

Project Web

Inh 2021/2022

Darshni Banwarie
500785940
Milan van den Bovenkamp
10/11/2021

Design Rationale

Inhoudsopgave

Probleemstelling	1
Design Challenge	2
Iteraties en Ontwerpkeuzes	3
Zoekgedrag	5
Multi Device	7
Context Flavors	8
Data	9
Conclusie	10
Bronnenlijst	11

Introductie

De afgelopen 8 weken heb ik mij gefocust op het bedenken en ontwerpen van een multi-device app voor op reis met 2 fases.

Mensen hebben meestal een overstap tijd van 6 minuten en dat is niet genoeg tijd om even tussendoor te stoppen bij een winkel op het station.

In dit verslag leg ik mijn concept uit dat bestaat uit een voor- en tijdens fase waarin mensen tijdens hun overstap tijd consumptie bestellen en dit ophalen om vervolgens rustig hun trein te halen.

Verder presenteer ik mijn ontwerpkeuzes op basis van theorie; How People look for information door Donna Spencer, Context Flavors (D.E.T.A.I.L.S), Designing with context en 3C Strategieën.

Probleemstelling

Vanuit de klant

Sommige NS klanten hebben 10 minuten of langer overstaptijd, wat hun de kans geeft om rustig ergens wat eten en drinken te halen. Niet alle reizigers hebben dezelfde overstaptijd, soms krijgen zij een overstap tijd van 3- tot 6 minuten; wat niet genoeg tijd is om iets te kopen op het station.

Reizigers kiezen natuurlijk ervoor om hun trein op tijd te halen, maar zullen dan geen prettige rit ervaren vanwege dorst en/of honger.

Tegen welke problemen loopt de klant aan?

- De overstaptijd is te kort om consumptie te kopen voor de trein rit



Afb. 1: Afbeelding van Amsterdam Centraal station

Design Challenge

“Hoe kan de NS reizigers helpen om tijdens hun korte overstaptijd consumptie te kunnen kopen?”

De oplossing - fases

Voor de overstaptijd kan de gebruiker via de NS Afhaal app hun eten en drinken bestellen. In de app wordt er aangewezen dat ze de bestelling bij een NS automaat kunnen halen en op welk perron. De app genereert na het bestellen een QR-code. Tijdens dit proces krijgt de gebruiker een korte uitleg waar en hoe zij hun bestelling kunnen ophalen met de QR-code.

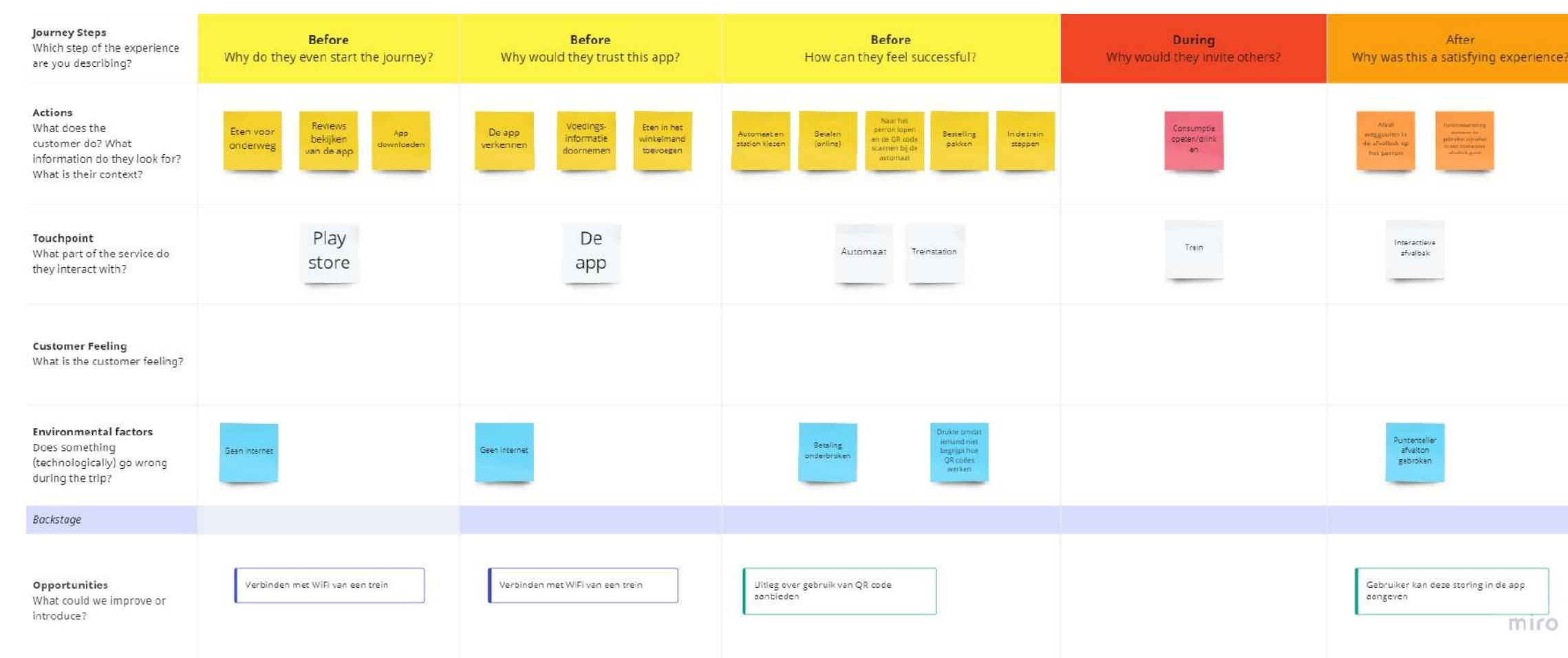
Tijdens de overstaptijd zoekt de gebruiker naar de gekozen NS automaat en scannen zij hun QR-code. De automaat herkent dit en haalt de bestelling tevoorschijn. De gebruiker haalt dit op en stapt rustig de trein in. Dit proces heeft een maximale duur van 2 minuten.

