

BACHELORPROEF

Single vs. Multi page applications

Wanneer kies je wat?

Bachelor	Toegepaste Informatica
Keuzetraject	Software Engineer
Academiejaar	2022 - 2023
Student	Thibaut Wittevrongel
Interne begeleiders	Jill VandenDriessche, Dieter Mourisse (Howest)

BACHELORPROEF

Single vs. Multi Page applications

Wanneer kies je wat?

Bachelor	Toegepaste Informatica
Keuzetraject	Software Engineer
Academiejaar	2022 - 2023
Student	Thibaut Wittevrongel
Interne begeleiders	Jill VandenDriessche, Dieter Mourisse (Howest)

Toelating tot bruikleen

De auteur(s) geeft (geven) de toelating deze bachelorproef voor consultatie beschikbaar te stellen en delen van de bachelorproef te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze bachelorproef.

The author(s) gives (give) permission to make this bachelor dissertation available for consultation and to copy parts of this bachelor dissertation for personal use. In all cases of other use, the copyright terms have to be respected, in particular with regard to the obligation to state explicitly the source when quoting results from this bachelor dissertation.

6/06/2023

Woord vooraf

In het kader van mijn opleiding bachelor Toegepaste Informatica met keuzetraject Software Engineering heb ik de kans gekregen om een eigen scriptie te mogen schrijven. Dit als finale opdracht van mijn opleiding. De vraag waarop antwoord gegeven wordt gaat als volgt: 'Single vs. Multi Page applications: wanneer kies je wat?'.

Deze onderzoeksvraag kwam uit de onderzoeksgroep van TI. Om deze vraag op te lossen heb ik na een uitvoerig kwalitatief en kwantitatief onderzoek een antwoord kunnen vinden op de onderzoeksvraag.

Graag zou ik als eerste mijn bachelorproef begeleiders, Dieter Mourisse en Jill VandenDriesche, lectoren aan Howest Brugge, van harte bedanken. Beiden stonden altijd klaar om me feedback te geven en om me nieuwe inzichten te geven als ik op bepaalde punten even vastliep. Ook Heidi Terryn, lector aan Howest Brugge, wil ik graag in de schijnwerpers zetten. Zonder de lessen over het schrijven van een scriptie die ik kreeg van haar zou dit niet gelukt zijn.

Verder wil ook nog een dankwoord richten aan alle personen die hebben meegewerkt aan het testen van de applicatie. Zonder hen zou ik geen goed onderzoek hebben kunnen uitvoeren.

Ik wens u veel leesplezier toe.

Thibaut Wittevrongel

Sint-Joris, juni 2023

Samenvatting

Het kiezen van een juiste architectuur voor een webapplicatie is niet altijd even gemakkelijk. Er zijn verschillende soorten architecturen die elk hun voor- en nadelen hebben.

In dit onderzoek wordt er gekeken naar de verschillen tussen een Single Page application (SPA) en een Multi Page application (MPA).

Er wordt gekeken naar de verschillen op vlak van gebruikerservaring, performance en development ervaring. Om deze verschillen te kunnen aantonen worden er twee applicaties ontwikkeld. Beiden met volledig dezelfde functionaliteit, maar elk ontwikkeld met een verschillende structuur.

Deze applicaties worden dan getest op verschillende vlakken. De resultaten van deze testen worden dan vergeleken om zo een antwoord te kunnen geven op de onderzoeksvraag: 'Single vs. Multi page applications: wanneer kies je wat?'.

Sleutelwoorden: MPA – SPA – performance – UX – webapplicatie

Abstract

Choosing the right architecture for a web application is not always easy. There are different types of architectures, each of which has its advantages and disadvantages.

This study looks at the differences between a Single Page application (SPA) and a Multi Page application (MPA).

It looks at the differences in terms of user experience, performance, and development experience. To demonstrate these differences, two applications are developed. Both with completely the same functionality but each developed with a different structure.

These applications are then tested in different areas. The results of these tests are then compared to answer the research question: 'Single vs. Multi page applications: when do you choose what?'.

Keywords: MPA - SPA - performance - UX - web application

Inhoudsopgave

Woord vooraf

Samenvatting

Abstract

Inhoudsopgave

Lijst met figuren

Lijst met tabellen

Verklarende woordenlijst

1	Inleiding.....	11
1.1	Algemeen	11
1.1.1	Wat is een Single page application?	11
1.1.2	Wat is een Multi page application?	12
1.2	Probleemstelling.....	13
1.3	Onderzoeksvraag.....	13
1.4	Experiment	13
1.5	Structuur van deze bachelorproef	13
2	Theoretisch kader/Literatuuronderzoek.....	14
3	Experiment.....	15
3.1	Methodologie.....	15
3.1.1	Uit welke elementen bestaat een typische webapplicatie?	15
3.1.2	Welke applicatie wordt er ontwikkeld?	15
3.1.3	De ontwikkelde API	18
3.1.4	Ontwikkelen van de MPA	19
3.1.5	Ontwikkelen van de SPA.....	22
3.2	Onderzoeksverloop en data-analyse methode(n).....	27
3.2.1	User experience testing.....	27
3.2.2	Verschillende soorten UX testen	27
3.2.3	UX testen binnen dit onderzoek	28
3.2.4	Performance testen	30
3.2.5	Performance indicators	31
3.2.6	Performance testing tools.....	31
3.3	Resultaten	32
3.3.1	Hoe verschilt de gebruikerservaring tussen een SPA en MPA?	32
3.3.2	Hoe verschilt de SPA en MPA qua performance?	32
3.3.3	Is er een groot verschil in development ervaring?.....	36
3.3.4	Hoe verloopt het deployen van een SPA en een MPA?.....	37
4	Conclusie.....	39
5	Discussie	40
	Referentielijst.....	41
	Lijst van bijlagen	43
	Bijlagen	44

