Door: Annelieke van der kwast

Datum: 20-03-2024

Klas: SD-21

ID Scan Demo Examen

Inhoudsopgave

[Inleiding 2](#_Toc165043259)

[Eisen voor het project 2](#_Toc165043260)

[User Stories 4](#_Toc165043261)

[Technische ontwerpen 10](#_Toc165043262)

[Activity diagram 10](#_Toc165043263)

[Use-Case Diagram 11](#_Toc165043264)

[Class diagram 11](#_Toc165043265)

[Design 12](#_Toc165043266)

[Code documentatie 19](#_Toc165043267)

[Sprint 1 19](#_Toc165043268)

[Sprint 2 25](#_Toc165043269)

[Sprint 3 32](#_Toc165043270)

[Code feedback 36](#_Toc165043271)

[Week 13 code feedback 36](#_Toc165043272)

[Week 14 code feedback 36](#_Toc165043273)

[Week 15 code feedback 36](#_Toc165043274)

[Week 17 code feedback 37](#_Toc165043275)

[Testplan 38](#_Toc165043276)

[Verbetervoorstellen 42](#_Toc165043277)

[Verbetervoorstellen product en proces week 13 42](#_Toc165043278)

[Verbetervoorstellen product en proces week 14 42](#_Toc165043279)

[Verbetervoorstellen product en proces week 15 42](#_Toc165043280)

[Verbetervoorstellen product en proces week 16 43](#_Toc165043281)

[Verbetervoorstellen product en proces week 17 43](#_Toc165043282)

[Sprint review + retrospectief 44](#_Toc165043283)

[Sprint review + retrospectief 2-04-2024 44](#_Toc165043284)

[Sprint review + retrospectief 15-04-2024 45](#_Toc165043285)

[Conclusie 47](#_Toc165043286)

# Inleiding

In het kader van mijn stage bij Siip, werk ik aan het uitbreiden van hun ID scan demo app. Die app laat zien hoe Siip het onboardings proces uitvoert. Dit proces bevat het verzamelen van gegevens met behulp van een paspoort, identiteitsbewijs of rijbewijs en daarna laat die alle door de AVG goed gekeurde gegevens zien.

Mijn examenopdracht is dan ook gericht op het uitbreiden van de huidige ID scan demo app, met de focus op de verandering van de appkleuren.

Dit houdt in dat ik de app zodanig aanpas dat onze verkoopmedewerkers

de kleuren van de app kunnen wijzigen en daarnaast het icoon/logo en de achtergrondfoto kunnen aanpassen.

De verkoopmedewerker kan bijvoorbeeld voordat die op gesprek gaat met PostNL het logo naar PostNL veranderen en de app helemaal oranje maken.

Deze aanpasmogelijkheden kan verkoopmedewerkers helpen om het bedrijf/organisatie of vereniging waar ze op bezoek gaan positief te beïnvloeden en voor Siip te laten kiezen. Omdat zij als bedrijf/organisatie of vereniging direct een voorproefje krijgen van hoe de app eruit zal zien in hun eigen huisstijl.

Mijn werk bij Siip aan deze opdracht was ook het ontwerpen van een overzichtelijke user interface (UI). Maar nu tijdens mijn examen is het de bedoeling dat ik de ontwerpen in de huidige app ga realiseren.

In het hoofdstuk “Eisen voor het project” kunt u dan ook verder lezen over de items die in de app moeten voor mijn examenopdracht, en de items die er na mijn examenopdracht nog in gaan komen.

## Eisen voor het project

Voor de eisen van dit project gebruik ik de MoSCoW-methode dit is een methode die verschillende soorten prioriteiten stelt aan het project.

De must haves, deze eisen moeten in het eindresultaat terugkomen, zonder deze eisen is het product niet bruikbaar.

De should haves, deze eisen zijn zeer gewenst, maar zonder is het product wel bruikbaar.

En de could haves, deze eisen zullen alleen aan bod komen als er tijd genoeg is.

De won’t haves heb ik uit dit project gehaald omdat ze niet van toepassing zijn.

Het project moet aan de volgende dingen voldoen:

* Must have,

Je moet als verkoop medewerker de kleuren van de app kunnen aanpassen. Ook als de kleuren dan zijn aangepast moet je de app helemaal kunnen afsluiten en weer opstarten in de gekozen kleuren.

* Should have,

Is ook nog dat je het icoon/logo van de app kan veranderen en de achtergrond foto.

* Could have,

Een mini telefoon die als voorbeeld aangeeft hoe de app er uit komt te zien.

# User Stories

Hier onder staan alle user stories waar ik in het project mee te maken krijg en aan moet voldoen.

