga, kiire, lühikesed koodiread. [19]	Puudub kogemus, manuaalne mälu haldamine [20], platvormist sõltuv, limiteeritud raamistike toetus. [19]
Java	Kogemus olemas, kiirem Python'ist. [19]	Vähem informatsiooni OpenCV kohta võrreldes teiste keeltega, pikad koodiread, aeglasem C++'st. [19]
MATLAB	Hea matemaatilisteks arvutusteks.	Puudub kogemus, suletud lähtekoodiga. [21]
Python	Kogemus olemas, toetab palju erinevaid raamistikke, lihtne õppida, lühikesed koodiread. [19]	Aeglane, võtab palju mälu. [19]

OpenCV põhineb C++ keelel ning on seetõttu ka kõige populaarsem valik raamistiku kasutamisel [18]. C++'i aga ei valita, sest töö autoril puudub varasem kogemus selle programmeerimiskeelega ning keele juurde õppimine võtaks palju lisa-aega. Samal põhjusel ei võeta kasutusele ka MATLAB'i.

OpenCV'l on olemas ka Java API, kuid selle keele toetus raamistiku kasutamiseks ei ole nii suur ning seetõttu oleksid ka võimalused, võrreldes Python'i ja C++'iga, limiteeritud.

Python'i ja Java võrdluses eelistab autor kasutada Python'it, sest sellel on parem tugi ja rohkem infot OpenCV raamistiku kasutusel ning Python'iga on mugavam arvutusi teha, sest sellel on mitmeid matemaatilisi raamistikke (näiteks NumPy), mis teevad koodi lihtsamaks ja loetavamaks. Arvutuste tegemine on projektis vajalik piltide töötlemiseks ja värviharmooniate leidmiseks.

4.3.2 Arenduskeskkonna valik

Arenduskeskkonna ehk IDE (*Integrated Development Environment*) valikul lähtus autor varasemalt kasutatud Python'i jaoks loodud keskkondadest. Nendeks on PyCharm, Spyder ja IDLE. PyCharm on võrreldes kahe teise mainitud IDE'ga palju võimekam, sest

sellel on rohkem pluginaid ning seda on võimalik arendajal vastavalt oma vajadustest rohkem kohandada. Autor eelistab isiklikult kõige rohkem PyCharm'i, kuna on olemas pikaajaline kogemus JetBrains'i (PyCharm'i loonud ettevõtte) arenduskeskkondadega ja autor leiab, et antud keskkonnas on failihaldus kõige mugavam.

4.3.3 Tagarakenduse raamistik

Mobiilirakenduse loomiseks on otsustatud kasutada Python programmeerimiskeelt, mistõttu tuleb lähemalt tutvuda Python'i raamistikega.

Autor lähtus raamistiku valimisel populaarseimatest 2022. aasta valikutest. Valikute seas on Django, Flask, Hug, Bottle, CherryPy, Dash ja Falcon. Kõik eelmainitud raamistikud peale Django on mikroraamistikud. [22]

Kõige tuntumad ja kõige paremini dokumenteeritud on Django ja Flask. Django on mõeldud kasutamiseks komplekssetes ja suuremamahulistest projektides, kuid Flask on mikroraamistik. Flask on seetõttu parem valik ka käesoleva projekti jaoks, sest ei ole plaanitud luua mahukat süsteemi. Lisaks on Flask'il olemas hea *unit testing* ehk üksustestimise tugi ning mugav vigade jälgimise mehhanism. [22]

Tabelis 2 on toodud erinevate tagarakenduse raamistike plussid ja miinused, millega arvestati projekti jaoks sobiva raamistiku valikul.

Tabel 2. Tagarakenduse raamistike võrdlus.

Raamistik	Plussid	Miinused
Django	Kiire, paindlik, turvaline, hästi dokumenteeritud, lai kasutajaskond. [22]	Suuremahuline raamistik, raske õppida, vähe standardeid. [23]
Flask	Skaleeritav, paindlik, hästi dokumenteeritud, lai kasutajaskond. [24]	Laiendusi peab lisama manuaalselt. [24]
Hug	Kõige kiirem Python'i raamistik. [22]	Vähe informatsiooni, väike kasutajaskond.
Bottle	Kiire, väikesemahuga. [25]	Ei ole eriti skaleeritav [25], väike kasutajaskond.
CherryPy	Saadaval sisseehitatud serveriga, kiire, stabiilne. [26]	Vähe dokumenteeritud. [26]
Dash	Lihne kasutada, hea dokumentatsioon. [25]	Mõeldud eelkõige analüütiliste armatuurilaua vaadete loomiseks, kõik funktsionaalsused ei ole tasuta. [25]
Falcon	Hästi optimeeritud koodibaas, hea vigade ja haavatavuste haldus, kiire. [22]	Piiratud funktsionaalsus, vähe sõltuvusi. [27]

4.3.4 Andmebaasi valik

Andmebaasiga suhtlemiseks on esmalt vaja andmebaasi mootorit. Andmebaasi mootor on andmebaasi juures asuv programm, millele korralduste andmisega opereeritakse andmebaasiga [28].

Projektis soovitakse kasutada andmebaasi ainult eelnevalt valitud värvide ja leitud värviharmoniatega salvestamiseks ning kuna ei soovita ka rakendada kasutajate põhist süsteemi, siis selle projekti raames sobib kasutada väiksemahulist andmebaasi mootorit. Seetõttu otsustati valida SQLite. Autoril on ka olemas varasem kogemus teiste

andmebaasi mootoritega, näiteks PostgreSQL, MySQL, SQL Server, kuid need on sobivad pigem projektidesse, kus on vaja suurt andmebaasi.

Andmebaasi päringute lihtsustamiseks on hea tava ka kasutada ORM'i. ORM ehk *Object-Relational Mapping* on andmete muutmine ühest vormist teise kasutades objekt orienteeritud programmeerimiskeeli. Andmebaasi kontekstis see tähendab, et lihtsustatakse arendaja jaoks andmebaasiga suhtlemist ja üldjuhul ei ole vaja kirjutada ise SQL-lauseid.

Autor otsustas valida ORM raamistikuks SQLAlchemy, sest see on kiire, suure kasutajaskonnaga ja lihtne koodi lisada [29]. Lisaks on sellel olemas hea tugi Flask'iga. Täpsemalt on olemas laiendus nimega „Flask-SQLAlchemy“. SQLAlchemy toimib hästi ka SQLite'iga. SQLAlchemy võimaldab kasutada SQLite's ka kuupäeva ja kellaaja tüüpe, mis algselt ei ole SQLite's saadaval [30].

4.3.5 Kliendipoolne tehnoloogia

React Native on Facebooki poolt arendatud Javascript'i raamistik, mis toetab mobiilirakenduste loomist nii iOS kui ka Android telefonidele. Töö autor otsustas eesrakenduse arendamiseks kasutada React Native'it, sest see on üks kõige laialdasemalt tuntud ja kasutusel olev mobiilirakenduse loomise raamistik, mis tähendab, et projekti koodi arendusel tekkivate probleemide esinemisel on lahenduse leidmine tõenäolisemalt kiirem ja kergem võrreldes teiste raamistikega. Lisaks peetakse React Native'i plussideks selle head jõudlust, komponentide taaskasutatavust ning ühilduvust mitmete kolmanda osapoolte teenustega. [31]

Tabelis 3 on toodud võrdlus erinevate tuntumate mobiilirakenduse loomise raamistike vahel, millega arvestati valiku tegemisel. [32]

Tabel 3. Klientrakenduse raamistike võrdlus.