Lijst met figuren

Figuur 1 demonstratie SPA lifecycle.....	12
Figuur 2 demonstratie MPA lifecycle	12
Figuur 3 wireframe beginpagina van de applicatie	16
Figuur 4 wireframe opslaan van een menu in de applicatie	16
Figuur 5 wireframe loginpagina van de applicatie	17
Figuur 6 wireframe ingrediëntenlijstpagina van de applicatie	18
Figuur 7 voorbeeld van de structuur van een URL	18
Figuur 8 indexpagina van de applicatie	20
Figuur 9 menuCreator pagina van de applicatie.....	20
Figuur 10 receptpagina van de applicatie.....	21
Figuur 11 savedMenus pagina van de applicatie	21
Figuur 12 ingrediëntenlijstpagina van de applicatie.....	22
Figuur 13 profielpagina van de applicatie.....	22
Figuur 14 browser compatibility van de template tag	24
Figuur 15 leeftijden testers	29
Figuur 16 grafiek vergelijking initieel ingeladen data.....	32
Figuur 17 grafiek vergelijking hoeveelheid data ingeladen na volledige flow door de applicatie	33
Figuur 18 aantal requests verkregen SPA.....	33
Figuur 19 aantal requests verkregen MPA	34
Figuur 20 grafiek vergelijking FPC	34
Figuur 21 grafiek vergelijking speed index	35
Figuur 22 grafiek vergelijking laadtijd op 3G-netwerk.....	36
Figuur 23 vergelijking marktaandeel van verschillende Linuxdistributies	38

Lijst met tabellen

Tabel 1 scenario's usertesten.....29

Lijst met afkortingen

API – Application programming interface

FPC – First contentful paint

MPA – Multi Page Application

SEO – Search engine optimization

SPA – Single Page Application

UX – User experience

VPS – Virtual Private Server

Verklarende woordenlijst

Webcrawler: Een webcrawler is een stuk software dat automatisch webpagina's bezoekt. Dit om informatie te verzamelen en vaak deze daarna te indexeren.

Wireframe: Een wireframe is een visuele representatie van een website of applicatie, waarin de lay-out, structuur en functionaliteit worden weergegeven zonder grafische elementen of styling.

1 Inleiding

1.1 Algemeen

Deze bachelorproef is tot stand gekomen vanuit een onderwerp dat is aangeboden uit het opleidingsteam Toegepaste Informatie van de Hogeschool West-Vlaanderen. Dit om een duidelijker antwoord te kunnen geven op de onderzoeksvraag 'Single vs. Multi page applications: wanneer kies je wat?'.
 Het onderstaande experiment gaat over het opzetten van desbetreffende applicaties om hierop een antwoord te formuleren.

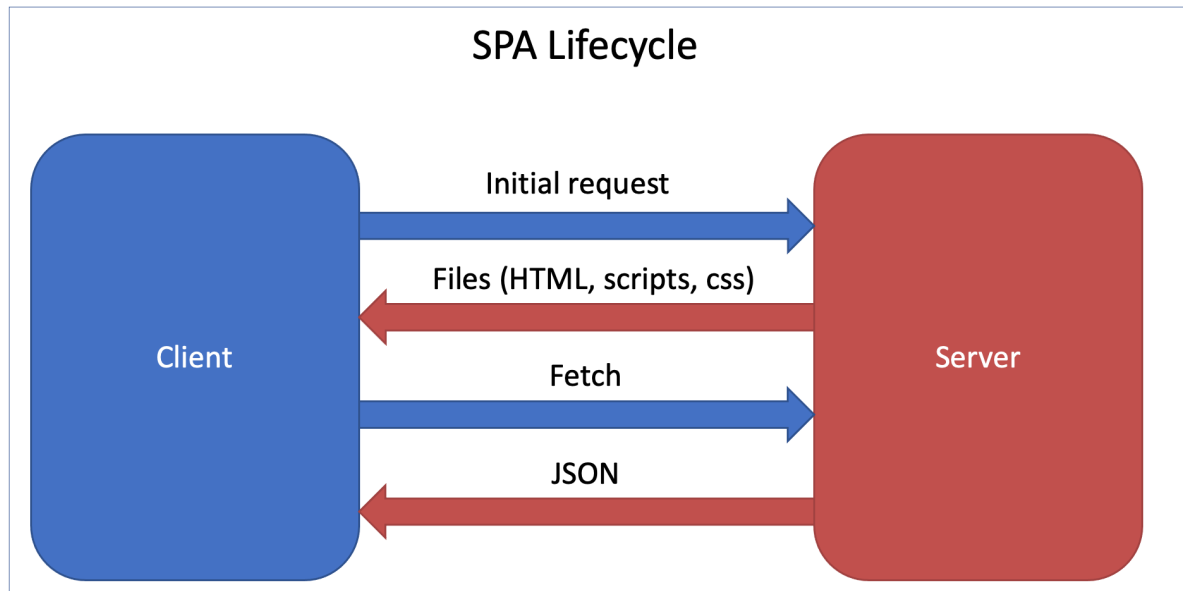
1.1.1 Wat is een Single page application?

Een Single Page Application, afgekort een SPA, is een website of webapplicatie die gemaakt is uit één webpagina. Wanneer een gebruiker surft naar een SPA krijgt de browser van de gebruiker alle broncode (HTML, JavaScript, CSS) binnen die nodig is om heel de website te laten werken tijdens het laden van de website. Wanneer deze SPA zeer groot of complex is kan dit ervoor zorgen dat de laadtijd die nodig is om deze code in te laden iets groter is dan een traditionele website. Maar na deze langere initiële laadtijd verloopt het navigeren op de site zelf wel veel sneller omdat er geen andere requests meer moeten gestuurd worden om verschillende pagina's op te halen.

De manier waarop de navigatie werkt op een SPA is door middel van JavaScript. Met JavaScript kan de inhoud van de website worden bijgewerkt en lijkt het alsof de gebruiker toch aan het werken is op verschillende pagina's. Dit terwijl de gebruiker zich nog altijd bevindt op de originele ingeladen pagina.

Als de SPA nood heeft aan data of inhoud die afkomstig is van een externe server kan er gebruik gemaakt worden van JavaScript om met bijvoorbeeld een fetch request een API of verschillende API's aan te spreken. De data die daarna ontvangen wordt kan dan verwerkt worden door JavaScript om de inhoud van de SPA aan te passen naar het formaat dat nodig is op de website. Dit allemaal zonder dat de website een reload hoeft uit te voeren.