Link interactieve mock-up: shorturl.at/oLMN6

Het proces in het kort

In de beginfase heb ik content onderzoek gepleegd om te kunnen bepalen welke data beschikbaar zal zijn in de oplossing en hoe dit geïmplementeerd kan worden. Verder gaf dit mij inzicht in wat voor beperkingen er kunnen ontstaan zoals: een product is niet meer beschikbaar in de automaat en mensen die last hebben van allergieën.

Na dit allemaal in kaart te hebben gebracht ben ik verder gegaan met het onderzoeken van de customer journey door middel van een customer journey map. Hierin vond ik belemmeringen zoals: **het perron is erg druk, mensen begrijpen niet hoe een QR-code werkt en door technische redenen is een betaling onderbroken.**



Afb. 2: De customer journey map (klik link voor meer duidelijkheid: https://miro.com/app/board/o9J_lUrkJC0=/)

Tenslotte heb ik iteraties aan de hand van peerfeedback sessies gepleegd (procesboek, p. 10, 11, 18 en 21).

Iteraties en ontwerpkeuzes

Aan de hand van peerfeedback heb ik verschillende iteraties gemaakt en daarop mijn ontwerpkeuzes gebaseerd.

In versie 0.8 en 0.9 kon de gebruiker nog niet op een efficiënte manier een perron en station kiezen om uiteindelijk een NS automaat te vinden. Dit werd allemaal handmatig gedaan.

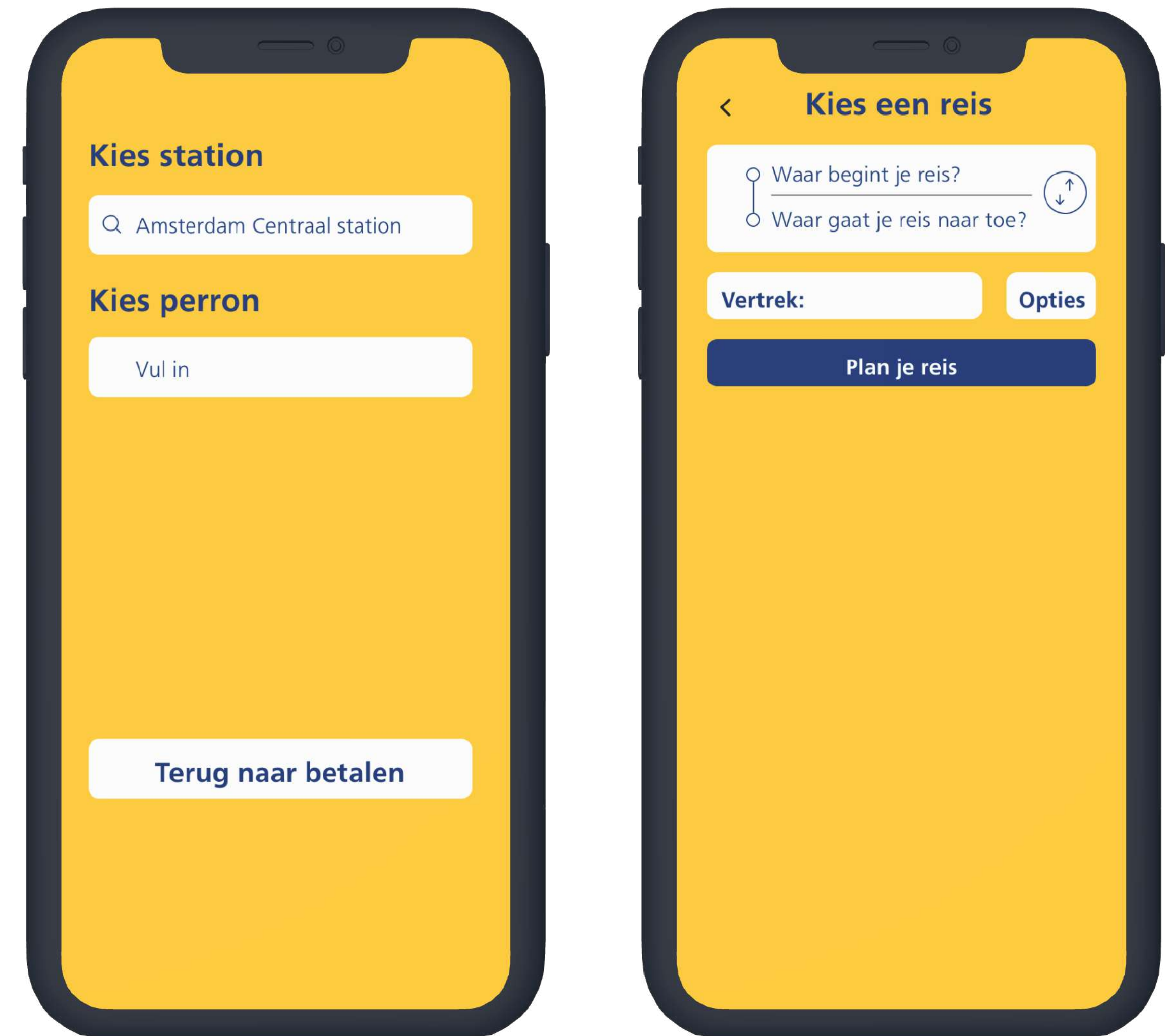
Toen kreeg ik als feedback om net als de officiële NS app de trein reizen in de app te implementeren, zodat de gebruiker het perron en automaat voor zich had en ook meteen de trein rit.