Raamistik	Plussid	Miinused
React Native	Ühilduvus erinevate seadmetega, kiire arendusprotsess, hea jõudlus, kiire tagasiside koodi muutuste kohta, taaskasutatavus. [32]	Raske vigade haldus, ühilduvusprobleemid, pole piisavalt mooduleid, raskesti õpitav. [32]
Flutter	Ühilduvus erinevate seadmetega, kiire tagasiside koodi muutuste kohta, kiire arendusprotsess. [32]	Väike kasutajaskond, suuremahuga rakendused. [32]
Xamarin	Hea jõudlus, on olemas riistvara toetus, mugav hallata, palju õppematerjale, ühilduvus erinevate seadmetega. [32]	Ühilduvusprobleemid, peab tundma C#'i, suuremahuga rakendused. [32]
Ionic	Ühilduvus erinevate seadmetega, kerge õppida, kiire prototüüpimine, mugav testida. [32]	Madal jõudlus, pluginatest sõltuv, ei saa kohe tagasisidet koodi muutmise korral, suuremahuga rakendused. [32]

5 Mobiilirakenduse arendus

Projekti alustas autor tagarakenduse arendusest. Kuna töö kõige olulisem komponent on värvide tuvastamine ning harmoneeruvate värvide leidmine, tuli esmalt tutvuda selle loomise võimalustega ehk teha valmis põhiline funktsionaalsus. Kui olulisemad funktsioonid olid olemas, sai alustada kliendipoolse ehk eesrakenduse arendusega.

5.1 Tagarakenduse arendus

Tagarakendus ehk teenus, mis tegeleb äriloogikaga, on loodud kasutades Flask'i ning omab SQLite andmebaasi, kus on talletatud kõik klientrakenduse äriloogikaga seonduv info. Järgnevates peatükkides on lähemalt kirjeldatud veebiteenuse arenduse erinevaid aspekte.

5.1.1 Päringud

Päringute tegemisel lähtutakse REST API arhitektuurist. REST API suhtleb veebiteenusega kasutades HTTP päringuid, et täita standardseid andmebaasifunktsionaalsusi, milleks on andmete loomine, pärimine, muutmine, kustutamine – need neli tegevust on tuntud kui CRUD (*Create, Read, Update, Delete*) [33]. Käesolevas projektis kasutatakse GET, POST, PUT ja DELETE toiminguid, andmed talletatakse päringus JSON kujul ning iga päringu vastusega kaasneb ka staatuse kood, mis näitab kas päring oli edukas või mitte.

Tabelis 4 on kirjeldatud projekti tagarakenduse kaudu tehtavad päringud.

Tabel 4. Mobiilirakenduse API päringud.

Päring	Toimingu tüüp	Kirjeldus
/colors	GET	Tagastatakse kõik andmebaasi salvestatud värvid.
/colors	POST	Andmebaasi lisatakse uus värv.
/colors	DELETE	Andmebaasist eemaldatakse id järgi värv.
/colors	PUT	Andmebaasis muudetakse värvi id järgi leitud värvi kommentaar. Päring lisandus peale testimist.
/colors/name	POST	Sisendina võetakse pilt, pildi värvipiirkondade detailsuse eelistus ning x- ja y-koordinaadid. Päringu vastusena tagastatakse valitud pildi koordinaatidel oleva värvinimetus, RGB-kood ja üldine värvitoon.
/harmonies/info	POST	Sisendina võetakse kaks RGB-koodi. Päringu vastusena tagastatakse kahe eelnevalt sisestatud värvi kokkusobivus värviteooria tavadest lähtuvalt.
/harmonies/colors	POST	Sisendina võetakse värvipaleti id ning tagastatakse selle järgi värvipaletis olevad värvid ning info nende kahe värvi harmoneerumise kohta.
/harmonies	GET	Tagastatakse kõik andmebaasi salvestatud värviharmoniate kohta käiv informatsioon.
/harmonies	POST	Andmebaasi salvestatakse kahe värvi kohta käiv info (juhul kui selle RGB-koodiga värve andmebaasis eelnevalt ei ole) ja värviharmoniad.
/harmonies	DELETE	Andmebaasist eemaldatakse id järgi kahe värvi värviharmoniad.
/harmonies	PUT	Andmebaasis muudetakse värvipaleti id järgi leitud värvipaleti kommentaar. Päring lisandus peale testimist.

5.1.2 Algoritmid

Käesolevas peatükis on kirjeldatud kuidas on teostatud pildi pealt värvi leidmine ning kuidas on loodud kahe värvi kokkusobivuse algoritm.

5.1.2.1 Värvide leidmine

Värvide leidmise algoritmi sisendid on pilt, pildi värvi piirkondade detailsuse eelistus ja koordinaadid ning selle algoritmi eesmärgiks on tagastada koordinaatidel paikneva värvinimetuse, üldine värvitoon ja RGB-kood.

Kui värvi piirkonna detailsuse eelistus on märgitud tõeseks, valitakse pildilt värv piksli järgi. Vastasel juhul segmenteeritakse pilt, mille tulemusena on pildi eri piirkondades paiknevad värvid üldistatud ning alles üldistatud värvidega pildilt leitakse värvinimi. Segmenteeritud pilti ei kuvata kasutajale, seda kasutatakse ainult tagarakenduses värvide leidmiseks.

Värvinimetuse leidmisel kasutatakse CSV-faili, mis on komadega eraldatud väärtuste fail [34], kus talletatakse informatsiooni värvinimede ning nende RGB- ja HEX-koodide kohta. Võttes kasutusele eelnevalt mainitud faili saab arvutada, milline failis olev RGB-kood on kõige lähedasem sisendina antud RGB-koodile.

Üldise värvitooni leidmisel kasutatakse HSV-värvimudelit, sest HSV'd kasutades on kerge määrata värvinimetuse vahemikke. Konverteerides RGB-koodi HSV-värvimudeli vormi ja käies läbi võimalikud värvi vahemikud, ongi võimalik leida üldine värvitoon. Näiteks oranži värvitooni vahemik HSV formaadis on (0, 19, 146) kuni (9, 255, 255). Värvide vahemikud on kujutatud joonisel 4.

```
color_dict_HSV = list({'Red1': [[9, 255, 255], [0, 19, 146]],
                      'Orange': [[24, 255, 255], [10, 19, 101]],
                      'Brown1': [[24, 255, 145], [0, 19, 35]],
                      'Yellow': [[35, 255, 255], [25, 19, 35]],
                      'Green': [[89, 255, 255], [36, 19, 35]],
                      'Blue': [[128, 255, 255], [90, 19, 35]],
                      'Purple': [[158, 255, 255], [129, 19, 35]],
                      'Red2': [[180, 255, 255], [159, 19, 146]],
                      'Brown2': [[180, 255, 145], [159, 19, 35]],
                      'Black': [[180, 255, 34], [0, 0, 0]],
                      'Gray': [[180, 18, 230], [0, 0, 35]],
                      'White': [[180, 18, 255], [0, 0, 231]]
                      }.items())
```

Joonis 4. Põhivärvitoonide vahemikud HSV-värvimudelis.

5.1.2.2 Värviharmoniate leidmine

Värviharmoniate leidmisel võeti kasutusele taustauuringu käigus leitud värviteooria reeglid: monokromaatiline, analoogiline ja komplementaarne harmoonia. Lisati ka eraldi meetod, mis kontrollib, kas kumbki sisendina antud värv on neutraalne (must, hall või valge), sest neutraalsed värvid sobivad kokku kõikide teiste värvidega.