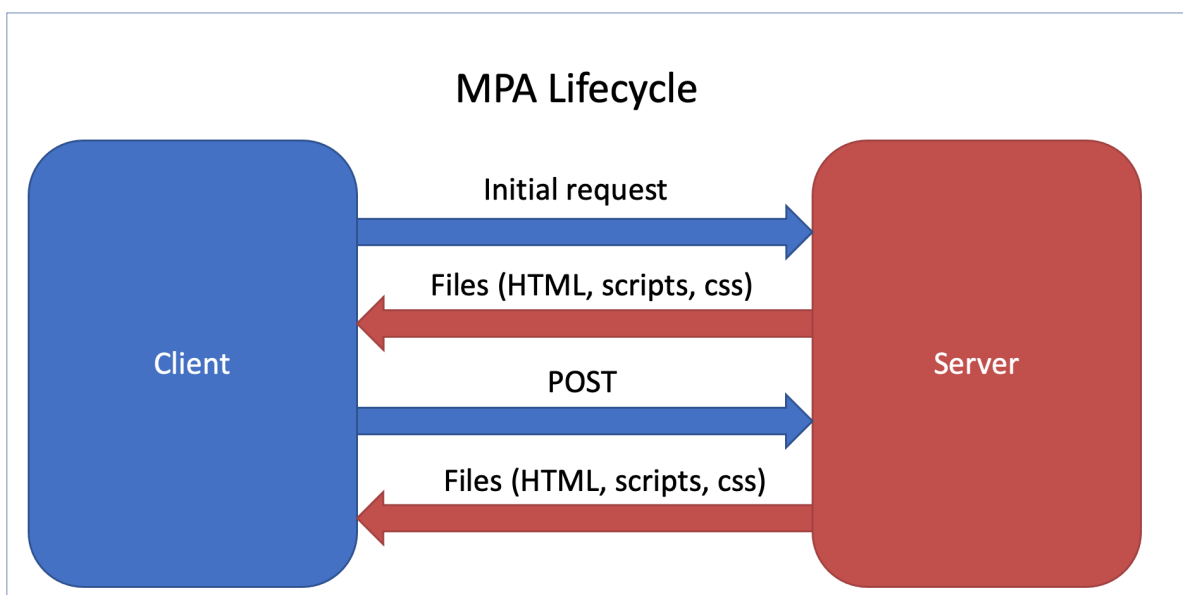
Single vs. Multi page applications
Inleiding



Figuur 1 demonstratie SPA lifecycle

1.1.2 Wat is een Multi page application?

Een Multi Page Application, afgekort een MPA, is een website die uit meerdere webpagina's bestaat. Wanneer een gebruiker surft naar een MPA krijgt de gebruiker de gevraagde pagina te zien en worden alle bestanden die nodig zijn om deze pagina te laten werken gedownload. Wanneer de gebruiker hierna naar een volgende pagina klikt tijdens het navigeren op de MPA downloadt de browser terug de volledige volgende pagina en alle bestanden die hiervoor nodig zijn. Dit zorgt ervoor dat er veel meer data tussen de gebruiker en de server verstuurd wordt.



Figuur 2 demonstratie MPA lifecycle

1.2 Probleemstelling

Het probleem dat zich voordoet, is dat het lastig is om te kiezen tussen single page applicaties of multi page applicaties als structuur voor een website. Dit is een vraag die vaak wordt gesteld door personen of organisaties die een website willen bouwen en bezig zijn met het maken van een beslissing over de structuur van de site. Als het probleem niet wordt opgelost, kan dit leiden tot ontevredenheid met de gebruikerservaring van de website, een lage prestatie van de site, hogere ontwikkelings- en onderhoudskosten en een afname van het aantal bezoekers op de site. Het is daarom belangrijk om een weloverwogen beslissing te maken over de structuur van de site om deze problemen te voorkomen.

1.3 Onderzoeksvraag

In deze bachelorproef wordt volgende onderzoeksvraag behandeld: “Single vs. Multi page applications: wanneer kies je wat?”. Om op deze vraag een goed antwoord te kunnen geven wordt deze opgedeeld in verschillende deelvragen:

1. Uit welke elementen bestaat een typische webapplicatie?
2. Wat is een Single page application?
3. Wat is een Multi page application?
4. Hoe verschilt de gebruikerservaring tussen een SPA en MPA?
5. Hoe verschilt de SPA en MPA qua performance?
6. Is er een groot verschil in development ervaring?
7. Hoe verloopt het deployen van een SPA en een MPA?

1.4 Experiment

Om een antwoord te formuleren op de onderzoeksvraag wordt er een experiment opgezet. In dit experiment worden er twee identieke applicaties ontwikkeld. Het enige verschil tussen de twee applicaties is dat de ene in de structuur van een MPA wordt gemaakt. De andere wordt gemaakt in de structuur van een SPA. Deze twee applicaties worden daarna vergeleken met elkaar om te zien welke voor- en nadelen beide structuren hebben. Dit om dan op het einde een conclusie te kunnen hebben over beide structuren.

1.5 Structuur van deze bachelorproef

Eerst zal het kort gaan over de geraadpleegde literatuur dat binnen dit onderzoek gebruikt is. Daarna volgt een hoofdstuk over hoe het experiment is opgezet. Vervolgens worden alle resultaten overlopen. Als laatste volgt de conclusie die een antwoord geeft op de hoofdvraag en een kort overzicht over de volgende stappen van dit onderzoek.

2 Theoretisch kader/Literatuuronderzoek

Om op de hoogte te zijn over de al uitgevoerde onderzoeken over MPA's vs. SPA's zijn er tijdens dit onderzoek verschillende artikels doorgenomen.[1]–[8]

In de meeste artikels worden vaak dezelfde zaken vergeleken met elkaar. Dit gaat dan vooral over de gebruikservaring, performance, development ervaring en de positieve en negatieve kanten van een MPA of SPA.

Een veel voorkomend onderwerp dat niet wordt aangehaald binnen dit onderzoek is search engine optimization (SEO). Een SPA zou namelijk veel minder goed scoren op vlak van SEO dan een MPA. Dit komt doordat de inhoud van een MPA statisch is waardoor het makkelijker is voor webcrawlers om de website een score te geven. Ook kunnen verschillende pagina's van een MPA betere scores krijgen waardoor deze hoger in de search engine komen te staan dan de enkele SPA-pagina.

Om de scope van dit onderzoek te beperken is er in binnen dit onderzoek niet dieper op SEO ingegaan.

Vervolgens werd tijdens dit onderzoek ook de MDN-documentatie geraadpleegd. Deze website geeft veel informatie rond allerlei zaken binnen web development.[9]

3 Experiment

3.1 Methodologie

3.1.1 Uit welke elementen bestaat een typische webapplicatie?

Om te kunnen beginnen aan dit experiment moet er eerst gedacht worden aan de elementen waaruit een typische webapplicatie bestaat. Dit zodat de applicaties die ontwikkeld worden tijdens dit onderzoek zo dicht mogelijk tegen applicaties liggen die te vinden zijn op het web.

Het eerste element waaruit een webapplicatie bestaat is de effectieve content die een bedrijf of persoon wil tonen. Dit zijn bijvoorbeeld pagina's met informatie over een product of een bepaald onderwerp. Deze pagina's zijn dan vooral opgebouwd uit tekstelementen en foto's.