Als bijvoorbeeld het perron wordt gewijzigd, krijgt de gebruiker automatisch een update in de app hierover. Dit heb ik uiteindelijk uitgewerkt in versie 1.0.

Link v0.8: <https://www.figma.com/file/UGovXZyJgVUavSrWerKVrr/Project-Web-v.0.8?node-id=0%3A1>

Link v0.9: <https://www.figma.com/proto/o5mGoLGU3Jwr681zJKX1D9/Project-web-v-0.9?page-id=0%3A1&node-id=2%3A11&viewport=241%2C48%2C0.06&scaling=min-zoom&starting-point-node-id=2%3A653>

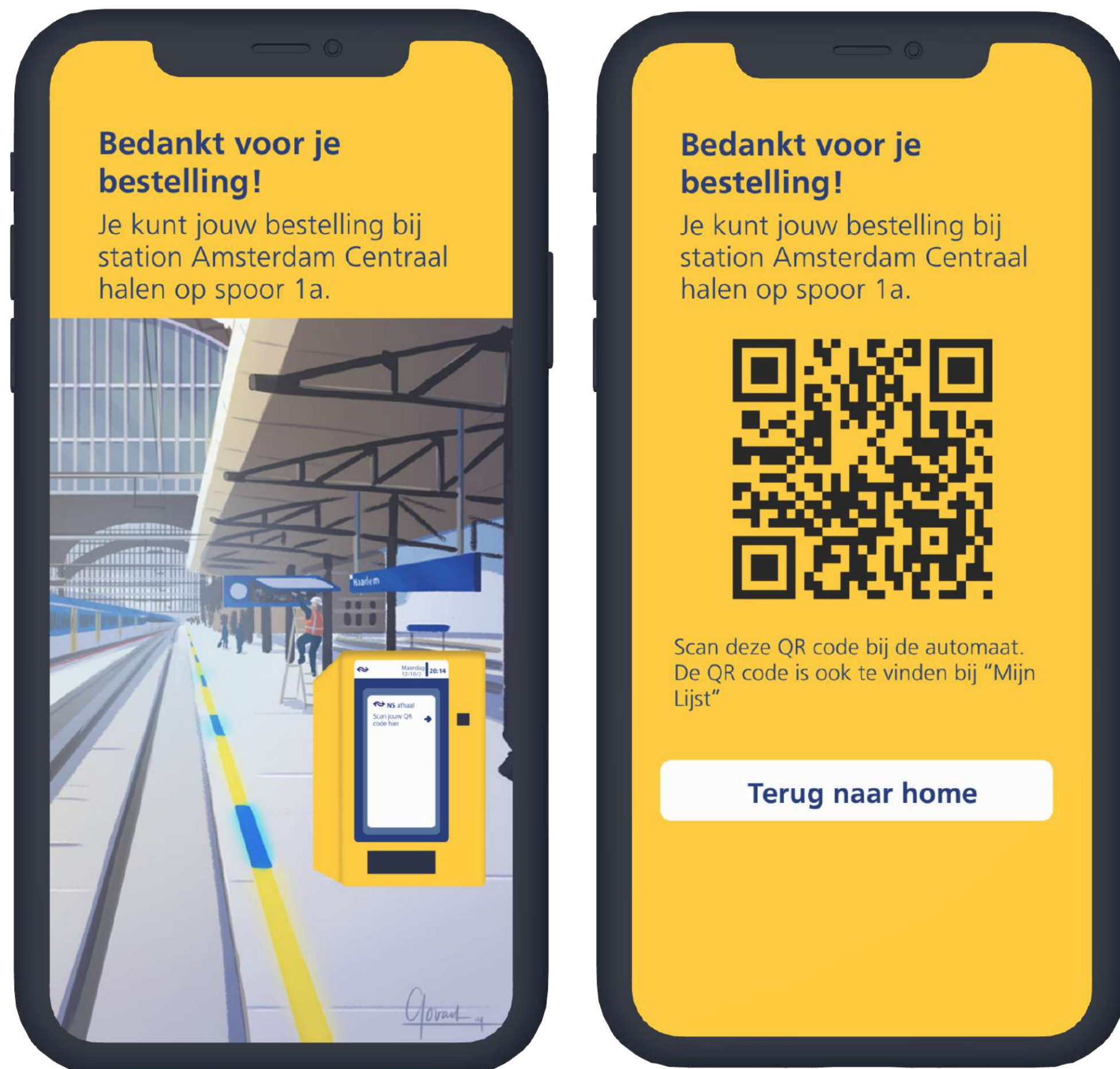
Link v1.0: <https://www.figma.com/proto/KTvH8tXNralfOqfLUvSW4C/project-web-v-1.0?page-id=0%3A1&node-id=2%3A680&viewport=241%2C48%2C0.08&scaling=min-zoom&starting-point-node-id=2%3A636>



Afb. 3: Versie 0.8 en v1.0 iteratie om een perron uit te kiezen.