|  |  |  |
| --- | --- | --- |
| 1. Titel: Laadscherm | Prioriteit: gemiddeld | Verwachte tijd: 60 minuten |
| User Story:  Als een gebruiker/ verkoopmedewerker wil ik het intro scherm te zien krijgen als ik de app opstart. |  |  |
| Acceptatiecriteria:  Als de laadschermen in correcte volgorde in beeld komen. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Onboarding start | Prioriteit: hoog | Verwachte tijd: 5 minuten |
| User Story:  Als een gebruiker/ verkoopmedewerker wil ik bij het onboarding processen kunnen kiezen uit de 3 onboarding mogelijkheden (ID, paspoort en rijbewijs) en de onboarding kunnen starten. |  |  |
| Acceptatiecriteria:  Wanneer je uit de 3 onboarding types kan kiezen. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Onboarding start, scan je paspoort, id of rijbewijs | Prioriteit: hoog | Verwachte tijd: 5 minuten |
| User Story:  Als een gebruiker/ verkoopmedewerker wil ik bij het onboarding processen kunnen starten en mijn paspoort, ID of rijbewijs kunnen scannen. |  |  |
| Acceptatiecriteria:  Wanneer je, je rijbewijs, paspoort of ID kan scannen. |  |  |
| 1. Titel: Na het onboarding | Prioriteit: hoog | Verwachte tijd: 5 minuten |
| User Story:  Als een gebruiker/ verkoopmedewerker wil ik mijn deelbare gegevens zien, en zien dat het onboarden gelukt is. |  |  |
| Acceptatiecriteria:  Wanneer je, je gegevens kan inzien. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Onboarding start tandwiel | Prioriteit: hoog | Verwachte tijd: 120 minuten |
| User Story:  Als een verkoopmedewerker wil ik op het tandwiel kunnen drukken die op het onboarding startscherm staat. Om het thema van de app aan te kunnen passen. |  |  |
| Acceptatiecriteria:  Wanneer je op het tandwiel kan klikken en je bij het thema kiezen scherm terecht komt. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema kiesscherm, kiezen | Prioriteit: hoog | Verwachte tijd: 240 minuten |
| User Story:  Als een verkoopmedewerker wil ik op het thema kiesscherm kunnen kiezen uit bestaande thema’s. |  |  |
| Acceptatiecriteria:  Wanneer uit bestaande thema’s worden gekozen. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema kiesscherm, Nieuw thema aanmaken | Prioriteit: hoog | Verwachte tijd: 720 minuten |
| User Story:  Als een verkoopmedewerker wil ik op het thema kiesscherm nieuwe thema’s kunnen aanmaken door op een knop te klikken. |  |  |
| Acceptatiecriteria:  Wanneer je nieuwe thema’s kan aanmaken en opslaan. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema kiesscherm, Bewerken | Prioriteit: gemiddeld/laag | Verwachte tijd: 420 minuten |
| User Story:  Als een verkoopmedewerker wil ik op het thema kiesscherm thema’s kunnen bewerken door op een pen/kwast icoon te klikken. |  |  |
| Acceptatiecriteria:  Wanneer je bestaande thema’s kan verwijderen, behalve Siip thema default. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema kiesscherm, verwijderen | Prioriteit: gemiddeld/laag | Verwachte tijd: 60 minuten |
| User Story:  Als een verkoopmedewerker wil ik op het thema kiesscherm thema’s kunnen verwijderen door op een prullenbak icoon te kunnen klikken. |  |  |
| Acceptatiecriteria:  Wanneer je bestaande thema’s kan verwijderen, behalve Siip thema default. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema maken stappen voor het logo | Prioriteit: gemiddeld | Verwachte tijd: 120 minuten |
| User Story:  Als een verkoopmedewerker wil ik een logo kunnen uploaden van de club, vereniging of bedrijf waar ik op bezoek ga. |  |  |
| Acceptatiecriteria:  Wanneer je een logo kan uploaden en naar de volgende stap kan. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema maken stappen voor de achtergrond | Prioriteit: gemiddeld | Verwachte tijd: 600 minuten |
| User Story:  Als een verkoopmedewerker wil ik een bijpassende achtergrond kunnen uploaden van de club, vereniging of bedrijf waar ik op bezoek ga. |  |  |
| Acceptatiecriteria:  Wanneer je een bijpassende achtergrond kan uploaden en naar de volgende stap kan. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema maken stappen voor de kleuren | Prioriteit: hoog | Verwachte tijd: 420 minuten |
| User Story:  Als een verkoopmedewerker wil ik de bijpassende kleuren kunnen kiezen die van de club, vereniging of bedrijf zijn waar ik op bezoek ga. |  |  |
| Acceptatiecriteria:  Wanneer je de bijpassende kleuren kan kiezen en naar de volgende stap kan. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema maken stappen voor het lettertype | Prioriteit: gemiddeld | Verwachte tijd: 120 minuten |
| User Story:  Als een verkoopmedewerker wil ik een bijpassend lettertype kunnen kiezen die van de club, vereniging of bedrijf zijn waar ik op bezoek ga. |  |  |
| Acceptatiecriteria:  Wanneer je een bijpassend lettertype kan kiezen en naar de volgende stap kan. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema maken stappen voor het opslaan | Prioriteit: hoog | Verwachte tijd: 120 minuten |
| User Story:  Als een verkoopmedewerker wil ik mijn zelfgemaakt thema een naam kunnen geven en opslaan op mijn telefoon. |  |  |
| Acceptatiecriteria:  Wanneer je het thema kan opslaan op je telefoon. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Thema maken stappen voor het voorbeeld | Prioriteit: laag | Verwachte tijd: 720 minuten |
| User Story:  Als een verkoopmedewerker wil ik een voorbeeld in het klein zien van de app met de gekozen kleuren, lettertype, logo en achtergrond. |  |  |
| Acceptatiecriteria:  Wanneer het kleine voorbeeld overeenkomt met het ontwerp. |  |  |

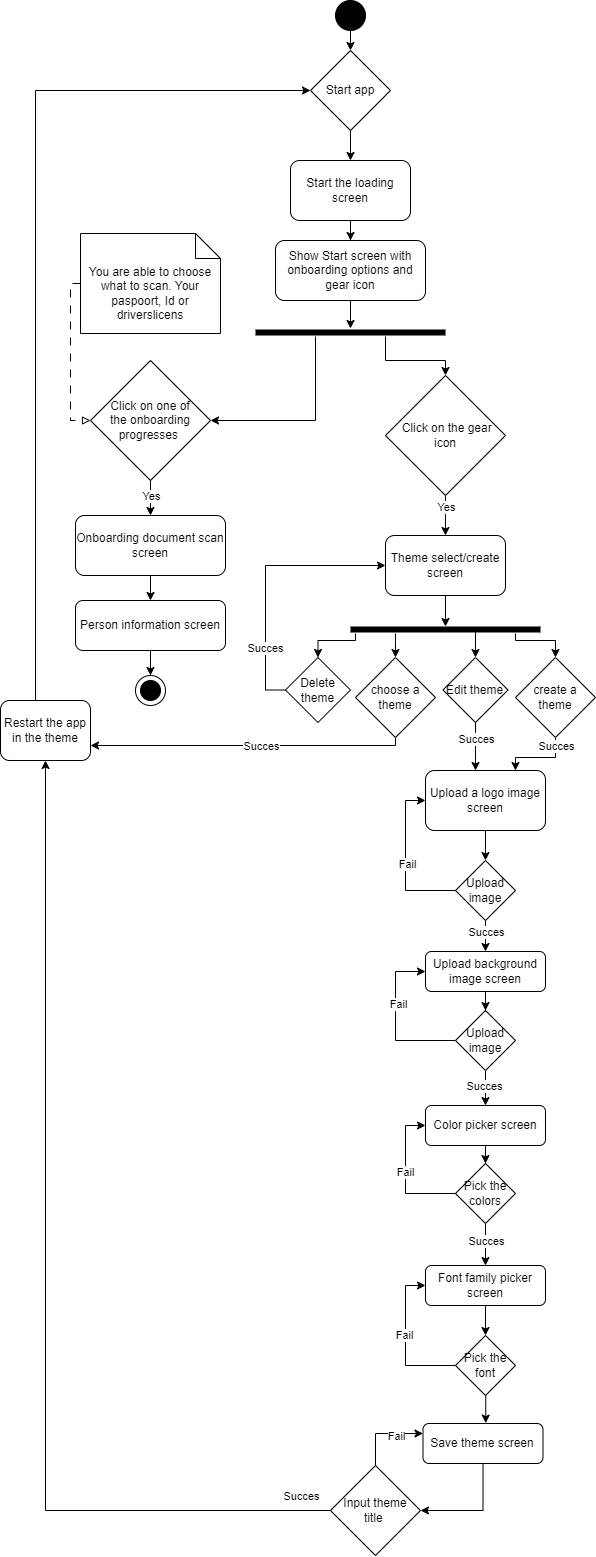
|  |  |  |
| --- | --- | --- |
| 1. Titel: Error | Prioriteit: gemiddeld/laag | Verwachte tijd: 120 minuten |
| User Story:  Als een gebruiker/verkoopmedewerker wil ik een duidelijk scherm die mij laat zien als er wat fout gaat in de app. En wat er fout gaat in de app. |  |  |
| Acceptatiecriteria:  Wanneer je een pop-up te zien krijgt van wat er fout gaat in de app. |  |  |

|  |  |  |
| --- | --- | --- |
| 1. Titel: Uiterlijk app | Prioriteit: hoog | Verwachte tijd: 600 minuten |
| User Story:  Als een gebruiker/verkoopmedewerker wil ik dat de app er goed uitziet. |  |  |
| Acceptatiecriteria:  Wanneer je de app er hetzelfde uitziet als het ontwerp. |  |  |

# Technische ontwerpen

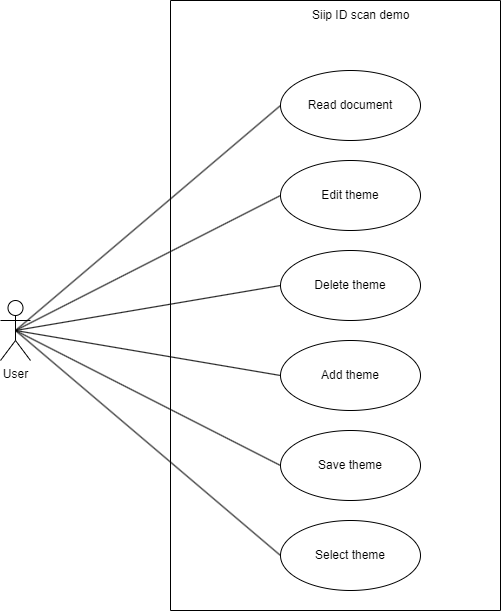
Alle technische ontwerpen zijn in het Engels omdat dat bij mijn stage bedrijf wordt verwacht.