Een volgend element dat veel voorkomt op verschillende websites is een vorm van navigatie. Dit kan bijvoorbeeld een navigatiemenu zijn of een knop die een user navigeert naar een ander deel van de website.

Een vorm van user input is een volgend element dat op zeer veel sites terugkomt. Dit kan bijvoorbeeld tekst input, een lijst checkboxen of een formulier zijn.

Ten vierde is een vorm van authenticatie. Veel websites hebben een manier om hun users te kunnen herkennen en om persoonlijke content van gebruikers bij te houden. Hier hoort dan ook natuurlijk een manier bij om als gebruiker een eigen account aan te maken.

Heel veel websites bestaan dus uit bovenstaande elementen. Daarom zal er in dit onderzoek zeker voor gezorgd worden dat alle elementen die vernoemd zijn aanwezig zijn in de applicaties.

3.1.2 Welke applicatie wordt er ontwikkeld?

Een website wordt natuurlijk niet ontwikkeld zonder een bepaalde use case. Daarom is er in dit onderzoek gedacht aan een onderwerp dat in beide een SPA en MPA gegoten kan worden. Belangrijk is dat het idee voldoende breed genoeg is om ervoor te zorgen dat alle elementen van een typische webapplicatie toepasbaar zijn.

Het concept van de webapplicatie is dat de gebruiker een lijst van maaltijden kan opvragen die gebruikt kan worden als weekmenu. Na dat een weekmenu is opgevraagd kan de gebruiker de recepten bekijken van elke maaltijd. Ook een boodschappenlijst met alle ingrediënten van het weekmenu kan worden opgevraagd in de applicatie.

Wanneer de applicatie geopend wordt is het eerste element dat een gebruiker zal zien een formulier. In dit formulier kan de gebruiker aangeven van welke maaltijdperiodes de gebruiker maaltijden wil ontvangen. Daarna kan ook het aantal maaltijden per periode ingegeven worden. Na het verzenden van het formulier vindt ook de allereerste navigatie plaats. De gebruiker wordt doorgestuurd naar een pagina met gegenereerde maaltijden.

Single vs. Multi page applications Experiment

Insert Title of APP

Login

Select the type of recipes you want to receive and how many per type.

☐ Breakfast ☐ Lunch ☐ Diner

5

Generate Menus

Figuur 3 wireframe beginpagina van de applicatie

Op deze pagina kan de gebruiker aan de hand van knoppen maaltijden opnieuw genereren en navigeren naar het recept van de maaltijd. Wanneer de gebruiker blij is met het volledige menu kan deze opgeslagen worden. Aan een menu kan er een naam meegegeven worden, hier bevindt zich een tekstinput.

Insert Title of APP

Thibaut

Recipes: Breakfast

Image Avocado Toast

Recipes: Lunch

Image Wrap with

Recipes: Diner

Image Steak with fries & salade

Name of save:

Week-menu1

Save

Save menu

Figuur 4 wireframe opslaan van een menu in de applicatie

Single vs. Multi page applications Experiment

Wanneer een menu wordt opgeslagen wordt er gecontroleerd of de gebruiker ingelogd is. Als dit niet het geval is wordt de gebruiker doorgestuurd naar de login pagina. Hier kan dan ingelogd of een nieuw account aangemaakt worden. Authenticatie wordt hier gebruikt om opgeslagen menu's te linken aan de gebruiker.

Insert Title of APP

Login

Login

Username:

Thibaut

Password:

◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇

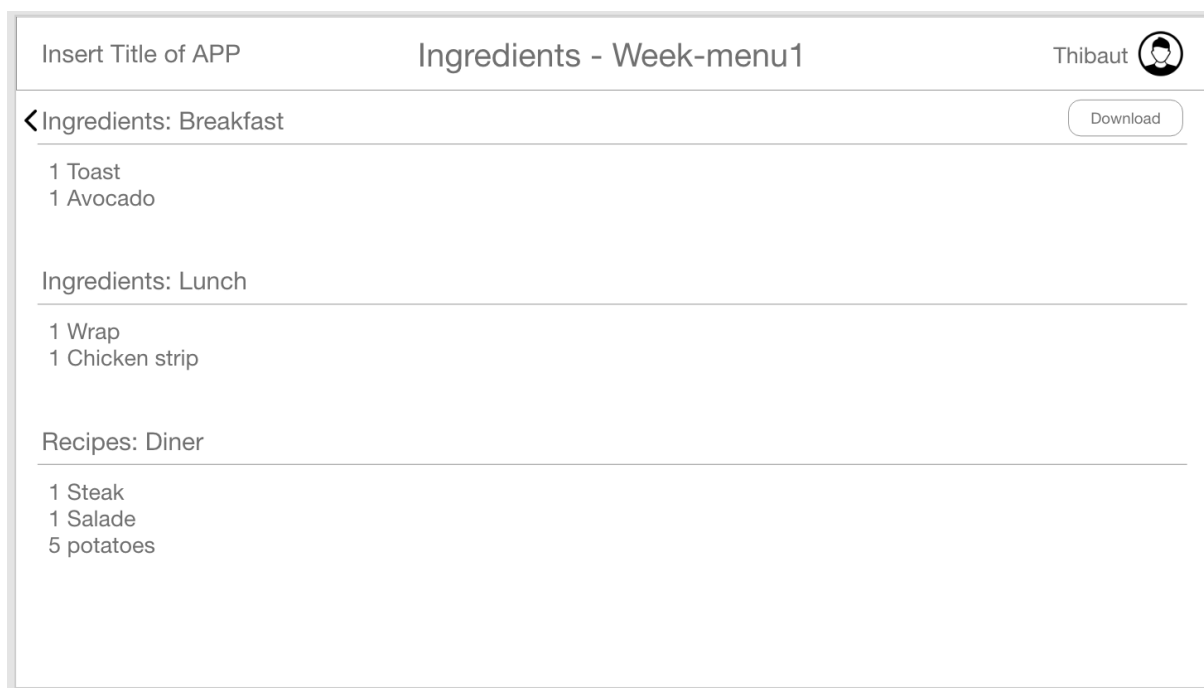
Login

Don't have an account? [Register here!](#)

Figuur 5 wireframe loginpagina van de applicatie

Na dat een menu aangemaakt is kan een gebruiker deze opnieuw opvragen via de profielpagina. Op deze pagina is er ook de mogelijkheid om uit te loggen. Wanneer een opgeslagen menu geopend wordt kunnen alle recepten terug geraadpleegd worden. Ook is er een knop die wanneer erop geklikt wordt de user navigeert naar de ingrediëntenpagina. Deze pagina bevat alle ingrediënten die nodig zijn om heel het menu te maken. Deze lijst kan daarna gedownload worden.