Algoritmi alustatakse esimesena valitud värvile harmoneeruva värvi leidmisest. Selleks tuleb muuta RGB-värvimudelisse olev värv HSL-värvimudelisse, sest see värvimudel kuvab värvitooni väärtuse kraadides ja enamasti projektis kasutatavad värviteooria reeglid kokkusobivate värvide leidmiseks kasutavad kraade – näiteks komplementaarsed värvid paiknevad teineteisest värvirattal 180 kraadi kaugusel ja analoogsed värvid 60 kraadi kaugusel.

Kui harmoneeruvad värvid on leitud, arvutatakse kui sarnane on teisena valitud värv esimese värviga harmoneeruvatele värvidele. Selleks muudetakse värv $L^*a^*b^*$ värvimudelisse, mis lihtsustab ΔE leidmist. Mida väiksem on ΔE , seda sarnasemad on värvid. ΔE leidmise valemid on kirjeldatud peatükis „Värviharmoniate arvutamine”. Töö autor otsustas kasutada ΔE 2000. aastal välja töötatud versiooni, sest hinnanguliselt annab see kõige täpsema tulemuse.

5.2 Eesrakenduse arendus

Vastavalt analüüsile, valiti kasutajaliidese arendamiseks React Native.

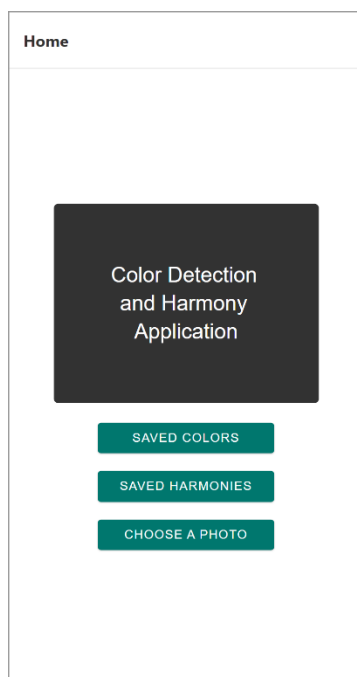
5.2.1 Kasutajaliidese arhitektuur

Kasutajaliides on struktuuriliselt jaotatud erinevate vaadete vahel, kus iga vaate kausta kuulub vaade ise, vaates kasutatav päringute fail, stiilifail ja vajaduse korral ka vaate faili loetavuse lihtsustamiseks lisafailid. Eraldi on olemas ka komponentide kaust, mis sisaldab üldistatud informatsiooni erinevate klasside kohta, mis võimaldab koodiosi taaskasutada. Kasutajal on võimalik vaadete vahel navigeerida, näiteks avavaates kuvatud nupule „*Saved colors*” vajutades suunatakse kasutaja ekraanile, kus on kuvatud kõik kasutaja poolt valitud värvid.

5.2.2 Rakenduse vaated

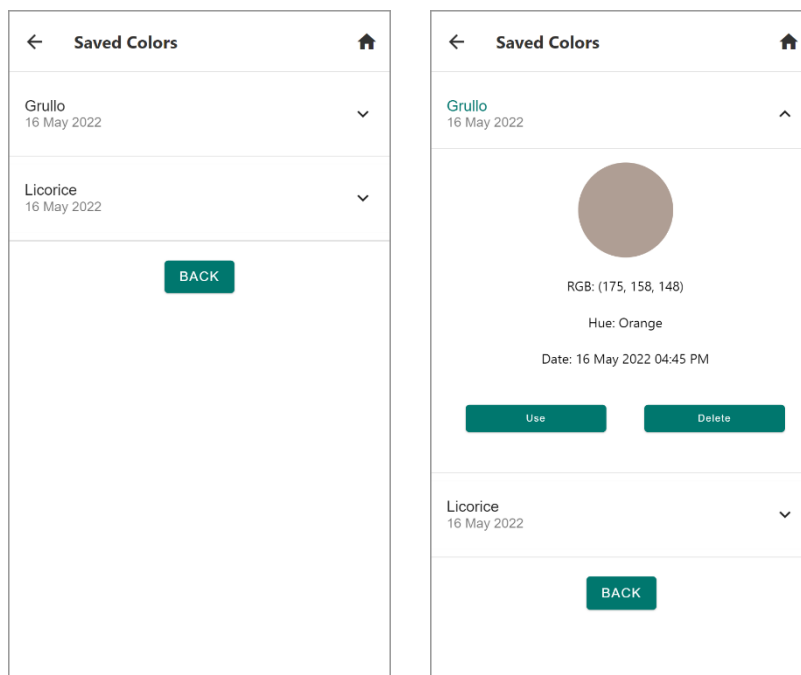
Loodud rakendusel on kuus vaadet:

- Avavaade – avaekraanil on kuvatud menüü, mis võimaldab kasutajal liikuda salvestatud värvide, salvestatud värviharmoniate või pildi valimise vaatesse. Avavaade on kuvatud joonisel 5.



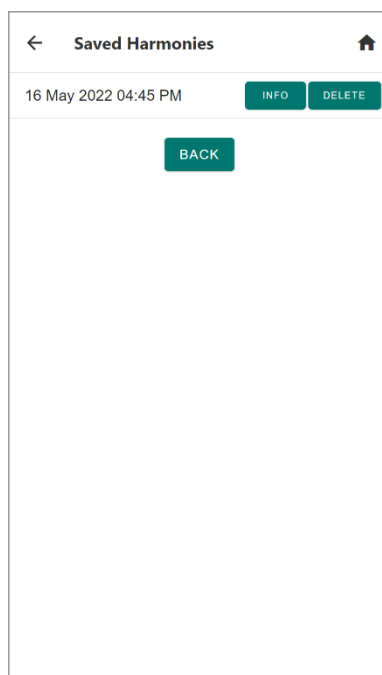
Joonis 5. Mobiilirakenduse avavaade.

- Salvestatud värvide vaade – vaates kuvatakse kasutajale kõik eelnevalt kasutaja poolt salvestatud värvid. Lisaks värvinimele on kuvatud ka RGB-kood, üldine värvitooni nimetus ning värvi salvestamise ajatempel. Salvestatud värvi on võimalik kustutada ning taaskasutada – nende tegevuste läbiviimiseks on loodud nupud. Kustutamise korral eemaldatakse värv tabelist ning juhul kui soovitakse värvi taaskasutada, suunatakse kasutaja pildi valimise ekraanile. Antud vaade on näidatud joonisel 6.



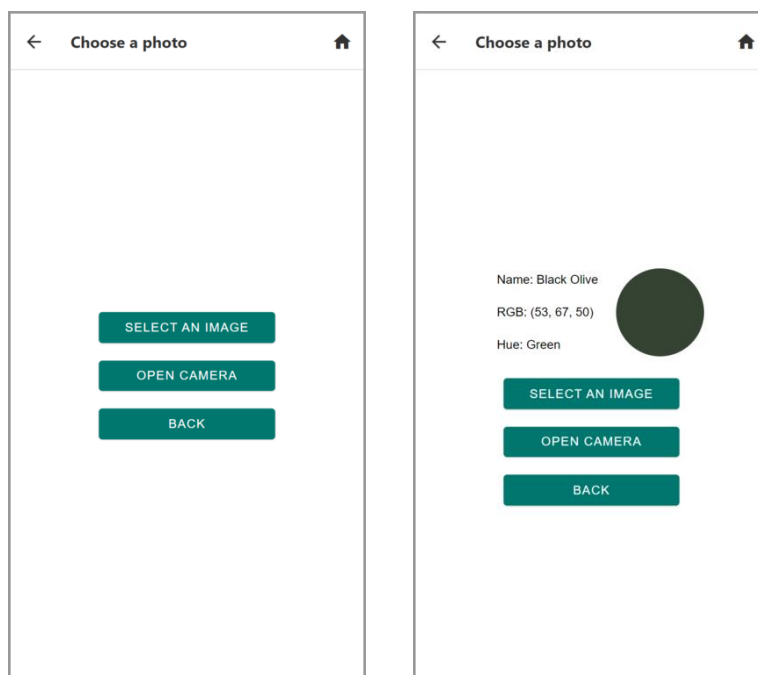
Joonis 6. Mobiilirakenduse salvestatud värvide vaade nimekirjas (vasakul) ja avatuna (paremal).