## Activity diagram

In deze activity diagram kan je zien hoe het scherm verloop gaat. Dit is van start tot het einde.

## Use-Case Diagram

Dit geeft aan wat de gebruiker voor een acties allemaal kan aanpassen en zien.



## Class diagram

# Design

Hieronder kunt u het design van de door mij ontworpen extra functies zien. Dit is dan ook de nice to have, als alles zo zou gaan werken als in het ontwerp. Maar in dit examen gaat het niet om het uiterlijk. Maar het gaat er wel om dat de kleuren-kiezer kan worden gebruikt om het uiterlijk van de app aan te passen. Alsnog krijg je op deze manier een idee hoe de flow van de app moet gaan lopen;

1. Opstart/laadscherm (zichtbaar voor +/-4 seconden):

Afbeelding met tekst, schermopname, Elektrisch blauw, logo

Automatisch gegenereerde beschrijving

1. Landings/home scherm:

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

1. Nadat je op het tandwiel icoon hebt geklikt kom je op de thema keuze pagina:

Afbeelding met tekst, schermopname, nummer, Lettertype

Automatisch gegenereerde beschrijving

1. Hier onder zie je de stappen voor als je op de ‘Nieuw thema maken’ knop hebt geklikt, de stappen gaan van links naar rechts:

Afbeelding met tekst, schermopname, Lettertype, diagram

Automatisch gegenereerde beschrijvingAfbeelding met schermopname, tekst, ontwerp

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

1. Error scherm mocht er iets fout gaan tijdens het uploaden van een foto:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

1. Daarna ga je door naar het uploaden van een achtergrond:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijvingAfbeelding met schermopname, tekst, ontwerp

Automatisch gegenereerde beschrijvingAfbeelding met tekst, voetbal, schermopname, bal

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, logo, Merk

Automatisch gegenereerde beschrijving

1. Error schermen uploaden van een achtergrond:

Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

1. Afbeelding met tekst, schermopname, ontwerp

   Automatisch gegenereerde beschrijvingAfbeelding met tekst, elektronica, schermopname, multimedia

   Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, ontwerp

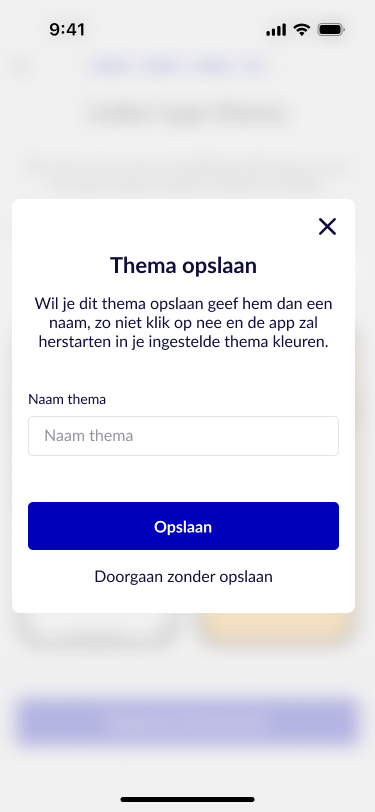
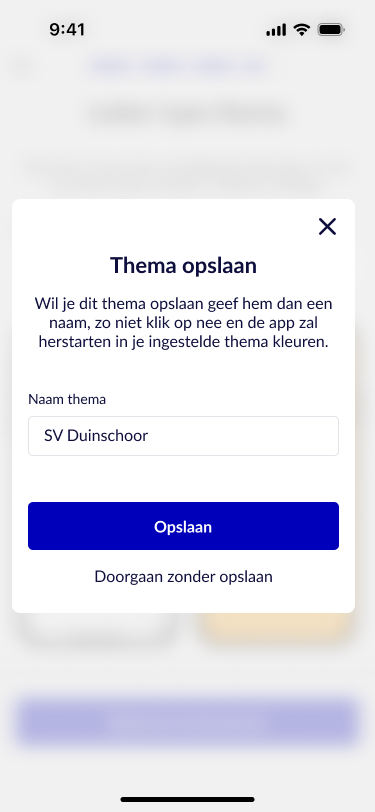
   Automatisch gegenereerde beschrijvingNa het toevoegen van de achtergrond komt het belangrijkst en de must have van dit examen, de kleuren kiezer:
2. Na het kleuren kiezen kan je ook nog een lettertype instellen:

Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijving

1. En als laatste het opslaan van je nieuwe thema:



1. Error tijdens het opslaan:

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Als u nu weer terug gaat naar het thema keuze pagina dan zie je ook een prullenbak icoon. Als je daar op zou drukken zal de volgende schermen reeks zich starten.

1. Afbeelding met tekst, schermopname, Lettertype, ontwerp

   Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, nummer

   Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, nummer

   Automatisch gegenereerde beschrijving De schermen reeks voor het verwijderen van een thema:
2. Afbeelding met tekst, schermopname, Lettertype, ontwerp

   Automatisch gegenereerde beschrijvingError scherm, tijdens het verwijderen ging wat verkeerd:

Het bewerken van een thema ziet er op zich hetzelfde uit als een thema aanmaken.

Alleen zijn de plaatjes, de kleuren en het lettertype al ingevuld.

Waardoor bijvoorbeeld het logo zo weer opnieuw kan worden ingevuld (hetzelfde geldt voor de achtergrond, kleuren en het lettertype):

Afbeelding met tekst, schermopname, Lettertype, logo

Automatisch gegenereerde beschrijvingAfbeelding met schermopname, tekst, ontwerp

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Lettertype, diagram

Automatisch gegenereerde beschrijving

# Code documentatie

## Sprint 1

(eerste sprint van 20 maart tot 2 april)

Ik was de eerste sprint dan ook erg trots op mijn progress\_bar widget en mijn drop down widget.

Mijn theme\_progress\_bar.dart :  
Ik maak 4 variabelen die ik kan invullen als ik de theme progress bar widget gebruik.

En als deze variabelen dan zijn ingevuld komt kijkt die met een loop en if statements wat die moet teruggeven.