Iteraties en ontwerpkeuzes

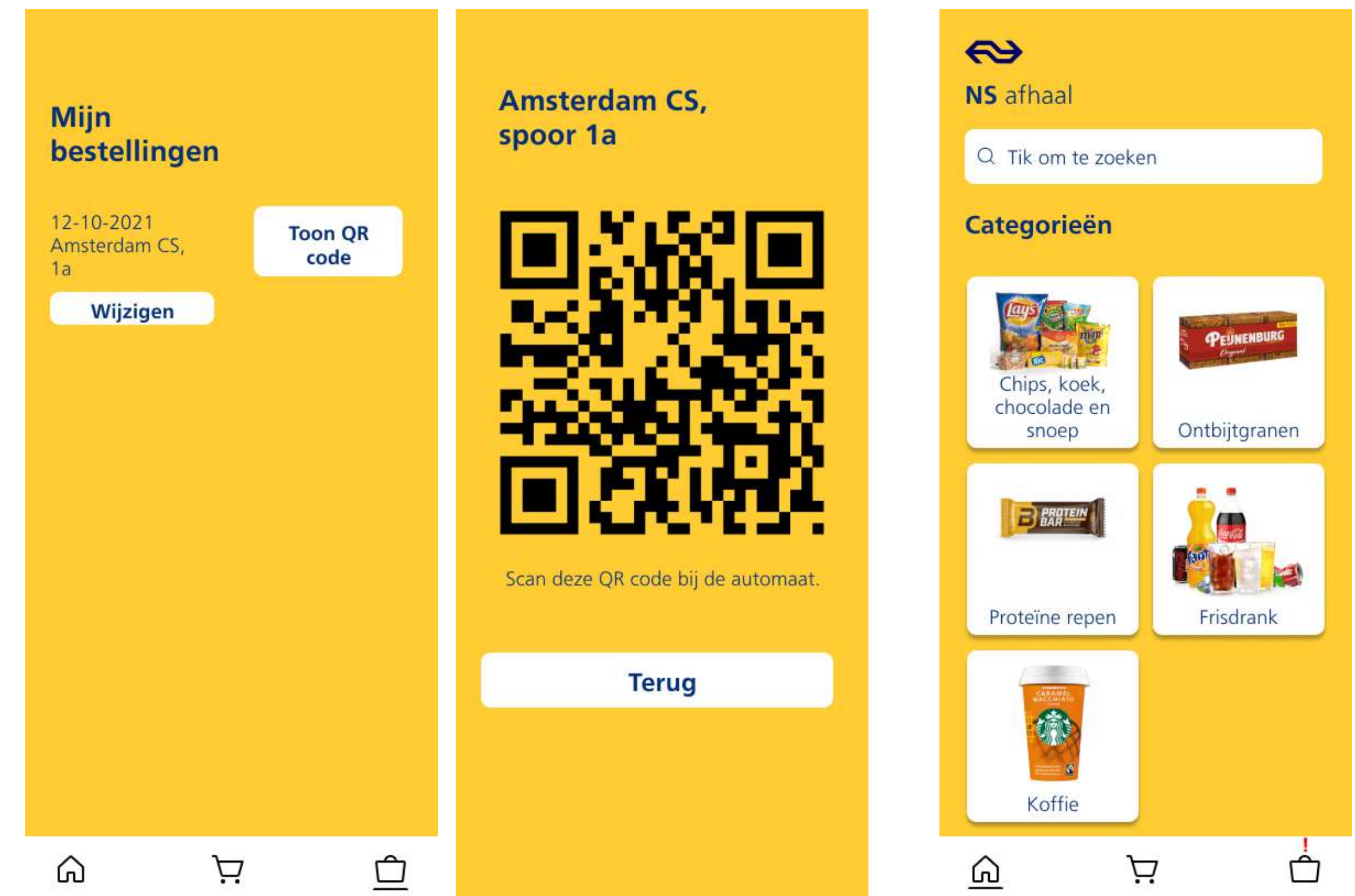
Het is zeer belangrijk dat de gebruiker ten alle tijden geïnformeerd is over de hoe en waar van hun bestelling. Nadat zij een bestelling hebben geplaatst krijgen zij achter elkaar twee schermen te zien waar er beschreven wordt waar zij hun bestelling kunnen ophalen (afb. 4).



Afb.4: Overzicht informatie bestelling.

Tenslotte kan de gebruiker de locatie van hun bestelling nog bij "Mijn bestellingen" vinden (afb. 5).

Als het spoor wijzigt, krijgen ze ook een push notificatie in hun telefoon en in de app staat er een roep teken bij "Mijn bestellingen" om hun aandacht te trekken (afb. 6).