Single vs. Multi page applications Experiment



Figuur 6 wireframe ingrediëntenlijstpagina van de applicatie

3.1.3 De ontwikkelde API

Om authenticatie en externe data beschikbaar te maken in een webapplicatie wordt er vaak een Application Programming Interface (API) gebruikt. Een API is eigenlijk een stuk technologie die ervoor zorgt dat applicaties met elkaar kunnen communiceren. In dit onderzoek zorgt de opgezette API er dus voor dat er communicatie tussen de webapplicatie en de server mogelijk is.

Er zijn tal van manieren om een API op te zetten. Zo is er de keuze om een Simple Object Acces Protocol (SOAP), Remote Procedure Call (RPC) of een Representational State Transfer (REST) API te maken. In dit onderzoek is er gebruik gemaakt van de REST-structuur.[10]

De gekozen programmeertaal om deze API mee te ontwikkelen is PHP. Meer specifiek met het Laravel framework. De reden om Laravel te kiezen was omdat deze uitgebreid tijdens de opleiding is gezien en dit framework alle mogelijkheden aanbiedt die nodig is voor dit onderzoek.

Een API bestaat uit verschillende endpoints. Deze endpoints worden allemaal opgenomen in een API-specificatie. Aan de hand van dit document kunnen developers weten welke endpoints er toegankelijk zijn en welke parameters eventueel meegeven moeten worden.

Een endpoint ziet er als volgt uit:

Base-url End-point Query 1 Query 2

<https://api.thibautwittevrongel.com/api/recipes?mealtime=breakfast&amount=1>

Figuur 7 voorbeeld van de structuur van een URL

Zoals te zien op de afbeelding bestaat een endpoint uit volgende elementen:

- Base-URL: Dit is de URL van de server waarop de API gehost staat.
- Endpoint: Dit is het pad dat gedefinieerd staat binnen de API-specificatie om een bepaalde actie uit te voeren.
- Query's: Aan een endpoint kunnen vaak query's meegegeven worden. De eerste query wordt altijd gestart met een vraagteken. Daarna als er nog query's toegevoegd moeten worden kan dit gedaan worden met een ampersand als scheidingsteken.

Hieronder een kort overzicht van alle endpoints die ontwikkeld zijn in de Laravel API:

- POST /login
Deze endpoint wordt gebruikt om de gebruiker te laten inloggen. De developer kan via dit request de gebruikersnaam en het wachtwoord meesturen. Daarna geeft de API een response terug met een JWT-token als het request succesvol is of met een error code als het mislukt is.
- POST /logout
Met het /logout endpoint kan de developer de gebruiker laten uitloggen. Wanneer dit request binnenkomt in de API verwijdert deze de JWT-token.
- GET /recipes/{ID}
Wanneer de developer deze endpoint oproept, geeft de server het recept terug dat het meegegeven ID bevat.
- GET /recipes/random/{params}
Met dit endpoint kan er door middel van parameters een lijst van willekeurige recepten als response teruggestuurd worden.
- GET /menus
Dit endpoint kan gebruikt worden om alle opgeslagen menu's van een gebruiker op te halen.
- POST /menus
Met de POST-versie van het /menus endpoint kan er een nieuw menu aangemaakt worden voor een bepaalde gebruiker.
- GET /recipes/in-menu/{menu-ID}
Dit endpoint kan gebruikt worden om alle recepten van een menu op te halen.
- GET /recipes/in-menu/{menu-ID}/ingredients
Met dit endpoint is het mogelijk om alle ingrediënten die nodig zijn om een bepaald menu klaar te maken op te halen.

Deze API wordt ook gehost op dezelfde server als waar de SPA en MPA staan.

3.1.4 Ontwikkelen van de MPA

De eerste webapplicatie die ontwikkeld wordt tijdens dit onderzoek is de MPA. Deze applicatie wordt alleen ontwikkeld met HTML, SCSS en JavaScript. De reden hiervoor is om te zorgen dat frameworks of externe library's de applicatie niet zouden kunnen beïnvloeden.

Single vs. Multi page applications Experiment

De MPA is opgebouwd uit negen verschillende HTML-pagina's. De eerste pagina die wordt ingeladen is de indexpagina. Deze pagina bestaat uit een navigatie-element naar de loginpagina en een formulier om een menu te genereren.

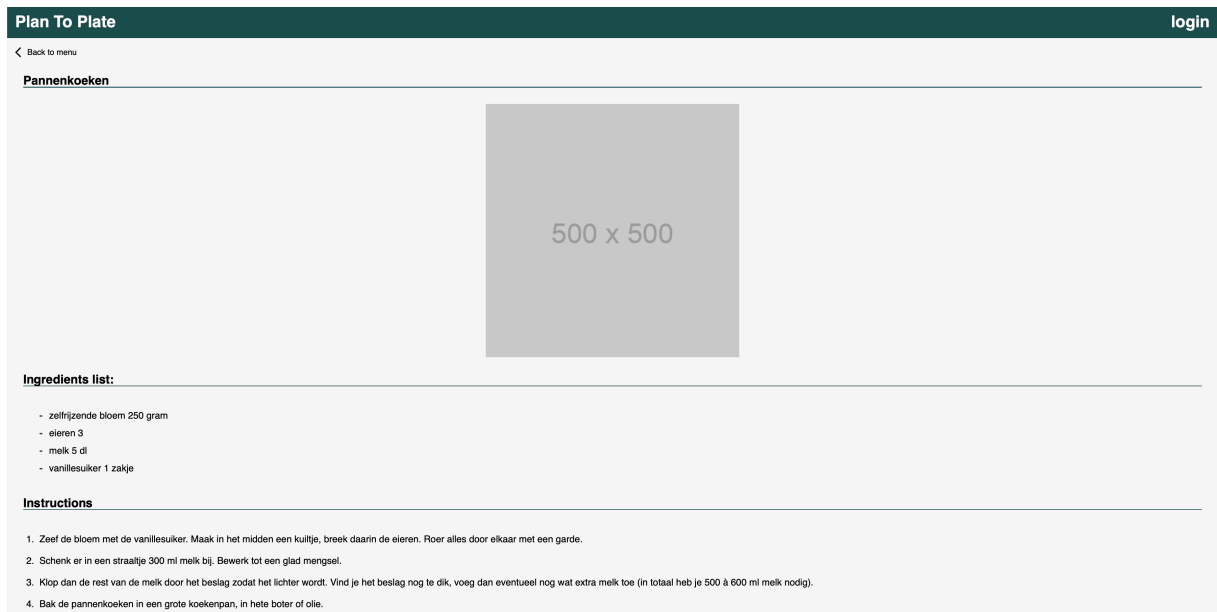
Figuur 8 indexpagina van de applicatie

Nadat een gebruiker een menu genereert op de indexpagina wordt de gebruiker doorgestuurd naar de tweede pagina. De tweede pagina is de menuCreator pagina. Hier kan een gebruiker een menu aanpassen naar eigen voorkeuren en kunnen de recepten van het menu geraadpleegd worden.