Neem bijvoorbeeld theme\_pick\_color\_screen.dart deze is heeft de eerste en tweede vol en de derde half vol en de laatste nog leeg;

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Dit zou er in de code dan als volgt uitzien, theme\_pick\_color\_screen.dart:

   return Scaffold(

      appBar: AppBar(

        surfaceTintColor: Colors.transparent,

        backgroundColor: idTheme.colorAppBackground,

        leading: Row(

          children: [

            IconButton(

              highlightColor: Colors.transparent,

              icon: Icon(

                Icons.arrow\_back\_rounded,

                color: idTheme.colorAppText,

              ),

              onPressed: () {

                Navigator.of(context).pop();

              },

            ),

          ],

        ),

        title: const ThemeProgressBar(

          step1: 'full',

          step2: 'full',

          step3: 'half',

          step4: 'empty',

        ),

      ),

      body: … rest van mijn code

Door het meegeven van de:

        title: const ThemeProgressBar(

          step1: 'full',

          step2: 'full',

          step3: 'half',

          step4: 'empty',

        ),

Aan de ThemeProgressBar widget, kan die de mee gekregen variables gebruiken om deze op de correcte plaats te krijgen.

import 'package:flutter/material.dart';

import 'package:id\_scan\_nfc\_demo/theme.dart';

class ThemeProgressBar extends StatelessWidget {

  const ThemeProgressBar({

    required this.state1,

    required this.state2,

    required this.state3,

    required this.state4,

    super.key,

  });

  final String state1;

  final String state2;

  final String state3;

  final String state4;

  Widget \_buildProgressItem(String state) {

    return Container(

      height: 6,

      width: 43,

      margin: const EdgeInsets.symmetric(horizontal: 4),

      decoration: BoxDecoration(

        color: idTheme.colorProgressBackground,

        borderRadius: BorderRadius.circular(10),

      ),

      child: Container(

        height: 6,

        margin: (state == 'half')

            ? const EdgeInsets.only(right: 21.5)

            : (state == 'empty')

                ? const EdgeInsets.only(right: 43)

                : null, // null helemaal blauw/primary thema kleur

        decoration: BoxDecoration(

          color: idTheme.colorProgress,

          borderRadius: BorderRadius.circular(10),

        ),

      ),

    );

  }

  @override

  Widget build(BuildContext context) {

    Size size = MediaQuery.of(context).size;

    return SizedBox(

      width: size.width \* 0.625,

      child: Row(

        crossAxisAlignment: CrossAxisAlignment.center,

        mainAxisAlignment: MainAxisAlignment.center,

        children: [

          \_buildProgressItem(state1),

          \_buildProgressItem(state2),

          \_buildProgressItem(state3),

          \_buildProgressItem(state4),

        ],

      ),

    );

  }

}

Maar ook heb ik deze sprint een drop down widget gemaakt (en nog wel meer widgets) die best pittig was. Dit omdat ik het nooit eerder heb aangepast en omdat de documentatie moeilijk te lezen was in het begin.

Ik heb het lettertype die je in dit theme\_drop\_down\_widget.dart ziet en in de pubspec.yaml gedefinieerd. Dit in de pubspec.yaml definiëren is cruciaal anders weet de app niet het pad naar het lettertype of hoe dik gedrukt een lettertype moet zijn.

Hieronder zie je dan ook de 3 lettertypes waar je uit kan kiezen in de dropdown, Lato, Changa en Comic Neue. Deze drie lettertypes kunnen uiteraard in de toekomst nog worden uitgebreid en aangepast.

import 'package:flutter/material.dart';

import 'package:id\_scan\_nfc\_demo/theme.dart';

enum FontLabels {

  lato(

    'Lato',

    TextStyle(

      fontFamily: 'Lato',

      fontSize: 16,

    ),

  ),

  changa(

    'Changa',

    TextStyle(

      fontFamily: 'Changa',

      fontSize: 16,

    ),

  ),

  comicNeue(

    'Comic Neue',

    TextStyle(

      fontFamily: 'ComicNeue',

      fontSize: 16,

      fontWeight: FontWeight.bold,

    ),

  );

  const FontLabels(this.label, this.fontStyle);

  final String label;

  final TextStyle fontStyle;

}

class ThemeDropDownWidget extends StatefulWidget {

  const ThemeDropDownWidget({super.key});

  @override

  State<ThemeDropDownWidget> createState() => \_ThemeDropDownWidgetState();

}

class \_ThemeDropDownWidgetState extends State<ThemeDropDownWidget> {

  FontLabels selectedValue = FontLabels.lato;

  @override

  Widget build(BuildContext context) {

    Size size = MediaQuery.of(context).size;

    return Container(

      margin: const EdgeInsets.symmetric(horizontal: 19),

      decoration: BoxDecoration(

        borderRadius: BorderRadiusDirectional.circular(5),

        color: idTheme.colorAppWhiteBackground,

      ),

      child: Column(

        children: [

          Row(

            mainAxisAlignment: MainAxisAlignment.center,

            children: <Widget>[

              DropdownMenu(

                menuStyle: MenuStyle(

                  backgroundColor: MaterialStateColor.resolveWith(

                    (states) => idTheme.colorAppWhiteBackground,

                  ),

                  surfaceTintColor: MaterialStateProperty.all(

                    idTheme.colorAppWhiteBackground,

                  ),

                ),

                initialSelection: FontLabels.lato,

                onSelected: (value) {

                  setState(() {

                    selectedValue = value!;

                  });

                },

                trailingIcon: Icon(

                  Icons.keyboard\_arrow\_down\_rounded,

                  color: idTheme.colorAppText,

                ),

                selectedTrailingIcon: Icon(

                  Icons.keyboard\_arrow\_up\_rounded,

                  color: idTheme.colorAppText,

                ),

                width: size.width \* 0.9,

                inputDecorationTheme: const InputDecorationTheme(

                  border: InputBorder.none,

                  contentPadding: EdgeInsets.symmetric(horizontal: 16),

                ),

                textStyle: selectedValue.fontStyle,

                dropdownMenuEntries:

                    FontLabels.values.map((FontLabels fontlabel) {

                  return DropdownMenuEntry(

                    value: fontlabel,

                    label: fontlabel.label,

                    trailingIcon: fontlabel == selectedValue

                        ? const Icon(Icons.check\_outlined)

                        : null,

                    style: ButtonStyle(

                      backgroundColor: MaterialStateProperty.all(

                        idTheme.colorAppWhiteBackground,

                      ),

                      overlayColor:

                          MaterialStateProperty.all(idTheme.colorAppBackground),

                    ),

                  );

                }).toList(),

              ),

            ],

          ),

        ],

      ),

    );

  }

}

De widget die je dan ook krijgt is de gewenste drop down met indicatie welke op het moment actief is:

Afbeelding met tekst, schermopname, Lettertype, software

Automatisch gegenereerde beschrijving

## Sprint 2

(van 2 april tot 15 april)

Deze twee weken heb ik het hele opslaan van een thema werkend gekregen en het kunnen kiezen van een thema. Hier onder kan je dan ook zien in de code hoe je een thema kan kiezen. Het thema opslaan is een behoorlijk lang bestand maar kan je in Bitbucket zeker inkijken, het is commit : b039ca3, je kan ook een latere nettere versie inkijken bij commit: f220427.

Voordat we er verder op ingaan moet ik er even wat bij vermelden. De regel,

    ref.read(customThemeNotifierProvider.notifier).//met de methode

is nodig om met SharedPreferences informatie van onze lokale database te kunnen ophalen en uitwisselen.

Hieronder kan je dan ook het thema kiezen in de theme\_picker\_screen.dart.