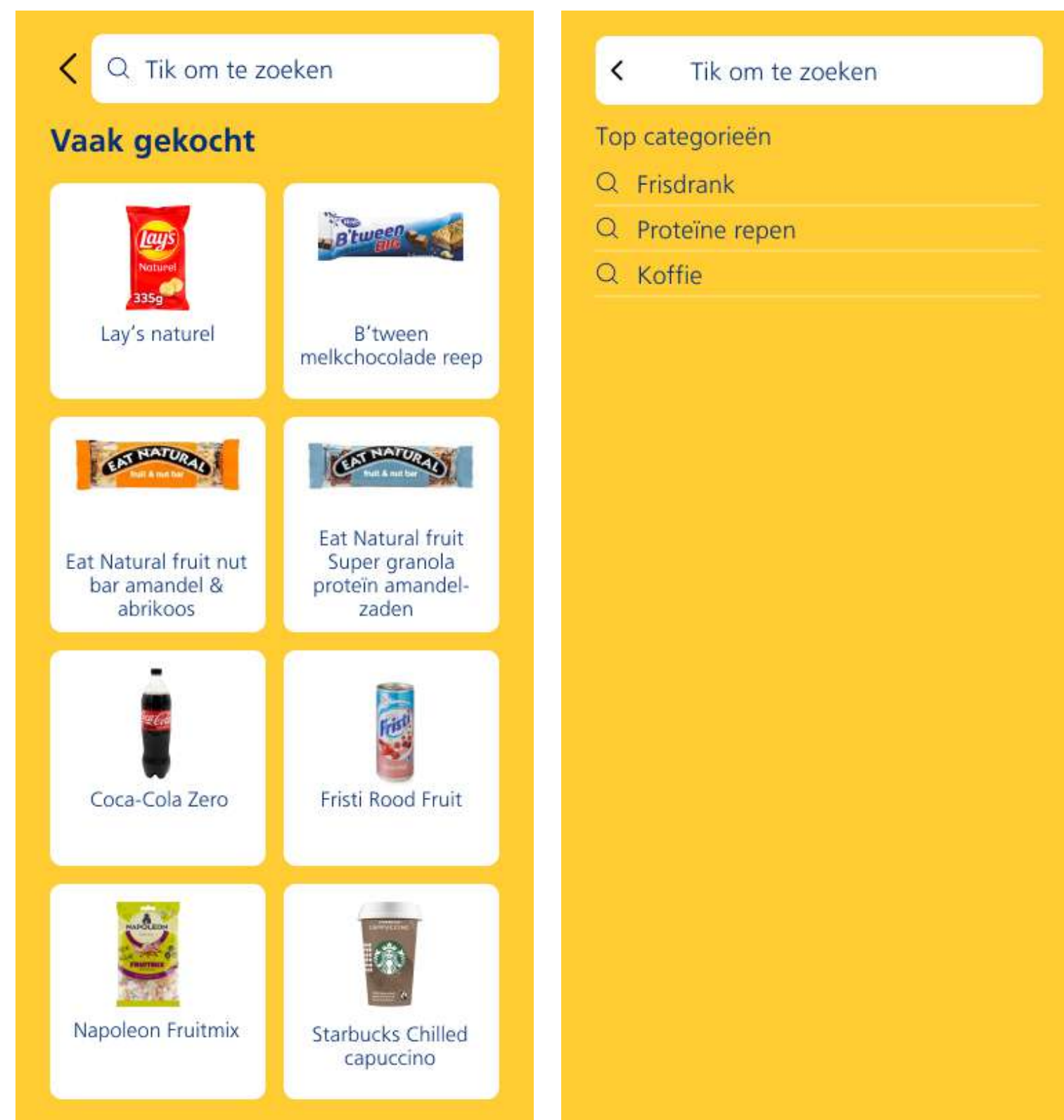


Afb.5: Overzicht informatie over de bestelling.

Afb.6: Overzicht scherm wanneer er een wijziging perron plaatsvindt.