- Salvestatud värviharmoniade vaade – vaates näidatakse kasutajale nimekirjas kõiki eelnevalt salvestatud värvipalette. Võimalus on vajutada kahele nupule „Info“ ja „Delete“, millest esimene kuvab täpsema värviharmoniade info ja teine eemaldab kirje tabelist. Salvestatud värvide vaade on kujutatud joonisel 7.



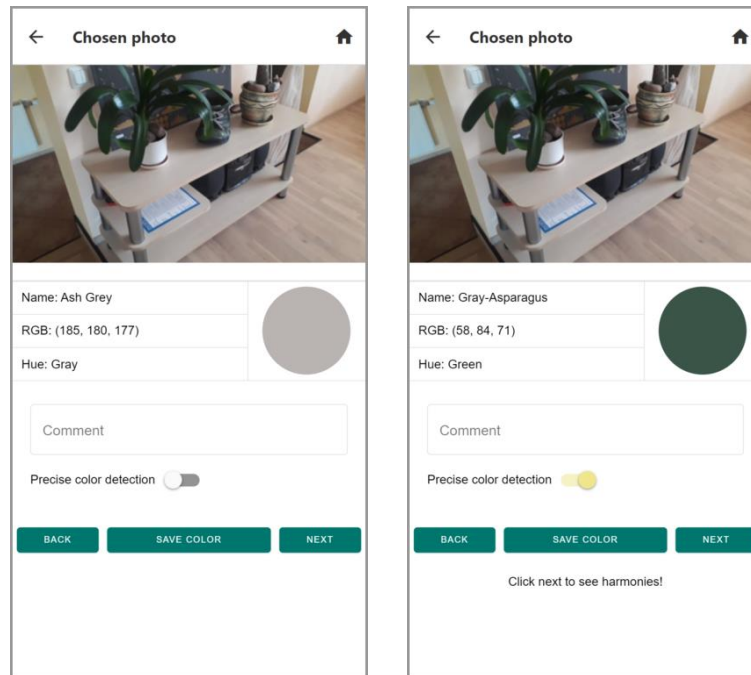
Joonis 7. Mobiilirakenduse salvestatud värviharmoniade vaade.

- Pildi valimise vaade – kasutajal on pilti võimalik valida nii oma lokaalsest pildigaleriist kui ka tehes kaameraga pildi, mille puhul suunatakse kasutaja telefonis vaikimisi valitud pildi tegemise rakendusse ning kui pilt on tehtud, suunatakse kasutaja tagasi. Kohe kui pilt on valitud, liigub rakendus automaatselt järgmisesse vaatesse. Juhul kui enne pildi valimist pidi eelnevalt värvi valida, kuvatakse ekraanil ka valitud värv. Joonisel 8 on kuvatud pildi valimise vaates olukord kui pole valitud veel ühtegi värvi ning olukord, kus on valitud värv.



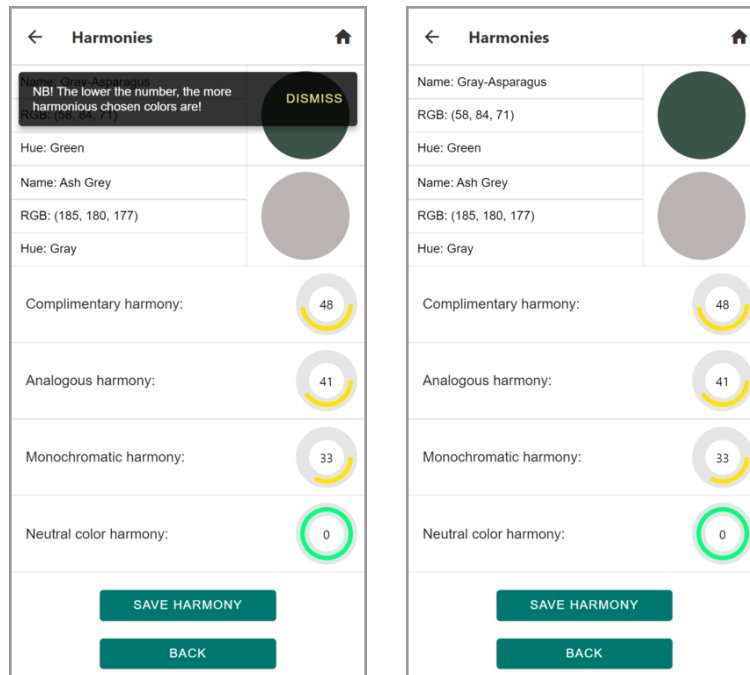
Joonis 8. Mobiilirakenduse pildi valimise vaade enne (vasakul) ja pärast (paremal) esimese värvi valimist.

- Pildi vaade – pildi ekraanil kuvatakse eelnevalt valitud pilt. Vajutades pildil valitud koha peale, saadetakse päring tagarakendusse, mis tagastab vajutuskohal paikneva värvi kohta informatsiooni. Vaates on võimalik ka valida, kas pildilt soovitakse värv valida täpselt piksli järgi või pildil peaksid värvid olema üldistatud. Kui värv on valitud, tekib võimalus vajutada nupule, mis salvestab valitud värvi andmebaasi. Juhul kui värvi valimine toimub esimest korda, ilmub nupp, mis võimaldab liikuda järgmisesse vaatesse ehk teise pildi valimise ekraanile. Kui värvi valimine toimub teist korda suunab nupuvajutus ekraanile, kus leitakse kuidas eelnevalt valitud värvid harmoneeruvad. Värv saab valida lõpmatu arv korra. Pildi vaade on toodud joonisel 9.



Joonis 9. Mobiilirakenduse pildi vaade enne (vasakul) ja pärast (paremal) esimese värvi valimist.

- Harmoneeruvate värvide vaade – vaatele minnes tehakse päring tagarakendusele, mille tulemusena arvutatakse eelnevalt valitud värvide põhjal nende värvide kokkusobivus erinevate värviteooria reeglite põhjal. Juhul kui vaatesse sisenetakse salvestatud harmooniate vaatest, siis harmooniaid ei arvutata, vaid värviharmooniate kohta käiv info päritakse andmebaasist. Värvide kokkusobivus kuvatakse ringdiagrammides, kus on kuvatud teisena valitud värvi kaugus esimese värvi harmoneeruvatest värvidest. Mida väiksem on diagrammis kujutatud number, seda sarnasemad on värvid ehk seda paremini sobivad valitud värvid kokku. Lisaks värviteooria reeglitele on kuvatud ka neutraalsete värvide põhjal kokkusobivuse diagramm, kus 0 tähistab neutraalse värvi olemasolu ehk värvide harmoneerumist ja 100 neutraalvärvide puudumist. Vaates on võimalik ka tagasi eelnevale ekraanile minna, kus saab uue värvi valida. Harmoneeruvate värvide vaade on näidatud joonisel 10.



Joonis 10. Mobiilirakenduse harmoneeruvate värvide hüpikaknaga (vasakul) ja tavaline (paremal) vaade.