Figuur 9 menuCreator pagina van de applicatie

Wanneer een recept geraadpleegd wordt opent de derde pagina. Dit is de receptpagina van het aangeklikte gerecht.

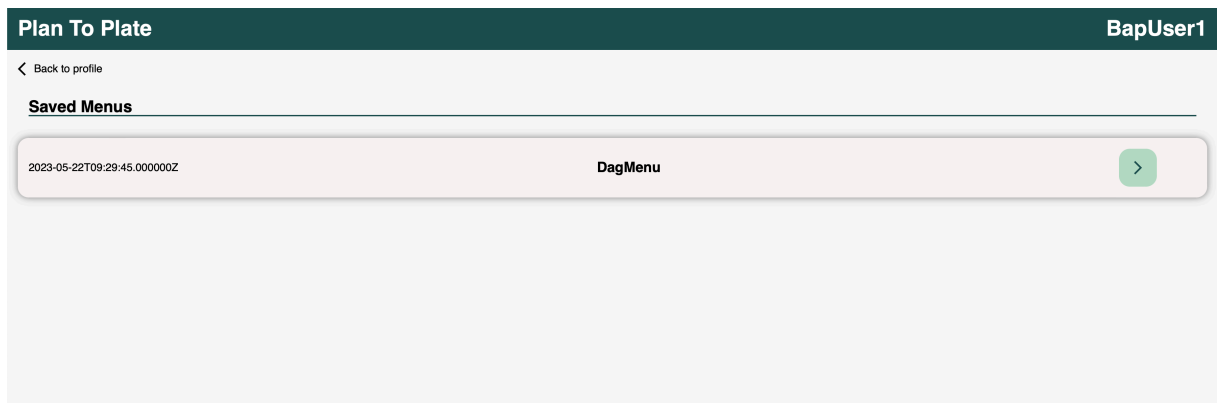
Single vs. Multi page applications Experiment



Figuur 10 receptpagina van de applicatie

De gebruiker kan een menu opslaan op de menuCreator pagina. Hiervoor moet een gebruiker wel ingelogd zijn. Als dit niet geval is wordt de gebruiker automatisch doorgestuurd naar de login of registerpagina.

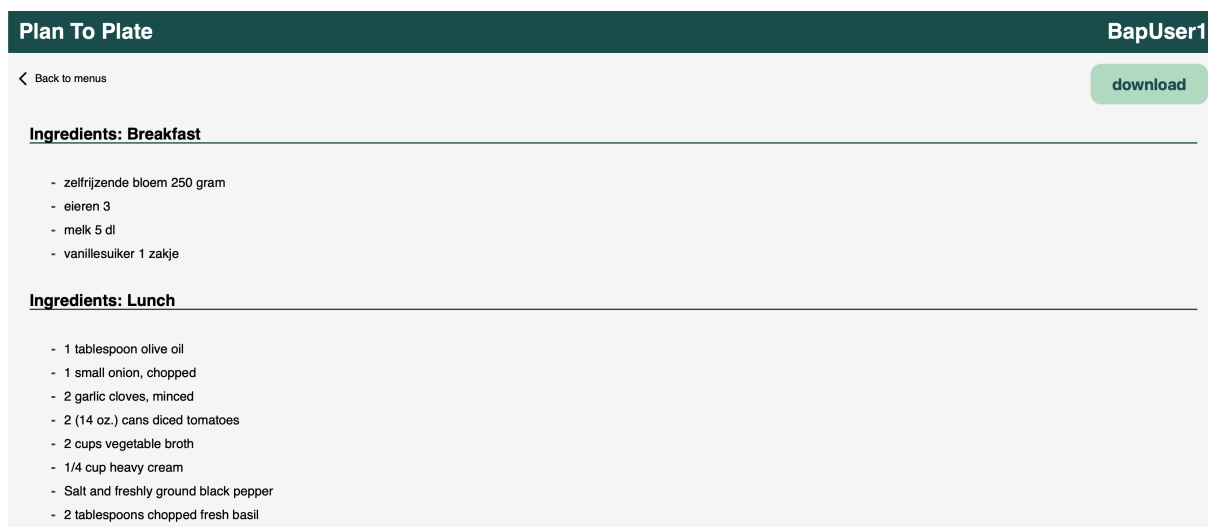
Wanneer de gebruiker hierna succesvol een account aanmaakt of inlogt, komt de savedMenus pagina tevoorschijn. Hier vindt een gebruiker alle opgeslagen menu's terug.



Figuur 11 savedMenus pagina van de applicatie

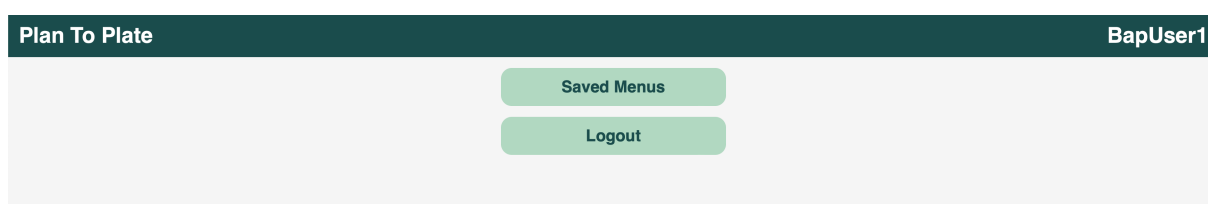
Een menu kan hier dan open geklikt worden en alle recepten kunnen hier opnieuw geraadpleegd worden. Deze pagina heet de menupagina. Hier heeft de gebruiker ook de mogelijkheid om een ingrediëntenlijst te openen en deze eventueel te downloaden.

Single vs. Multi page applications Experiment



Figuur 12 ingrediëntenlijstpagina van de applicatie

Als een gebruiker de opgeslagen recepten wil raadplegen kan dit ook via de profielpagina. Hier heeft de gebruiker de mogelijkheid om uit te loggen en om de opgeslagen menu's op te halen.