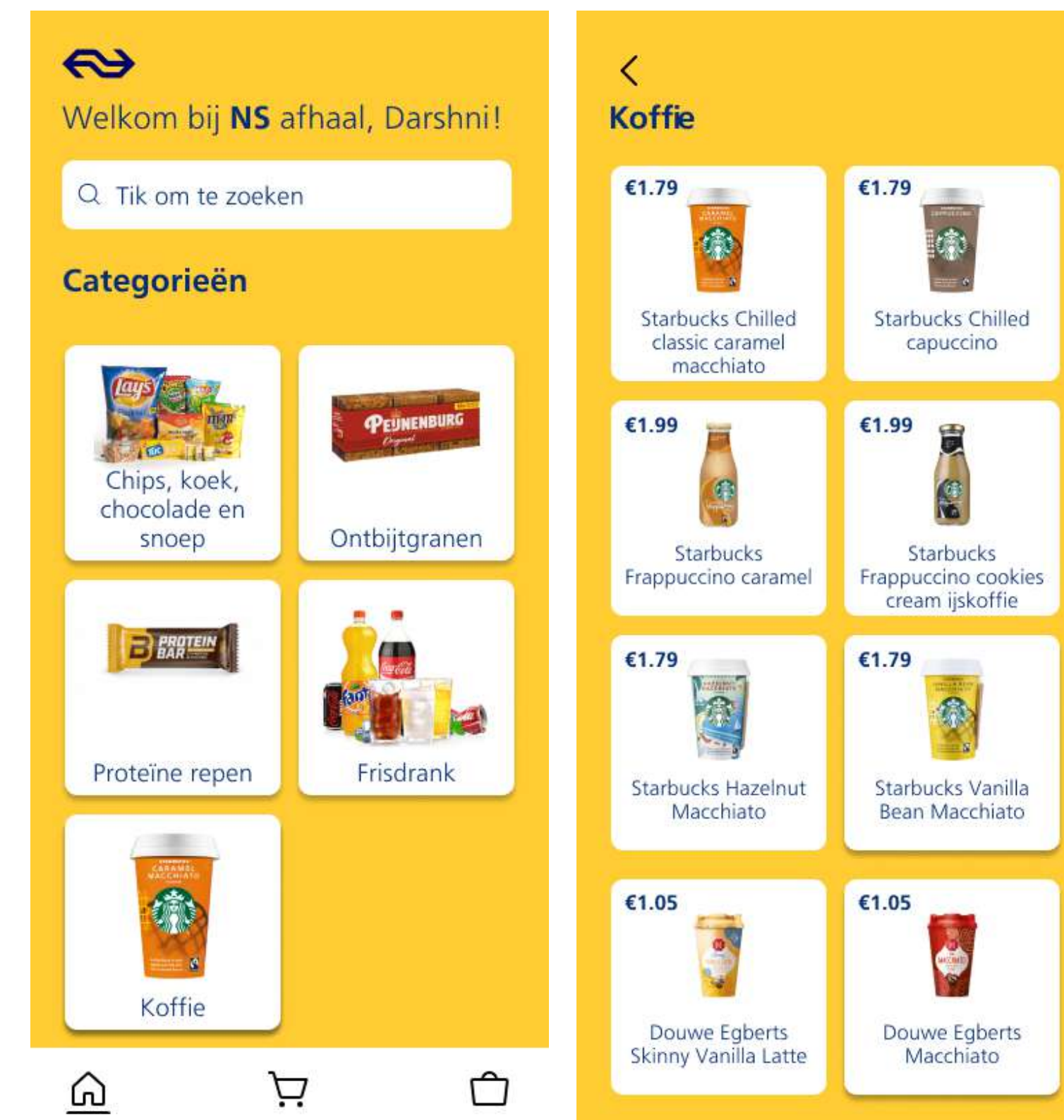
Zoekgedrag

Op basis van het boek How People Look for Information heb ik de user interface ook aangepast. In afbeelding 7 heb ik **Discovering unknown things** (Spencer, p. 97) toegepast. De gebruiker ziet wat heel vaak is gekocht in de app en welke categorieën vaker zijn bezocht. Dit is zodat ze producten kunnen ontdekken of een idee krijgen in wat ze kunnen kopen als ze dat niet weten.



Afb. 7: "Zoek een product" scherm.

Verder heb ik **Exploring** geïmplementeerd in de user interface (afb. 8). De gebruiker ziet een lijst van categorieën waarin zij kunnen ontdekken wat voor producten zij allemaal kunnen kopen. Dit heb ik uitgewerkt voor het geval de gebruiker geen specifieke idee heeft waar ze moeten beginnen.



Afb. 8: Overzicht van de producten scherm.