5.2.3 Tagarakendusega suhtlus

Klientrakenduses kasutatakse Fetch API't, mis on Javascript'i liides HTTP päringute tegemiseks tagarakendusest andmete pärimiseks. Kui andmeid soovitakse saata tagarakendusse, pannakse andmed JSON kujul päringuga kaasa. Klientrakenduse parema loetavuse tõttu on pandud vaadetes vajalike päringute tegemise loogika eraldi failidesse.

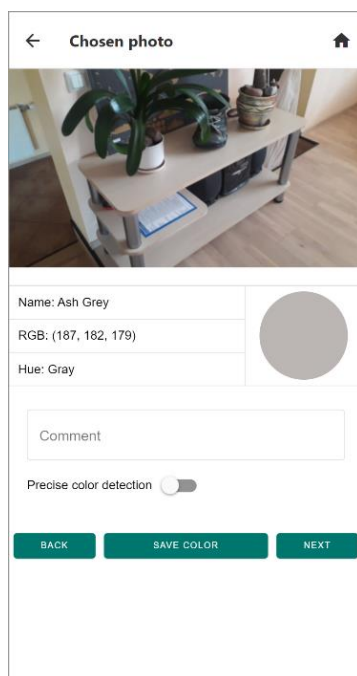
6 Tulemused

Tulemuste peatükk koosneb kolmest osast. Esimeses osas antakse rakendust kõrvalistel isikutel testida ja tehakse muudatused vastavalt tagasisidele. Teises osas kirjeldatakse lahti lõputöö käigus saavutatud tulemused ning luuakse ülevaade mobiilirakenduse kasutamise võimalustest. Kolmandas osas pakub autor välja ideid rakenduse edasiarenduseks.

6.1 Testimine

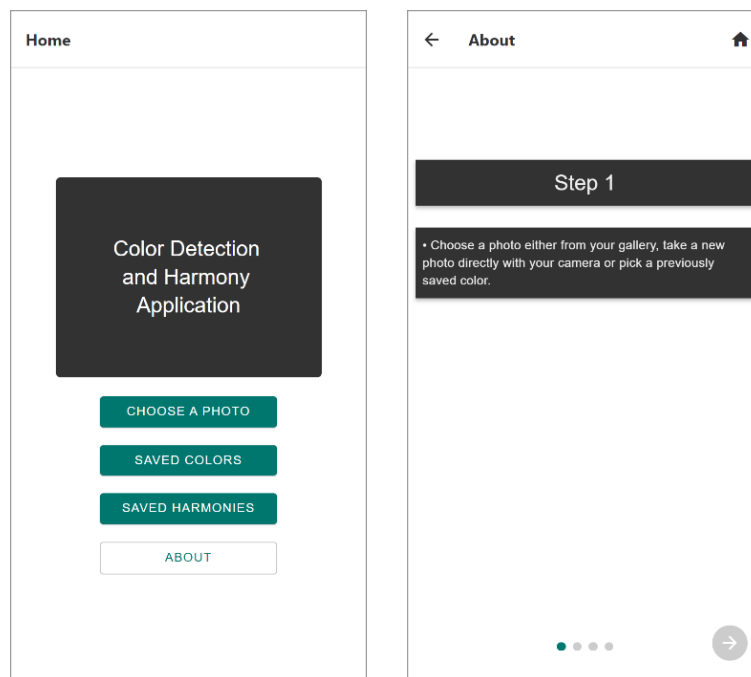
Rakendust anti testida kahel isikul Android telefonil. Tagasisides soovitati lisada järgmised muudatused:

- Lisada võimalus värvidele ja leitud harmooniatele kirjutada juurde kommentaar. Kommentaar teeks kasutajale näiteks selgemaks, mis kohast salvestatud värv pärineb. Rakendusele sai antud funktsionaalsus lisatud ning kommentaari saab ka soovi korral hiljem vastavalt salvestatud värvide või salvestatud harmooniate vaates muuta. Joonisel 11 on näidatud kommentaariväli pildi vaates.



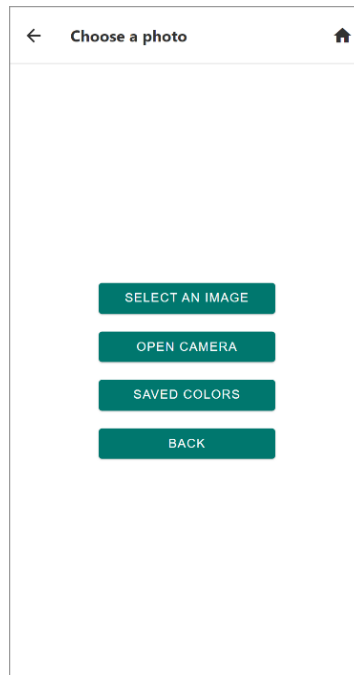
Joonis 11. Värvide salvestamise vaatesse lisatud kommentaariväli.

- Avavaate puhul arvati, et menüünuppudest peaks esimene olema “*Choose a photo*”, sest rakenduse esmakordsel kasutamisel ei ole salvestatud värvide ega harmooniate vaadetes veel andmeid. Muudatus viidi läbi.
- Luua kasutusjuhend, mis selgitaks milliseid tegevusi on rakenduses võimalik teha. Juhend lisati ning see on rakenduses saadaval avaakraani menüüs. Kasutusjuhend on joonisel 12.



Joonis 12. Täiendatud avaakraani menüü (vasakul) ja esimene kasutusjuhendi vaade (paremal).

- Soovitati lisada keele valimise võimalus, sest hetkel on rakendus ainult saadaval inglise keeles. Käesoleva rakenduse tõlkimine on aga väga ajakulukas, sest erinevaid värvuste nimetusi on üle 800. Kuna rakenduse esimese versiooni loomise aeg on piiratud, otsustas autor muudatust käesoleva töö raames mitte teha.
- Testitavas versioonis oli võimalik salvestatud värvi valida ainult vajutades avaakraanil vastavale menüünupule ja sealt võttes värvi, kuid kasutajate soov oli see tegevus teha menüüs, kus toimus muidu pildi valik. Autor lisas selle võimaluse rakendusse. See tähendab, et kasutaja saab nüüd leida värvi harmooniaid pilte valimata juhul kui salvestatud värvid on eelnevalt olemas. Uue menüünupuga vaade on kuvatud joonisel 13.



Joonis 13. Täiendatud pildi valimise menüü.

6.2 Realiseeritud funktsionaalsus

Bakalaureusetöö lõpptulemusena valminud mobiilirakenduses on täidetud kõik algselt püstitatud funktsionaalsed ning mittefunktsionaalsed nõuded. Rakenduse olulisemad meetodid ja vaated on täpsemalt kirjeldatud peatükis “Mobiilirakenduse arendus”. Rakendust anti ka testida kõrvalistel isikutel, mille tulemusena lisati rakendusele juurde uusi võimalusi ning parandati olemasolevat funktsionaalsust.

Programmikoodis funktsioonide juures, kus on kasutatud mõne teise isiku loodut, on viide allikatele toodud asjakohase meetodi kirjelduses.

Loodud rakendus on avalik ja kättesaadaval Github keskkonnas, mis annab võimaluse kõigil lähtekoodi uurida, käivitada ning täiendada.

Rakendus töötab nii veebis kui ka mobiiltelefonil. Juhul kui rakendus soovitakse käivitada Android telefonil, peab esmalt olema allalaetud Expo Go rakendus. Käivitades serveri- ja kliendipoolse rakenduse, kuvatakse QR-kood, mille kaudu saab Expo kasutada värvide tuvastamise ja harmooniate leidmise rakendust. iOS telefonide puhul saab QR-koodi skaneerimiseks kasutada vaikinisi pakutavat kaamerarakendust.