Dit gaat als volgt, we hebben namelijk al een opgeslagen lijst met CustomTheme, maar nu moeten we die lijst alleen nog ophalen en een CustomTheme er uithalen en actief zetten.

Het exacte stuk in de code die de lijst ophaalt gaat als volgt:

if (customThemeList != null)

                            for (var theme in customThemeList!)

                              ThemeObjectWidget(

                                colorPrimary: theme.primaryColor,

                                colorSecondary: theme.secondaryColor,

                                titleTheme: theme.nameCustomTheme!,

                                defaultTheme: false,

                                isSelected: selectedIndex == customThemeList!.indexOf(theme),

                                onTap: () {

                                  setState(() {

                                    selectedIndex = customThemeList!.indexOf(theme);

                                  });

                                },

Deze code loopt door de customThemeList heen en kijkt dan met theme welke index er geklikt is. De index is standard -1, zodat als je op de pagina komt het actief geklikte thema op Siip default staat.

Hieronder staat nog het stuk code van de complete widget:

class ThemePickerScreen extends ConsumerStatefulWidget {

  const ThemePickerScreen({super.key});

  @override

  ConsumerState<ThemePickerScreen> createState() => \_ThemePickerScreenState();

}

class \_ThemePickerScreenState extends ConsumerState<ThemePickerScreen> {

  List<CustomTheme>? customThemeList;

  int selectedIndex = -1;

  @override

  void initState() {

    super.initState();

    SchedulerBinding.instance.addPostFrameCallback((\_) async {

      customThemeList = await ref

          .read(

            customThemeNotifierProvider.notifier,

          )

          .getAllListsThemes();

      setState(() {});

    });

  }

  Color stringToColor(String colorString) {

    try {

      final int colorInt = int.parse(colorString.replaceFirst('#', ''), radix: 16);

      return Color(colorInt);

    } on FormatException {

      return const Color.fromARGB(0, 158, 158, 158);

    }

  }

  @override

  Widget build(BuildContext context) {

    Size size = MediaQuery.of(context).size;

    return Scaffold(

      appBar: const BackArrowAppbar(),

      body: Container(

        color: idTheme.colorAppBackground,

        child: Column(

          mainAxisAlignment: MainAxisAlignment.spaceBetween,

          children: [

            Expanded(

              child: SingleChildScrollView(

                child: Column(

                  mainAxisAlignment: MainAxisAlignment.spaceBetween,

                  children: [

                    ThemeTitleAndDescriptionHead(

                      titleHead: getText('pick\_theme\_title'),

                      descriptionHead: getText('pick\_theme\_description'),

                    ),

                    Container(

                      width: size.width,

                      margin: const EdgeInsets.symmetric(horizontal: 16),

                      child: Column(

                        children: [

                          ThemeObjectWidget(

                            colorPrimary: siipDefault.primaryColor,

                            colorSecondary: siipDefault.secondaryColor,

                            titleTheme: 'Siip default',

                            defaultTheme: true,

                            isSelected: selectedIndex == -1,

                            onTap: () {

                              setState(() {

                                selectedIndex = -1;

                              });

                            },

                            onTapDelete: () {},

                            onTapEdit: () {},

                          ),

                          if (customThemeList != null)

                            for (var theme in customThemeList!)

                              ThemeObjectWidget(

                                colorPrimary: theme.primaryColor,

                                colorSecondary: theme.secondaryColor,

                                titleTheme: theme.nameCustomTheme!,

                                defaultTheme: false,

                                isSelected: selectedIndex == customThemeList!.indexOf(theme),

                                onTap: () {

                                  setState(() {

                                    selectedIndex = customThemeList!.indexOf(theme);

                                  });

                                },

                                onTapDelete: () {

                                  PopUpDelete(

                                    theTheme: theme,

                                    onTapRefresh: () {

                                      SchedulerBinding.instance.addPostFrameCallback((\_) async {

                                        customThemeList = await ref

                                            .read(

                                              customThemeNotifierProvider.notifier,

                                            )

                                            .getAllListsThemes();

                                        setState(() {});

                                      });

                                    },

                                  ).show(context);

                                },

                                onTapEdit: () {

                                  Navigator.push(

                                    context,

                                    MaterialPageRoute(

                                      builder: (context) => ThemePickLogoBackground(

                                        editCustomTheme: theme,

                                      ),

                                    ),

                                  );

                                },

                              ),

                          Padding(

                            padding: const EdgeInsets.only(

                              top: 24,

                              left: 16,

                              right: 16,

                              bottom: 24,

                            ),

                            child: SafeArea(

                              child: SiipButton(

                                text: getText(

                                  'pick\_theme\_create\_new\_theme',

                                ),

                                onTap: () => {

                                  Navigator.push(

                                    context,

                                    MaterialPageRoute(

                                      builder: (context) => ThemePickLogoBackground(),

                                    ),

                                  ),

                                },

                                width: size.width,

                              ),

                            ),

                          ),

                        ],

                      ),

                    ),

                  ],

                ),

              ),

            ),

            Column(

              children: [

                Divider(

                  color: idTheme.colorAppStroke,

                  height: 0,

                  indent: 0,

                ),

                Padding(

                  padding: const EdgeInsets.only(

                    top: 24,

                    left: 16,

                    right: 16,

                    bottom: 37,

                  ),

                  child: SafeArea(

                    child: SiipButton(

                      // ignore: unnecessary\_null\_comparison

                      widgetState: selectedIndex == null ? WidgetState.disabled : WidgetState.primary,

                      text: getText('general\_continue'),

                      onTap: () {

                        CustomTheme selectedTheme;

                        if (selectedIndex == -1) {

                          selectedTheme = siipDefault;

                        } else {

                          selectedTheme = customThemeListselectedIndex];

                        }

                        ref.read(customThemeNotifierProvider.notifier).setTheme(selectedTheme);

                        ref.read(customThemeNotifierProvider.notifier).changeTokens();

                        Phoenix.rebirth(context);

                      },

                      width: size.width,

                    ),

                  ),

                ),

              ],

            ),

          ],

        ),

      ),

    );

  }

}

Zoals je hebt kunnen zien in de “Volgende” knop (getText(‘general\_continue’) hier boven in de code. Heeft dat best wel een grote onTap() met veel dingen die moeten worden aangepast, de setTheme, changeTokens en de rebirth. Dit is allemaal nodig om het nieuwe thema te zetten en de kleuren ‘tokens’ te veranderen. De Phoenix.rebirth doet iets heel eenvoudigs zoals de naam ook al een beetje zegt herstart het de app. Net zo als een echte phoenix uit de Griekse en Chinese mythologie, waar de phoenix oud wordt dan dood gaat en daarna herboren wordt uit zijn eigen as als kuiken.

De setTheme functie gaat naar de CustomThemeNotifier waar deze methode staat:

  void setTheme(CustomTheme theme) {

    state = theme;

  }

Dit is kort maar krachtig en zet het thema als de huidige state.