Figuur 13 profielpagina van de applicatie

Bij de MPA van dit onderzoek is gekozen om geen verschillende pagina's te maken per recept. Dit om de scope van het experiment niet te groot te maken. Daarom is er gekozen om op de receptpagina een generale structuur op te bouwen die daarna ingevuld wordt met de data die uit de API verkregen wordt.

3.1.5 Ontwikkelen van de SPA

De tweede webapplicatie die ontwikkeld wordt tijdens dit onderzoek is de SPA. Net zoals de MPA is deze volledig ontwikkeld met HTML, SCSS en JavaScript.

De SPA bestaat uit een enkele HTML-pagina. Met behulp van JavaScript kan de inhoud van de pagina veranderd worden. Dit zorgt ervoor dat de gebruiker het gevoel heeft dat er verschillende pagina's zijn terwijl de gebruiker zich nog altijd op dezelfde pagina bevindt.

Bij een SPA komt er qua development wel meer bij kijken. Zo moet er bijvoorbeeld veel meer rekening gehouden worden met de flow van de applicatie. Dit omdat de gebruiker niet meer kan navigeren naar een andere pagina. Dit moet allemaal gebeuren door middel van JavaScript. Ook moet er rekening gehouden worden dat de back- en forwardknoppen van de browser niet meer werken zoals bij een MPA. Nu moet dit door de developer volledig zelf bijgehouden worden.

Single vs. Multi page applications

Experiment

Bij een SPA zijn er drie vragen die gesteld kunnen worden:

1. Hoe kan de HTML aangepast worden?
2. Hoe kan de navigatie behouden worden net zoals een MPA?
3. Hoe kan er data van de server gebruikt worden binnen de SPA?

Om de HTML aan te passen in een webapplicatie zijn er verschillende methoden. Zo is een methode om met queryselectors in JavaScript bepaalde elementen te verwijderen of aan te passen. Ook is het mogelijk om met JavaScript nieuwe HTML-elementen te creëren en deze dan in het HTML-document te plaatsen.

Maar er bestaat een relatief nieuwe manier om een document aan te passen. Deze manier maakt gebruik van HTML template elementen. Een template-element kan worden gebruikt om HTML die nog niet direct nodig is op de webpagina op te slaan. De MDN-documentatie[11] definieert de template tag als volgt:

“Het <template> HTML-element is een mechanisme voor het opslaan van HTML die niet onmiddellijk moet worden weergegeven wanneer een pagina wordt ingeladen, maar die later tijdens runtime met behulp van JavaScript kan worden geconcretiseerd.

Zie een <template> als een inhoudsfragment dat wordt opgeslagen voor later gebruik in het document. Hoewel de parser de inhoud van het <template>-element verwerkt tijdens het laden van de pagina, doet hij dat alleen om ervoor te zorgen dat die inhoud geldig is; de inhoud van het element wordt echter niet weergegeven.”

Met deze template-elementen kan er op een propere manier HTML bijgehouden worden. Een voorbeeld hoe dit gebruikt is tijdens dit onderzoek gaat als volgt.

In de applicatie wil de gebruiker inloggen. Deze HTML staat momenteel niet zichtbaar in de browser, maar zit mee in de HTML als een template tag.

```
<template id="loginFormTemplate">
  <form action="index.html" method="post" class="auth-form">
    <label for="email">Email</label>
    <input type="text" name="email" id="email">
    <label for="password">Password</label>
    <input type="password" name="password" id="password">
    <input type="submit" class="button" value="Login">
    <p>Don't have an account? <a href="#register">Register here!</a></p>
  </form>
</template>
```

Wanneer de gebruiker een bepaalde actie uitvoert, is het mogelijk om via JavaScript de HTML in dit template-element zichtbaar te maken. Een voorbeeld hiervan is volgende code:

```
function showLoginForm(){
  clearMain();
  const loginFormTemplate = document.querySelector("#loginFormTemplate");
  const loginFormHtml = loginFormTemplate.content.firstElementChild.cloneNode(true);
```

```
loginFormHtml.querySelector("a").addEventListener("click", showRegisterForm);
loginFormHtml.querySelector('input[type="submit"]').addEventListener('click', login);

document.querySelector("main").appendChild(loginFormHtml);
}
```

Het eerste wat gebeurt wanneer deze functie wordt opgeroepen is dat de functie 'clearMain()' wordt uitgevoerd. In dit project zorgt deze functie dat het HTML-element 'main' volledig leeg gemaakt wordt.

Daarna wordt met behulp van een querySelector het gewenste template-element opgehaald. Hierna wordt de inhoud van het template-element, de HTML die zichtbaar gemaakt moet worden, gekloond in een andere variabele.

Vervolgens worden er aan de verschillende elementen binnen de opgehaalde HTML eventListeners toegevoegd. Als laatste wordt de nieuwe HTML toegevoegd aan het main-element.

Deze manier van werken kan worden toegepast op elk stuk HTML die pas later zichtbaar mag zijn in de flow van de applicatie. Ook wordt de template tag volgens de MDN-documentatie op alle browsers ondersteund.

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Desktop					Mobile					
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
template	✓ 26	✓ 13	✓ 22	✓ 15	✓ 8	✓ 26	✓ 22	✓ 14	✓ 8	✓ 1.5	✓ Yes

Figuur 14 browser compatibility van de template tag

Een tweede topic die zeker aangehaald moet worden bij het ontwikkelen van een SPA is de navigatie. Sinds er maar één HTML-pagina is zal de standaard navigatie die werkt binnen een MPA niet werken in dit geval.

Een oplossing voor dit probleem is om gebruik te maken van de History API.[12] Deze API geeft toegang tot de browsers session history. Dit betekent dat een developer met behulp van deze API zelf volledig de session history kan bepalen en de navigatie regelen binnen een SPA. Wanneer er gebruik gemaakt wordt van de History API en vanaf er sessions worden opgeslagen in de history werken de back- en forwardknoppen zoals ze in een MPA werken.

Single vs. Multi page applications

Experiment

Aan de hand van een voorbeeld uit de code van de SPA zal uitgelegd worden hoe de History API gebruikt kan worden.