6.3 Edasiarendus

Käesolevas peatükis on kirjeldatud ettepanekud, kuidas oleks võimalik loodud rakendust täiendada.

Rakenduses leitakse värviharmoniaid kahe värvi järgi, kuid tegelikkuses on olemas palju erinevaid esteetilisi värvikoosluseid ja mitmeid värviteooria reegleid, mis põhinevad mitmel värvil ehk kokkusobivust võib ka hinnata kolme, nelja ja enama värvi põhjal. Juhul kui hakataks hindama enam kui kahe värvi harmooniat, saaks liidestada rakendusega ka mõne veebilehe, kuhu lisatakse pidevalt uusi värvipalette. See tähendab, et värviharmoniaid ei pea leidma ainult värviteooria reeglite järgi.

Värviharmonia tuvastamise loogikat saab täiendada võttes arvesse erinevaid värvipimeduse vorme – praegune rakendus arvestab kokkusobivate värvide leidmisel nende inimeste värvitajuga, kel ei esine värvipimedust. Värvipimedad inimesed tajuvad aga lähtuvalt värvipimeduse liigist värvitoone erinevalt ning seetõttu nende jaoks näivad esteetilised hoopis mõnevõrra erinevad värvikooslused. Täiendades aga loodud värviharmoniate tuvastamise loogikat, oleks võimalik leida värvi kombinatsioone, mis näevad head välja värvipimedate inimeste jaoks.

Hetkel on loodud rakendus inglise keeles, kuid et töö jõuaks rohkemate inimesteni ning kasutajamugavust saaks paremaks teha, on võimalik rakendus tõlkida ka erinevatesse keeltesse. Kõige rohkem aega kuluks värvinimede tõlkimisele, sest erinevaid võimalusi on praeguses rakenduses 865. Klientrakenduses saab aga kasutada näiteks React Native'i jaoks loodud raamistikku „react-native-translation“.

Autor leiab, et rakenduse kujundust võiks täiendada. Näiteks kasutusjuhendite ekraanil oleks võimalik lisada juurde erinevaid pilte või ikoone, mis muudaks kasutajale juhendi lugemise meeldivamaks.

Värvide kohta on võimalik lisada juurde rohkem informatsiooni, näiteks lisaks RGB-koodile kuvada värv ka mõnes teises värviformaadis. Uute värvimudelite lisamise vajadust saab aga analüüsida alles peale seda, kui rakendus on laialdasemalt kasutusel ning saades teada, millises valdkonnas on rakenduse vastu kõige suurem huvi.

7 Kokkuvõte

Bakalaureusetöö eesmärk oli luua mobiilirakendus, mis aitab eelkõige värvipimedatel inimestel tuvastada piltidelt värve ning saada teada, kas värvid harmoneeruvad. Eesmärk sai täidetud ehk töö tulemusena loodi eelmainitud funktsionaalsustega rakendus.

Loodud rakenduses on võimalik kasutajal valida enda isiklikust galeriist pilt või luua kaameraga pilt, saada teada valitud pildil olevad värvid, salvestada värvid ning tuvastada, kas eelnevalt valitud värvid sobivad erinevate värviteooria reeglite järgi kokku. Kuna rakendust anti testida, sai ka funktsionaalsust täiendada lähtudes kasutajate soovidest.

Esialgne taustauuring andis ettekujutuse rakenduse loomise võimalustest. Rakenduse analüüs aitas valida loodava töö jaoks kõige paremad tehnoloogiad ja planeerida rakenduse üldist arhitektuuri.

Töö käigus õppis autor palju mobiilirakenduse arendusest, sest varasem kokkupuude antud valdkonnas oli minimaalne. Samuti sai autor tutvuda lähemalt suurema Python projekti loomisega – enne lõputöö tegemist oldi tehtud ainult väiksemahulisi Python'i projekte, suuremate projektide tegemiseks oli autor kasutanud teisi programmeerimiskeeli.

Kasutatud kirjandus

- [1] Inimene, *Värvipimedus*, [Online]. Loetud aadressil: <https://www.inimene.ee/haigused-ja-seisundid/list/haigused-ja-seisundid/varvipimedus/>. Kasutatud: 23.04.2022.
- [2] Colour Blind Awareness, *Types of Colour Blindness*, [Online]. Loetud aadressil: <https://www.colourblindawareness.org/colour-blindness/types-of-colour-blindness/>. Kasutatud: 23.04.2022.
- [3] I. Trofimova, „Värvipimesus“, *Kliinik*, [Online]. Loetud aadressil: https://www.kliinik.ee/haiguste_abc/varvipimesus/id-2071. Kasutatud: 23.04.2022.
- [4] M. Tammert, *Värviõpetus teoorias*. Tallinn: Tallinna Tehnikakõrgkool, OÜ Aimwell, 2002.
- [5] A. Rinde, „Multimeedium, värvid“, *Tallinna Ülikool Digitehnoloogiaste Instituut*, [Online]. Loetud aadressil: http://www.cs.tlu.ee/~rinde/mm_materjal/pdf/mm_color.pdf. Kasutatud: 23.04.2022.
- [6] M. Tammert, „Värvikombinatoorika CMYK-i baasil“, *Värviõpetuse teooria*, 2013, [Online]. Loetud aadressil: http://opiobjektid.tptlive.ee/Varviop/VT_Varvikujundus2.htm. Kasutatud: 23.04.2022.
- [7] K. Smith, “Creating Color Harmony”, *Sensational Color*, [Online]. Loetud aadressil: <https://www.sensationalcolor.com/creating-color-harmony/>. Kasutatud: 23.04.2022.
- [8] Sand Art, *Värvuste parem sobivus. 20 värvust valgest mustani!*, 2021, [Online]. Loetud aadressil: <http://sandart.ee/varvid>. Kasutatud: 23.04.2022.
- [9] M. Tammert, „Värvikorrastus“, *Värviõpetuse teooria*, 2013, [Online]. Loetud aadressil: http://opiobjektid.tptlive.ee/Varviop/VT_Varvikorrastus.htm. Kasutatud: 01.05.2022.
- [10] R. Tunnel, „7.4 Silmaring: Arvutigraafika II. Värviruumid ja -mudelid“, *Arvutiteaduse instituudi kursused*, 2015, [Online]. Loetud aadressil: <https://courses.cs.ut.ee/2015/eprogalused/Main/Silmaring-varviruumid>. Kasutatud: 01.05.2022.