Ook worden de tokens nog aangepast met de huidige, nu nieuw gekozen staat van het thema.

void changeTokens() {

    Color primary = state.primaryColor;

    Color secondary = state.secondaryColor;

    String fontStyle = state.customFont;

    // ignore: avoid\_print

    print('kleur 1 notifier: $primary');

    // ignore: avoid\_print

    print('kleur 2 notifier: $secondary');

    idTheme.colorIntroOverlay = LinearGradient(

      begin: Alignment.topCenter,

      end: Alignment.bottomCenter,

      stops: const [

        0.1,

        0.4,

        0.5,

        1,

      ],

      colors: [

        secondary.withOpacity(0.5),

        secondary.withOpacity(0.9),

        secondary,

        secondary,

      ],

    );

    idTheme.setColorIconPickerPrimary = primary;

    idTheme.setColorIconPickerSecondary = secondary;

    idTheme.colorAppBackgroundInverted = primary;

    idTheme.colorNavigationActive = primary;

    idTheme.colorNavigationLink = primary;

    idTheme.colorButtonBackgroundBrand = primary;

    idTheme.colorButtonBackgroundDisabeld = primary.withOpacity(0.35);

    idTheme.colorProgressbar = primary;

    idTheme.colorProgress = primary;

    idTheme.colorThemeItemBackgroundActive = primary.withOpacity(0.05);

    idTheme.colorThemeItemBorderActive = primary;

    // Font Tokens

    // Regular

    idTheme.regularExtremeLargeTitle = TextStyle(

      color: const Color(0xFF000000),

      fontSize: 70,

      fontWeight: FontWeight.normal,

      fontFamily: fontStyle,

    );

    idTheme.regularLargeTitle = TextStyle(

      color: const Color(0xFF000000),

      fontSize: 25,

      fontWeight: FontWeight.normal,

      fontFamily: fontStyle,

    );

….rest van de fonts die moeten worden aangepast

Met de

Color primary = state.primaryColor;

    Color secondary = state.secondaryColor;

    String fontStyle = state.customFont;

Pak je de huidige state (die we dus net opnieuw hebben gezet). Deze geef je mee aan de variabelen die dan vervolgens alle tokens aanpassen.

De afbeeldingen hieronder laten dan ook de pagina van het kleuren kiezen zien. Mocht je bijvoorbeeld op volgende klikken terwijl je Ajax kiest dan zal het scherm er uitzien in het Ajax thema, dan klik je op volgende, herstart de app met de passende intro en foto’s op het scherm. En als je dan weer terug komt bij het thema kies scherm ziet het er als volgt uit.

![Afbeelding met tekst, schermopname, software, Besturingssysteem

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, software, Lettertype

Automatisch gegenereerde beschrijving

App herstart en heeft nu het Ajax kleuren thema

## Sprint 3

(van 15 april tot 26 april)

Deze week was het vooral afronden voor het examen en het verslag afmaken. Maar ik heb nog wel wat opgelost wat wel grote impact heeft.

De thema’s die ik namelijk aanmaakte en die je dan kan kiezen worden nadat je er één actief hebt gezet en de app afsluit, en opnieuw opstart weer gereset naar het Siip default theme. Dit is niet de bedoeling, want je wil dat de app altijd openen in het laatst gekozen thema, de code aanpassingen zien er als volgt uit.

In theme\_picker\_screen.dart heb je een stuk code onderaan het bestand die de setTheme functie aanroept (dit stuk hebben we ook al in week 2 besproken):

 child: SafeArea(

                    child: SiipButton(

                      // ignore: unnecessary\_null\_comparison

                      widgetState: selectedIndex == null ? WidgetState.disabled : WidgetState.primary,

                      text: getText('general\_continue'),

                      onTap: () {

                        CustomTheme selectedTheme;

                        if (selectedIndex == -1) {

                          selectedTheme = siipDefault;

                        } else {

                          selectedTheme = customThemeListselectedIndex];

                        }

                        ref.read(customThemeNotifierProvider.notifier).setTheme(selectedTheme);

                        ref.read(customThemeNotifierProvider.notifier).changeTokens();

                        Phoenix.rebirth(context);

                      },

                      width: size.width,

                    ),

                  ),

Maar nu gaat er wat anders gebeuren met de setTheme methode in de CustomThemeNotifier, want nu hebben we een paar extra methodes toegevoegd namelijk lastTheme(state) en getLastTheme.

  void setTheme(CustomTheme theme) {

    state = theme;

    lastTheme(state);

  }

  Future<void> lastTheme(CustomTheme lastTheme) async {

    await \_customThemeService.lastTheme(lastTheme);

  }

  Future<void> getLastTheme() async {

    SharedPreferences prefs = await SharedPreferences.getInstance();

    if (prefs.getString('lastTheme') != null) {

      String lastTheme = prefs.getString('lastTheme')!;

      Map<String, dynamic> json = jsonDecode(lastTheme);

      state = CustomTheme.fromJson(json);

    }

    changeTokens();

  }

Hier wordt dan ook het laatste thema opgeslagen, met hulp van de custom\_theme\_service.dart. De lastTheme(state) verwijst naar de custom\_theme\_service.dart waar de methode lastTheme(lastTheme) instaat. Ik heb gekozen voor dezelfde naam omdat ze bij elkaar horen en je nooit methodes uit de theme service mag aanroepen in widgets, de CustomThemeNotifier is als het ware daar “the middelman” voor. Deze functie in de CustomThemeService zorgt er dan ook voor dat die de huidige staat van sharedPrefrences krijgt (sharedPrefrences is de database als ware). Dan gaan we de json van lastTheme encode en hem zetten. Dat houdt in dat je dan het laatst gekozen thema zet met de lastTheme die die mee krijgt.

  Future<void> lastTheme(CustomTheme lastTheme) async {

    SharedPreferences sp = await SharedPreferences.getInstance();

    String json = jsonEncode(lastTheme);

    await sp.setString('lastTheme', json);

  }

Nu we dat allemaal hebben gedaan en het laatste thema hebben opgeslagen komen na de Phoenix.rebirth weer terug bij de \_initializerAppState in de initializer\_app.dart, waar we het gezette laatste thema gaan ophalen. Door de getLastTheme();

class \_InitializerAppState extends ConsumerState<InitializerApp> {

  bool isLoading = true;

  @override

  void initState() {

    super.initState();

    ref.read(customThemeNotifierProvider.notifier).getLastTheme();

….rest van mijn code

Deze getLastTheme() gaat dan weer naar de CustomThemeNotifier waar die checkt of het laatste thema niet gelijk is aan null. Is dat niet zo dan zet die daarna het laatste gekozen thema als actief. Ook moet die dan natuurlijk weer de changeTokens() aanroepen om de kleuren en het font weer aan te passen.

  Future<void> getLastTheme() async {

    SharedPreferences prefs = await SharedPreferences.getInstance();

    if (prefs.getString('lastTheme') != null) {

      String lastTheme = prefs.getString('lastTheme')!;

      Map<String, dynamic> json = jsonDecode(lastTheme);

      state = CustomTheme.fromJson(json);

    }

    changeTokens();

  }

Tenslotte komen we weer aan bij het loadingscreen.dart waar het thema dan ook weer wordt opgehaald in de widget zelf. Dit om het achtergrondplaatje en het logo te krijgen.