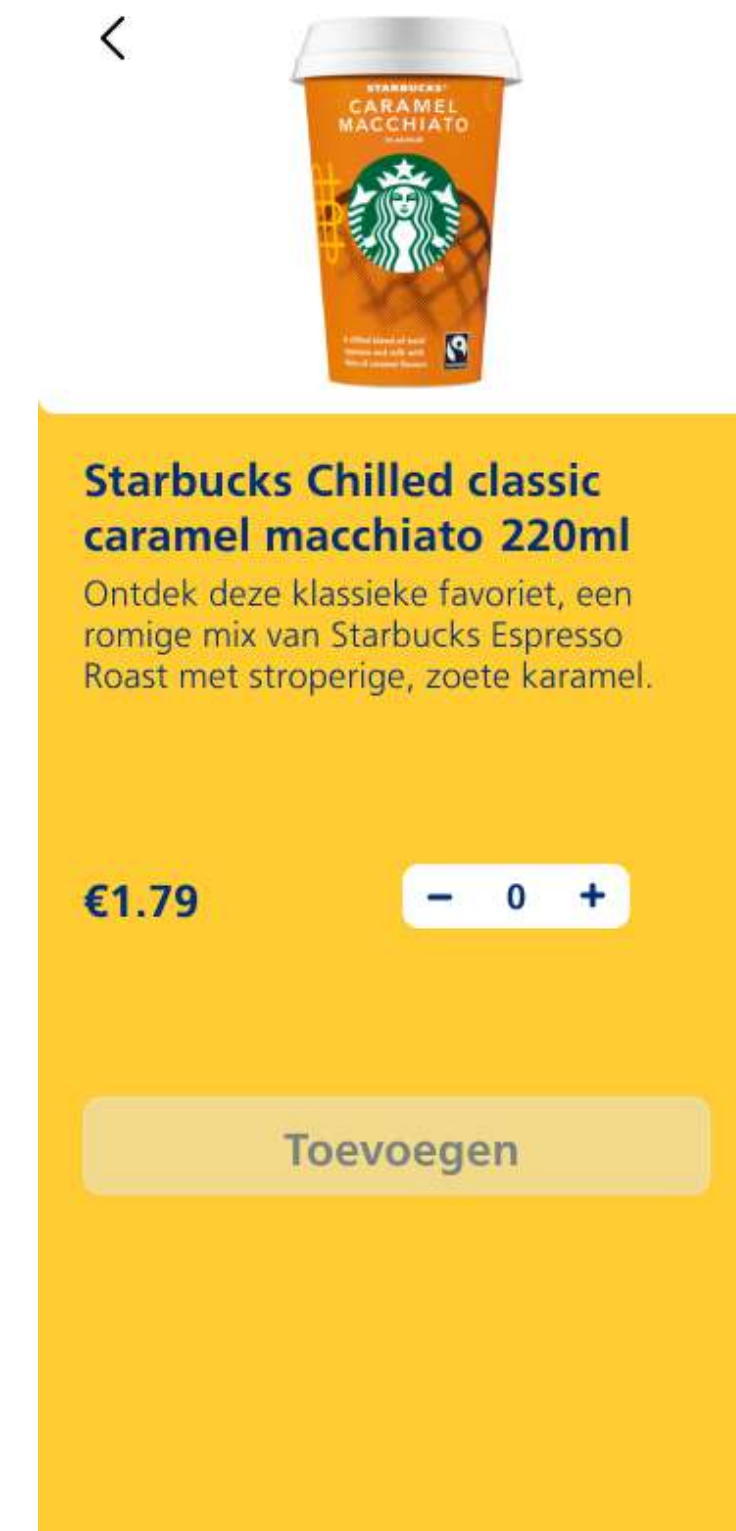
Zoekgedrag

Finding Known Items (afb. 9). Als de gebruiker even snel wilt weten op welk perron ze moeten zijn, kunnen zij even snel op de “Mijn bestellingen” scherm kijken en dan zetten ze hun telefoon weer weg.



Afb. 9: “Zoek een product” scherm.

Getting a broad idea (afb. 10): de gebruiker ziet een samenvatting van een productomschrijving. De meest belangrijke informatie staat er geschreven.



Afb. 10: Detailpagina van een product.

MultiDevice

Nadat de gebruiker hun bestelling heeft geplaatst, moeten zij dit ophalen bij een NS automaat. De automaat heb ik aangepast en meer persoonlijk gemaakt. De gebruiker krijgt altijd een vriendelijke groet met hun naam en tenslotte een overzicht van hun aankoop voordat zij het ontvangen (afb. 11).

In mijn oplossing is het multi device gedeelte van mobiel naar NS automaat.

Normaliter hebben automaten aan de onderkant een bak waaruit de producten komen, maar ik vond dat dat heel onhandig zou zijn voor mensen met een lichamelijke beperking. Dus heb ik het bakje rechtsboven aangebracht zodat ieder mens met gemak erbij kan komen.



Afb. 11: NS automaat.