- [11] L. Verou, "LCH colors in CSS: what, why, and how?", *Lea Verou – Life at the bleeding edge (of web standards)*, 2020, [Online]. Loetud aadressil: <https://lea.verou.me/2020/04/lch-colors-in-css-what-why-and-how/>. Kasutatud: 01.05.2022.
- [12] T. Riemersma, „Colour metric“, *CompuPhase*, 2019, [Online]. Loetud aadressil: <https://www.compuphase.com/cmetric.htm>. Kasutatud: 23.04.2022.
- [13] X-Rite, *A guide to Understanding Color Tolerancing*, 2015, [Online]. Loetud aadressil: https://web.archive.org/web/20151010064956/http://www.xrite.com/documents/literature/en/L10-024_Color_Tolerance_en.pdf. Kasutatud: 23.04.2022.
- [14] Z. Schuessler, „Delta E 101“, *Github*, 2020, [Online]. Loetud aadressil: <http://zschuessler.github.io/DeltaE/learn/>. Kasutatud: 23.04.2022.
- [15] J. Qian, „A simple review of CIE ΔE^* (color difference) equations“, *Techkon*, 2018, [Online]. Loetud aadressil: <https://techkonusa.com/a-simple-review-of-cie-delta-e-color-difference-equations/>. Kasutatud: 23.04.2022.
- [16] B. J. Lindbloom, „Useful Color Equations“, *Bruce Lindbloom*, 2013, [Online]. Loetud aadressil: <http://www.brucelindbloom.com/index.html?Math.html>. Kasutatud: 11.05.2022.
- [17] R. Kulhary, „OpenCV – Overview“, *GeeksForGeeks*, 2021, [Online]. Loetud aadressil: <https://www.geeksforgeeks.org/opencv-overview/>. Kasutatud: 23.04.2022.
- [18] OpenCV, *About*, 2022, [Online]. Loetud aadressil: <https://opencv.org/about/>. Kasutatud: 23.04.2022.
- [19] M. Chaddha, „C++ vs Java vs Python: Which One To Choose?“, 2021, [Online]. Loetud aadressil: <https://www.codingninjas.com/blog/2021/07/29/c-vs-java-vs-python-which-one-to-choose/>. Kasutatud: 23.04.2022.
- [20] Simplilearn, *C++ Vs. Python: Overview, Uses & Key Differences*, 2021, [Online]. Loetud aadressil: <https://www.simplilearn.com/tutorials/cpp-tutorial/cpp-vs-python>. Kasutatud: 23.04.2022.
- [21] B. Weber, „MATLAB vs Python: Why and How to Make the Switch“, [Online]. Loetud aadressil: <https://realpython.com/matlab-vs-python/>. Kasutatud: 23.04.2022.
- [22] J. Patel, „7 Top Python Frameworks You Should Consider For Your Web Project“, *Monocubed*, 2022, [Online]. Loetud aadressil: <https://www.monocubed.com/blog/top-python-frameworks/>. Kasutatud: 23.04.2022.

- [23] Benchmark, *Let's Understand The Pros And Cons Of Using Django*, [Online]. Loetud address: <https://www.benchmarkit.solutions/lets-understand-the-pros-and-cons-of-using-django/>. Kasutatud: 23.04.2022.
- [24] M. Deery, „What Is Flask and How Do Developers Use It? A Quick Guide“, *CareerFoundry*, 2021, [Online]. Loetud addressil: <https://careerfoundry.com/en/blog/web-development/what-is-flask/>. Kasutatud: 23.04.2022.
- [25] S. Kakarla, „The Top 10 Python Frameworks For Web Development“, *ActiveState*, 2022, [Online]. Loetud addressil: <https://www.activestate.com/blog/the-top-10-python-frameworks-for-web-development/>. Kasutatud: 23.04.2022.
- [26] Slant, *CherryPy Review*, [Online]. Loetud addressil: <https://www.slant.co/options/1740/~cherrypy-review>. Kasutatud: 23.04.2022.
- [27] Slant, *Falcon Review*, [Online]. Loetud addressil: <https://www.slant.co/options/1744/~falcon-review>. Kasutatud: 23.04.2022.
- [28] P. Rospel, „Andmebaasi mootor. Ajalooline ja klient / teenindaja arhitektuur“, *enos.itcollege.ee*, 2011, [Online]. Loetud addressil: https://enos.itcollege.ee/~priit/1.%20Andmebaasid/1.%20Loengumaterjalid/03/69678880001_3-4.htm. Kasutatud: 23.04.2022.
- [29] S. Rubabe, „ORM for Python“, *Medium*, 2020, [Online]. Loetud addressil: <https://medium.com/pragmatech/orm-for-python-b63cfbc39e7f>. Kasutatud: 23.04.2022.
- [30] SQLAlchemy, *SQLite*, [Online]. Loetud addressil: <https://docs.sqlalchemy.org/en/14/dialects/sqlite.html>. Kasutatud: 23.04.2022.
- [31] Technostacks, „Most Popular Mobile App Development Frameworks For App Developers“, 2021, [Online]. Loetud addressil: <https://technostacks.com/blog/mobile-app-development-frameworks>. Kasutatud: 23.04.2022.
- [32] F. Hasan, „What Are The Best Frameworks Available For Hybrid App Development?“, *Maven Cluster*, 2021, [Online]. Loetud addressil: <https://www.mavencluster.com/blog/top-6-hybrid-app-development-frameworks/>. Kasutatud: 23.04.2022.
- [33] IBM Cloud Education, *REST APIs*, 2021, [Online]. Loetud addressil: <https://www.ibm.com/in-en/cloud/learn/rest-apis>. Kasutatud: 23.04.2022.

[34] Microsoft, *CSV-failide Outlooki importimiseks loomine või redigeerimine*, [Online].

Loetud aadressil: <https://support.microsoft.com/et-ee/office/csv-failide-outlooki-importimiseks-loomine-voi-redigeerimine-4518d70d-8fe9-46ad-94fa-1494247193c7>. Kasutatud: 23.04.2022.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