 Widget build(BuildContext context) {

    final theme = ref.watch(customThemeNotifierProvider); //Het thema wordt //opgehaald

    Size size = MediaQuery.of(context).size;

 return Scaffold(

      body: FutureBuilder( //Dit is omdat het inladen nog al even kan duren

        future: Future.delayed(const Duration(milliseconds: 650)),

        builder: (context, snapshot) {

          if (snapshot.connectionState == ConnectionState.done) {

            return Stack(

              children: [

                Positioned(

                  top: 0,

                  child: SizedBox(

                    width: size.width,

                    height: size.height \* 0.6,

                    child: Image.memory(

                      theme.backgroundImage, //Hier wordt dan het plaatje //opgehaald

                      fit: BoxFit.cover,

                    ),

                  ),

                ),

                Container(

                  decoration: BoxDecoration(

                    gradient: idTheme.colorIntroOverlay,

                  ),

                  child: Center(

                    child: Column(

                      mainAxisAlignment: MainAxisAlignment.center,

                      children: [

                        Image.memory(

                          theme.logoImage, //Hier wordt dan het plaatje //opgehaald

                          width: size.height \* 0.2,

                        ),

                        FadeTransition(

                          opacity: \_controller,

                          child: SlideTransition(

                            position: \_offsetAnimation,

                            child: Padding(

                              padding: EdgeInsets.symmetric(

                                horizontal: size.width \* 0.1,

                              ),

                              child: \_buildGreeting(

                                size,

                              ),

                            ),

                          ),

                        ),

                      ],

                    ),

                  ),

                ),

              ],

            );

          } else { //Een tijdelijke vervanger van de body als die nog aan het //ingeladen is

return Container(

              color: Colors.black,

              width: size.width,

              height: size.height,

            );

          }

        },

      ),

    );

  }

}

# Code feedback

## Week 13 code feedback

(25 – 29 maart)

Deze week ben ik begonnen met programmeren, ook heb ik mijn eerste pull request (pr) aangemaakt. Deze pr’s hadden best wel veel feedback van Silke waar ik veel aan had. Gemiddeld 27 opmerkingen per 23 files (erbij bedenken dat er ook plaatjes in die files kunnen zitten) hierdoor kwam ik er achter dat ik:

* Beter moet lezen.
* Voordat ik mijn pr’s opnieuw laat nakijken, het nog een keer allemaal check.
* Het ontwerp ernaast moet houden om te zorgen dat de app niet afwijkt van het ontwerp.
* Feedback meteen opschrijven als ik het zowel mondeling als in Bitbucket krijg, omdat ik anders vaak de eerste feedback punten vergeet.

## Week 14 code feedback

(2 – 5 april)

Na het extra letten op de feedback van de eerste week heb ik deze week de volgende nieuwe feedback gekregen waar ik op moet letten en aan kan werken:

* De code die ik uit een voorbeeld schrijf en nog niet helemaal begrijp echt begrijpen voordat ik push en commit. (Dus meer onderzoek als ik een nieuwe widgets ga gebruiken die ik niet ken.)

## Week 15 code feedback

(8 – 12 april)

Deze week heb ik weer best wat feedback gekregen op mijn code. Daarom zal ik de volgende week weer op de volgende punten letten:

* Ik moet goeie naamgevingen geven aan mijn variabelen.
* Minder overbodige code zoals ‘mainAxisAlignment: MainAxisAlignment.start,’ en ‘crossAxisAlignment: CrossAxisAlignment.center,’ want soms doen die widget al wat ze moeten doen en staan ze al op de goeie plek waardoor dit dan overbodig is.
* Let goed op met dubbele en overbodige if statment checks zoals:

child: imgLogo != null && widget.imageType == ImageType.logo || imgBackground != null && widget.imageType == ImageType.background

aangezien er alleen logo en background is dus als logo != null dan hoef je in dit geval niet nog extra te checken of imgBackground != null.

## Week 17 code feedback

(22 – 26 april)

Deze week was de laatste week voor het inleveren van het examen. Ik heb dan ook niet heel veel feedback meer terug gekregen. Maar de feedback die ik terug heb gekregen was:

* Niet vergeten de lege bestanden waar ik dingen in heb getest te verwijderen voordat ik een pr aanmaak.
* Bestanden goede duidelijke namen geven.
* De SiipFailedPopUp widget gebruiken in plaats van eigen widgets maken pop-up widget te maken.

# Testplan

Dit testplan bevat het, aanmaken, verwijderen en editen van thema’s.

De verschillende kleuren is om de testen overzichtelijker te maken. De kleuren geven aan bij welk onderdeel de testen ongeveer horen, de kleuren/ testen kunnen iets overlappen.

Wit: De stappen voor het starten van de app en het onboarden.

Mint: De stappen voor het aanmaken van een nieuw thema en uploaden van een logo.

Lichtblauw: De stappen voor uploaden achtergrond.

Paars: De stappen voor het in gevuld hebben van de kleuren.

Oranje: De stappen voor een lettertype kiezen en de pop-up voor het opslaan en herstarten.

Grijs: De stappen voor het verwijderen van een thema.

Groen: De stappen voor het bewerken van een bestaand thema.

![Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Afbeelding met tekst, Lettertype, nummer, lijn

Automatisch gegenereerde beschrijving

Afbeelding met tekst, nummer, lijn, Lettertype

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Afbeelding met tekst, lijn, Lettertype, nummer

Automatisch gegenereerde beschrijving

Afbeelding met tekst, Lettertype, schermopname, lijn

Automatisch gegenereerde beschrijving

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

# Verbetervoorstellen

## Verbetervoorstellen product en proces week 13

20 tot 22 maart:

Dit was de eerste week en ik heb nog geen echte verbetervoorstellen kunnen vinden voor de app. Ook heb ik nog geen verbetervoorstellen voor het werkproces. Dit ging soepel, maar dat komt omdat er alleen maar gedocumenteerd is, en in draw.io diagrammen zijn gemaakt.

## Verbetervoorstellen product en proces week 14

25 tot 29 maart:

Verbetervoorstellen voor het product:

Het zou mooi zijn als je in de toekomst het tandwielicoon die je naar de themaschermen brengt onzichtbaar kan maken of dat die standaard niet zichtbaar is. Dit voor de klanten waaraan je de demo van de app geeft, zodat ze niet afgeleid kunnen raken van het tandwielicoon.

Verbetervoorstellen voor werkproces:

Het zou beter zijn als ik meteen de testplannen invul en niet wacht tot het einde van de dag of totdat ik een pr wil aanmaken. Dit om vergeten testen te voorkomen.

## Verbetervoorstellen product en proces week 15

2 tot 5 april:

Verbetervoorstellen voor het product:

Het zou mooi zijn en een nice to have om bij het kies themascherm de thema’s ook al in de font te zetten waarin ze zijn:

Afbeelding met tekst, schermopname, Lettertype, nummer

Automatisch gegenereerde beschrijving

Verbetervoorstellen voor werkproces:

Het zou handig zijn als ik even voor mezelf opschrijf wat ik wil zeggen in de stand-ups. Zodat ik niet onduidelijk kan overkomen en niet kan gaan stotteren. Op deze manier kan ik ook nog makkelijker vragen beantwoorden, omdat ik mij dan beter heb voorbereid.

## Verbetervoorstellen product en proces week 16

8 tot 12 april:

Verbetervoorstellen voor het product:

Toen ik met de verkoopmedewerker Remco sprak over de app kwam er het idee uit om in een latere versie er voor te zorgen dat je in de kleurenkiezer de kleuren uit het logo of achtergrondplaatje kan kiezen. Dit door bijvoorbeeld over het plaatje te zweven en dan met een soort pipet de kleur te kiezen.