ContextFlavors

Device: mobile en NS automaat

Environment: binnen (metro/bus) en buiten

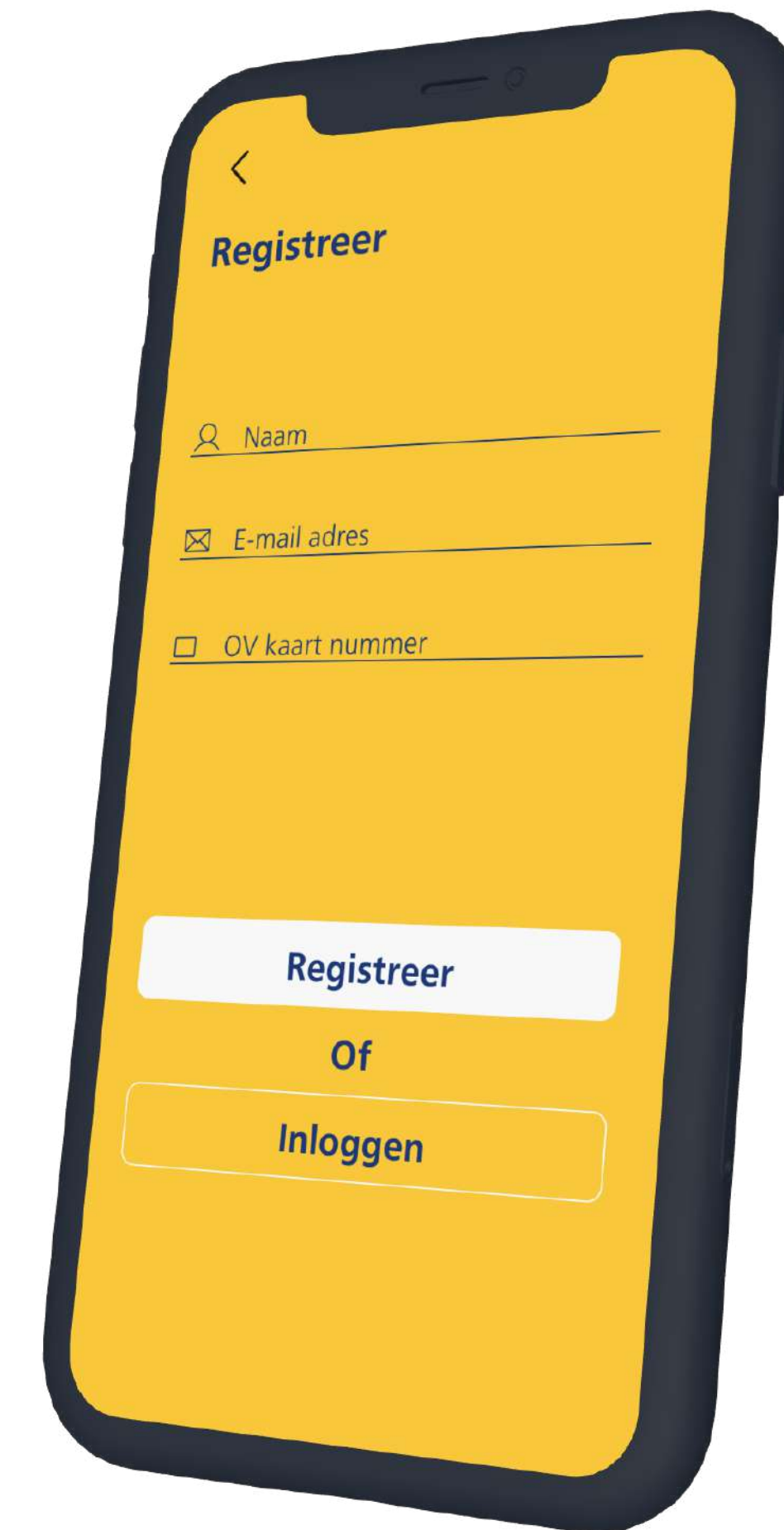
Time: niet zo zeer van toepassing. De gebruiker zou de app elk moment van de dag gebruiken.

Activity: consumptie bestellen

Individual: de app is best persoonlijk, omdat de gebruiker vaak bij hun naam wordt aangesproken wanneer ze op één van de devices bezig zijn (afb. 11).

Location: de app is niet meer location-based, omdat de gebruiker zelf een reis zal plannen in de app om bij het beginstation hun bestelling op te halen (afb. 3).

Social context: niet van toepassing.



Afb. 12: Overzicht registreerscherm.

Data

Welke Data is er allemaal?

- Naar aanleiding van mijn onderzoek is er data dat in een QR-code wordt gegenereerd. De API die ik hiervoor heb onderzocht is QR Code Generator. Deze API genereert en scanned QR-codes. **Link: <https://goqr.me/api/>**
- Verder moet de app bijhouden welke bestellingen allemaal binnen komen en dit verwerken in het systeem. Dit zal aan de hand van Online Food Ordering System gebeuren. **Link: <https://www.gloriafood.com/online-food-ordering-system-for-restaurants>**.
- Daarnaast hebben veel mensen voedselallergieën, dus het zou handig zijn als de app deze informatie aanbied. Dit verloopt via Spoonacular API. **Link: <https://spoonacular.com/food-api>**.

Conclusie

Design Challenge

“Hoe kan de NS reizigers helpen om tijdens hun korte overstaptijd consumptie te kunnen kopen?”

Peerfeedback

In de laatste week heb ik meer gefocust op het valideren van mijn product tijdens peerfeedback.

De belangrijkste interactie vind ik hoe de gebruiker aan een perron nummer zou komen. Dit had ik opgelost door NS planner te implementeren wanneer de gebruiker een perron en station kiest. En volgens de peerfeedback was de oplossing wel goed verbeterd.

Verder bleek uit conversaties met mijn peers dat zij de overstaptijden ook heel kort vinden en zelf niet vaak eten en drinken bij zich hebben. Dus zo een product zouden zij zelf willen gebruiken.

Keuzes

Bij de voedsel-en drankopties heb ik ervoor gekozen om houdbare items in de schappen toe te voegen. De gebruiker heeft de keuze tussen gezond voedsel (Proteïne bars en ontbijtgranen) en snoep. Dus de keuze is vrij bij hen met wat zij willen eten. Hetzelfde met drinken.

Normaliter hebben automaten aan de onderkant een bak waaruit de producten komen, maar ik vond dat dat heel onhandig zou zijn voor mensen met een lichamelijke beperking. Dus heb ik het bakje rechtsboven aangebracht zodat ieder mens met gemak erbij kan komen.

Bronnenlijst

Afb. 1: Z.J.A (z.d) Amsterdam Centraal, Amsterdam. Geraadpleegd op 8/11/2021 van https://www.zja.nl/image/2020/12/10/phs_amsterdam_centraal_amsterdam_copyright_zja_02.jpg%28mediaclass-carousel-big.43ec142639ef8d13fa10e99fe3672c3a21da2a53%29.jpg

Afb. 2-12: High-fidelity prototype, Darshni Banwarie. 2021.

Spencer, D (z.d). A Practical Guide to Information Architecture. Geraadpleegd op 8/11/2021 van https://drive.google.com/file/d/1-Fjg00DCMdDY91Sazv_6n2ewqdCKO0WW/view