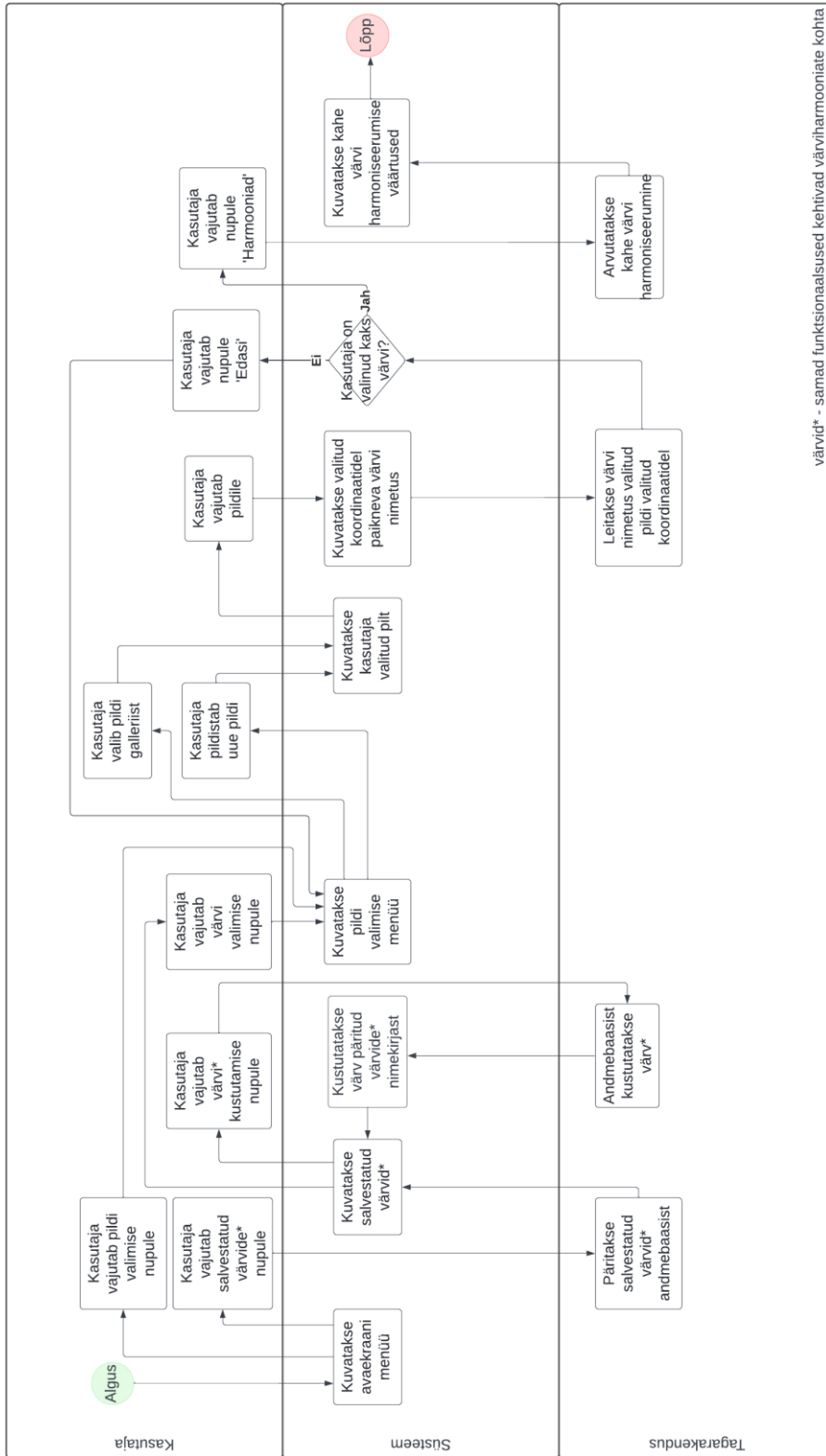
Mina, Gerlin Vainomäe

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Mobiilirakenduse loomine harmoneeruvate värvide leidmiseks värviteooria alusel”, mille juhendaja on Meelis Antoi
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2022

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtjaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Voodiagramm



Lisa 3 – ΔE_{CMC}

$$\Delta E_{CMC} = \sqrt{\left(\frac{\Delta L}{lS_L}\right)^2 + \left(\frac{\Delta C}{cS_C}\right)^2 + \left(\frac{\Delta H}{S_H}\right)^2}$$

$$\Delta C = C_1 - C_2$$

$$C_1 = \sqrt{a_1^2 + b_1^2}$$

$$C_2 = \sqrt{a_2^2 + b_2^2}$$

$$\Delta H = \sqrt{\Delta a^2 + \Delta b^2 + \Delta C^2}$$

$$\Delta L = L_1 - L_2$$

$$\Delta a = a_1 - a_2$$

$$\Delta b = b_1 - b_2$$

$$S_L = \begin{cases} 0.511 & L_1 < 16 \\ \frac{0.040975L_1}{1 + 0.01765L_1} & L_1 \geq 16 \end{cases}$$

$$S_C = \frac{0.0638C_1}{1 + 0.0131C_1} + 0.638$$

$$S_H = S_C(FT + 1 - F)$$

$$T = \begin{cases} 0.56 + |0.2\cos(H_1 + 168^\circ)| & 164^\circ < H_1 < 345^\circ \\ 0.36 + |0.4\cos(H_1 + 35^\circ)| & 164^\circ \geq H_1 \vee 345^\circ \geq H_1 \end{cases}$$

$$F = \sqrt{\frac{C_1^4}{C_1^4 + 1900}}$$

$$H = \arctan\left(\frac{b_1}{a_1}\right)$$

$$H_1 = \begin{cases} H & H \geq 0 \\ H + 360^\circ & H < 0 \end{cases}$$

Lisa 4 – ΔE_{94}

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L}{K_L S_L}\right)^2 + \left(\frac{\Delta C}{K_C S_C}\right)^2 + \left(\frac{\Delta H}{K_H S_H}\right)^2}$$

$$\Delta L = L_1 - L_2$$

$$\Delta C = C_1 - C_2$$

$$\Delta H = \sqrt{\Delta a^2 + \Delta b^2 + \Delta C^2}$$

$$C_1 = \sqrt{a_1^2 + b_1^2}$$

$$C_2 = \sqrt{a_2^2 + b_2^2}$$

$$\Delta a = a_1 - a_2$$

$$\Delta b = b_1 - b_2$$

$$S_L = 1$$

$$S_C = 1 + K_1 C_1$$

$$S_H = 1 + K_2 C_1$$

$$K_L = \begin{cases} 1 & \text{vaikimisi} \\ 2 & \text{tekstiilid} \end{cases}$$

$$K_C = 1$$

$$K_H = 1$$

$$K_1 = \begin{cases} 0.045 & \text{graafika} \\ 0.048 & \text{tekstiilid} \end{cases}$$

$$K_2 = \begin{cases} 0.015 & \text{graafika} \\ 0.014 & \text{tekstiilid} \end{cases}$$

Lisa 5 – ΔE_{2000}

$$\Delta E_{2000} = \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right) \left(\frac{\Delta H'}{K_H S_H}\right)}$$

$$\bar{L}' = (L_1 + L_2) / 2$$

$$C_1 = \sqrt{a_1^2 + b_1^2}$$

$$C_2 = \sqrt{a_2^2 + b_2^2}$$

$$\bar{C} = (C_1 + C_2) / 2$$

$$G = \frac{1}{2} \left(1 - \sqrt{\frac{C^{-7}}{C^{-7} + 25^7}} \right)$$

$$a'_1 = a_1(1 + G)$$

$$a'_2 = a_2(1 + G)$$

$$C'_1 = \sqrt{a'^2_1 + b_1^2}$$

$$C'_2 = \sqrt{a'^2_2 + b_2^2}$$

$$\bar{C}' = (C'_1 + C'_2) / 2$$

$$h'_1 = \begin{cases} \arctan(b_1 / a'_1) & \arctan(b_1 / a'_1) \geq 0 \\ \arctan(b_1 / a'_1) + 360^\circ & \arctan(b_1 / a'_1) < 0 \end{cases}$$

$$h'_2 = \begin{cases} \arctan(b_2 / a'_2) & \arctan(b_2 / a'_2) \geq 0 \\ \arctan(b_2 / a'_2) + 360^\circ & \arctan(b_2 / a'_2) < 0 \end{cases}$$

$$\bar{H}' = \begin{cases} (h'_1 + h'_2 + 360^\circ) / 2 & |h'_1 - h'_2| > 180^\circ \\ (h'_1 + h'_2) / 2 & |h'_1 - h'_2| \leq 180^\circ \end{cases}$$

$$T = 1 - 0.17 \cos(\bar{H}' - 30^\circ) + 0.24 \cos(2\bar{H}') + 0.32 \cos(3\bar{H}' + 6^\circ) - 0.20 \cos(4\bar{H}' - 63^\circ)$$

$$\Delta h' = \begin{cases} h'_2 - h'_1 & |h'_2 - h'_1| \leq 180^\circ \\ h'_2 - h'_1 + 360 & |h'_2 - h'_1| > 180^\circ \wedge h'_2 \leq h'_1 \\ h'_2 - h'_1 - 360 & |h'_2 - h'_1| > 180^\circ \wedge h'_2 > h'_1 \end{cases}$$

$$\Delta L' = L_2 - L_1$$

$$\Delta C' = C_2' - C_1'$$

$$\Delta H' = 2\sqrt{C_1' C_2'} \sin(\Delta h' / 2)$$

$$S_L = 1 + \frac{0.015(\bar{L}' - 50)^2}{\sqrt{20 + (\bar{L}' - 50)^2}}$$

$$S_C = 1 + 0.045\bar{C}'$$

$$S_H = 1 + 0.015\bar{C}'T$$

$$\Delta\theta = 30 \exp\left\{-\left(\frac{\bar{H}' - 275^\circ}{25}\right)^2\right\}$$

$$R_C = 2 \sqrt{\frac{C^{-7'}}{C^{-7'} + 25^7}}$$

$$R_T = -R_C \sin(2\Delta\theta)$$

$$K_L = 1$$

$$K_C = 1$$

$$K_H = 1$$

Lisa 6 – Mobiilirakenduse kättesaadavus

Kliendipoolne rakendus:

<https://github.com/gerlinv/color-detection-and-harmony-application-frontend>

Serveripoolne rakendus:

<https://github.com/gerlinv/color-detection-and-harmony-application-backend>