Verbetervoorstellen voor het werkproces:

Als de concentratie op is even een nieuwe kop thee halen.

## Verbetervoorstellen product en proces week 17

22 tot 26 april:

Verbetervoorstellen voor het product:

Nu ik aan het einde ben gekomen van het examen ben ik gaan nadenken en testen wat ik nog moet verbeteren voor in de app zelf. Het examen mag dan wel klaar zijn maar de app is dat nog niet. Wat ik moet verbeteren is:

* Het inladen van het thema want gaat nog een beetje stroef.

Je ziet dat er plaatjes worden ingeladen, dit moet zeker in de toekomst worden aangepast.

* De delete pop-up worden aangepast naar de SiipFailedPopUp.
* De edit-functie werkt nog niet compleet en daarom is het op het moment uit de app gelaten, maar het zou mooi zijn als die uiteindelijk wel werkt, zodat je je thema naast verwijderen ook kan bewerken.

En als laatst moet er veel meer nog getest worden op IOS in plaats van Android wat ik nu voornamelijk heb gedaan.

Verbetervoorstellen voor het werkproces:

Blijf op de commit berichten letten, na een paar dagen werken worden ze soms wat cryptisch en dit kan zeker voorkomen worden door er op te blijven letten.

# Sprint review + retrospectief

Tijdens de komende weken gaan er twee sprint reviews worden gehouden en ook daar achterna meteen twee retrospectieven. Deze review en retrospectieven kunnen mij de volgende sprint weer verder helpen om op mijn werkwijze te letten en te reflecteren.

## Sprint review + retrospectief 2-04-2024

Dit was de eerste sprint review en retrospectief tijdens dit examen. De dingen die wij dan ook hebben gereviewd is:

Wat heb je gedaan/afgemaakt:

* Intro scherm,

Dat is het laadscherm als je de app opstart.

* Onboarding

Het in kunnen scannen van je paspoort, ID of rijbewijs. Om te kunnen zien wat het inlogproces zou zijn.

* User stories

Het compleet afgemaakt van de user stories zodat ik duidelijk heb hoeveel tijd alles gaat kosten en om te zien wat prioriteit heeft en wat niet.

* Activity diagram

Zien welke activiteiten er in de app gaan plaatsvinden als je op bepaalde knoppen klikt.

* Use-case diagram

Goed duidelijk hebben gekregen wat een gebruiker nu wel en niet mag.

* Gear icon

Open het nieuwe scherm waar je uit thema’s kan kiezen.

Wat heb ik nog niet gedaan of afgemaakt:

* Hardcoded theme create theme flow (dit het kunnen aanmaken van een nieuw thema zonder alle logica er meteen achter)

Hij was bijna af maar ik zat heel lang vast op de drop down widget omdat ik die nog niet eerder heb gebruikt en alles moet wel volgens het ontwerp zijn. Wat vertraging heeft opgeleverd.

* Class diagram

Hij is gemaakt maar hij is gewoon nog niet goed genoeg en daar moet ik dus nog een keer goed en concreet bij de volgende sprint overheen kijken en aanpassingen maken waar nodig.

Tijdens de retrospectief kijken wij naar 3 criteria, start doing, stop doing en keep doing. Dit is een manier van zelfreflectie waar je kijkt wat je kan gaan doen voor beter resultaat, de start doing. Stop doing, waar je kijkt naar dingen die moeten stoppen voor een beter resultaat en de keep doing kijken naar het de dingen die al wel goed gaan en die door mogen blijven gaan.

Mijn start, stop en keep doing zijn als volgt;

Start doing:

* Voordat ik een pr aanmaak van een branch al het testplan helemaal up-to-date hebben gemaakt met de testen die ik al heb gedaan.

Dit omdat ik anders per ongeluk een test zou kunnen vergeten, dus alles wat ik doe meteen testen en documenteren.

Stop doing:

Hier was eigenlijk niks deze sprint over te zeggen.

Keep doing:

* Feedback opschrijven in een notitieboekje

Hier kwam ik halverwege deze sprint achter. Want na het krijgen van feedback op mijn code liep ik vaak terug naar mijn tafel maar was ik al de eerste punten weer vergeten. Ondanks dat ze ook in Bitbucket staan als commentaar is het voor mij toch handiger om ze ook op te schrijven zodat ik ze echt fysiek moet afvinken en moet hebben gecheckt.

## Sprint review + retrospectief 15-04-2024

Dit was de tweede sprint review en retrospectief tijdens dit examen. De dingen die wij dan ook hebben gereviewd is:

Wat heb je gedaan/afgemaakt:

* Het Aanmaken van een thema

Je kan nu thema’s aanmaken en opslaan, met logo, achtergrond, twee kleuren en font er bij.

* Kiezen uit een bestaand thema

Als je nu een thema hebt aangemaakt kan je hem kiezen. Of een ander bestaand thema kiezen.

* Verwijder thema

Je kan nu ook thema’s verwijderen.

Wat heb ik nog niet gedaan of afgemaakt:

* Aanpassen van een thema

Dit werkt nog niet helemaal en is me nog niet gelukt om te maken.

Ook tijdens dit retrospectief kijken wij naar 3 criteria, start doing, stop doing en keep doing. We hebben dan ook wat dingen gevonden waar ik op kan letten voor volgende sprint namelijk:

Start doing:

* Nog meer tijd vrijmaken voor documentatie,

Ik merk toch dat er nog meer en nog wel beter gedocumenteerd kan worden.

* Nog wat meer onderzoek doen tijdens het vastlopen bij problemen.

Stop doing:

* Als ik iets echt niet weet en het niet kan vinden op internet, alleen Silke voor hulp vragen aangezien zij mijn begeleider is.

Keep doing:

* Voordat ik een pr aanmaak het testplan bijwerken en updaten.
* Feedback opschrijven in een notitieboekje

# Conclusie

Tijdens dit project ben ik er wel achter gekomen dat ik de front-end mij beter ligt en dat ik dit leuker vind om te doen. Maar ondanks dat, het niet altijd handig is om de front-end eerst te maken. Ik heb namelijk tijdens dit project eerst alle stappen volgens het design en zonder logica gemaakt, met het goede uiterlijk. Maar toen ik uiteindelijk de logica er in moest krijgen had ik er wat moeite mee, omdat ik al meteen naar veel code keek wat soms verwarrend was.

Ik had begin van dit examen must, should en could haves opgesteld en ik kan zeggen dat de must haves zijn gehaald. De should haves zijn ook behaald maar kunnen zeker nog wel wat verbetering gebruiken voor gebruikersvriendelijkheid, en de could haves zijn nu nog over. Ik heb de voorbeeldtelefoon al wel leeg neer gezet op de plek waar die hoort maar hij werkt nog niet en moet in dus nog in de toekomst worden gemaakt.

De verbeterpunten voor de should en could haves zijn:

* De input voor de kleuren werkend krijgen.
* Het inladen van de afbeeldingen in het intro scherm, is niet meer te zien.
* Het kunnen editen van al bestaande thema’s.
* Het maken van de niet bestaande voorbeeldtelefoon vanuit het ontwerp.

Ook was ik heel blij met mijn begeleider Silke die mij goede feedback en uitleg gaf.

Ik hoop in de toekomst nog meer feedback en uitleg te krijgen, zodat ik mijzelf nog meer kan verbeteren en ontwikkelen.