Het allereerste wat moet gedaan worden om te werken met de History API is het maken van een functie. Deze functie zal de pagina waarvan weg genavigeerd wordt opslaan. Dit kan met volgende code:

```
function savePreviousPage(nameOfPage, url){
    const currentState = {page: nameOfPage}
    window.history.pushState(currentState, "", url);
}
```

Deze code neemt als parameters de naam van de pagina en de URL van deze pagina. Daarna wordt de naam van de pagina opgeslagen in een object binnen een variabele. Hierna wordt dan de History API aangesproken. De functie pushState zorgt ervoor dat de pagina toegevoegd wordt aan de browsers history session.[13] Deze functie neemt drie parameters:

1. State, dit is een willekeurig JavaScript object dat de developer kan opslaan in de browsers history stack.
2. Unused, de tweede parameter die de functie verwacht, wordt eigenlijk niet meer gebruikt. Deze is er wel nog door historische redenen. De veiligste manier om hier mee om te gaan is door een lege string mee te geven.
3. URL, de laatste parameter die kan meegegeven worden is een URL. Deze URL hoeft niet absoluut te zijn. Wanneer het een relatieve URL zou zijn, wordt deze tegenover de huidige URL toegepast.

Na dat deze functie uitgevoerd is staat de data opgeslagen binnen de browsers history stack. Deze functie wordt nu gebruikt bij elke navigatie tussen de verschillende delen van de website.

Nu moeten er nog twee functies voorzien worden om de volledige flow van de navigatie te laten werken.

Het eerste wat moet gebeuren is een evenListener toevoegen aan de browser window. Dit kan als volgt:

```
window.addEventListener('popstate', (event) => {
    replaceMain(event.state.page);
});
```

Wat in bovenstaande code gebeurt, is dat aan de browser window een eventListener gehangen wordt. Deze luistert naar het event 'popstate'. Het 'popstate' event wordt getriggerd elke keer dat er een verandering is met de history stack van de browser. Wanneer dit gebeurt, wordt de replaceMain functie uitgevoerd.

De replaceMain functie ziet er als volgt uit:

```
function replaceMain(page){
    if(page === "menuCreator"){
        initialiseMenuCreator();
    }
}
```

Single vs. Multi page applications

Experiment

```

    } else if (page === "profile"){
        showProfileScreen();
    } else if (page === "savedMenus"){
        initialiseSavedMenus();
    }
    else if (page === "viewSavedMenu"){
        initialiseViewMenu();
    }
}

```

Uit de vorige functie wordt aan bovenstaande functie de paginanaam meegegeven. De replaceMain functie loopt dan alle mogelijkheden af om een bepaalde actie uit te voeren. Dit alles samen zorgt ervoor dat de navigatie hetzelfde werkt als bij een MPA.

Een derde topic binnen een SPA is het verkrijgen van de data. Dit kan door middel van de Fetch API.[14] Met de fetch methode kan binnen een webapplicatie data opgehaald worden van een externe server. Aan dit request kunnen dan ook headers meegegeven worden. Hieronder een voorbeeld over hoe een fetch request eruitziet.

```

async function APIgetMenuRecipes(id){
    return fetch(`${API_URL}/recipes/in-menu/${id}`, {
        method: "GET",
        headers: {
            "Content-Type": "application/json",
            "Authorization": `Bearer ${token}`
        }
    })
    .then(response => response.json())
    .then(data => { return data; });
}

```

Binnen een fetch request wordt een URL meegegeven. Ook een object met aangepaste instellingen kunnen meegegeven worden zoals bijvoorbeeld: method, headers, body ...

Vervolgens krijgt het fetch request een response terug en kan daarna deze response verwerken.

Al deze bovenstaande technieken zijn in dit onderzoek gebruikt. Dit heeft ervoor gezorgd dat de SPA een zeer gelijkaardige functionaliteit heeft aan de MPA.

3.2 Onderzoeksverloop en data-analyse methode(n)

3.2.1 User experience testing

Tijdens dit onderzoek worden er user experience (UX) testen afgenomen. UX testen of usability testen zijn eigenlijk processen waarin een product of een deel van een product getest wordt op echte personen.

UX testen evalueren nu eenmaal hoe gebruikers omgaan met de service of het product. Tijdens een UX test wordt er gevraagd aan een tester om enkele taken uit te voeren. Terwijl een tester deze taken uitvoert kan door een onderzoeker bekeken worden hoe de tester reageert op bepaalde zaken. Dit kan bijvoorbeeld zijn op het design van een website of op hoe bepaalde componenten van een website reageren op elkaar.[15]

Nu wanneer zijn UX testen het best van toepassing? Dit verschilt van project tot project. Zo zijn er projecten waar er al testen gebeuren tijdens het designen van een nieuw product of feature. Dit om direct al verder te kunnen werken met het inzicht van echte personen.

Tijdens het ontwikkelen van een product kunnen UX testen ook afgenomen worden. Dit kan ervoor zorgen dat tijdens het ontwikkelen nog aanpassingen gemaakt kunnen worden voor een product uitgebracht wordt.

Als laatste kunnen testen ook altijd na een product release uitgevoerd worden. Dit kan om eventuele pijnpunten uit de applicatie aan te pakken die pas later naar boven komen.

3.2.2 Verschillende soorten UX testen

Er zijn verschillende soorten UX testen die kunnen uitgevoerd worden. Deze kunnen opgesplitst worden in volgende types.[16]

- Gemodereerd vs. Niet gemodereerd
- Op afstand vs. In persoon
- Exploratief vs. Vergelijkend

Gemodereerd + in persoon

Bij een gemodereerde test die persoonlijk afgenomen wordt is er altijd een onderzoeker aanwezig. Een voordeel hiervan is dat op het einde van de test de onderzoeker iets specifieker kan doorvragen op de ervaring van de gebruiker.

Een voorbeeld van dit soort test is de 'Lab usability test'. In deze test wordt een toestel klaar-gezet en altijd op dezelfde manier ingesteld. Hierna kan een tester de test uitvoeren onder toezicht van een onderzoeker. Dit zorgt ervoor dat de testen op een meer gestandaardiseerde manier verlopen.

Een ander voorbeeld van dit soort test is de 'Guerrilla test'. Hierbij gaat een onderzoeker op een publieke plaats zitten. Dit kan bijvoorbeeld een shoppingcenter, een treinstation of een luchthaven zijn. Daar probeert de onderzoeker willekeurige mensen te overtuigen om een user test uit te voeren. Dit is een goede manier om veel testdata te verkrijgen van verschillende profielen. Maar er moet wel goed gelet worden op de tijd. De meeste mensen spenderen niet graag meer dan vijf tot tien minuten van hun tijd hieraan.

Niet gemodereerd + op afstand

Bij een niet gemodereerde test op afstand wordt er meestal gebruik gemaakt van software om de test af te nemen. Bij deze soort testen hoeft geen onderzoeker aanwezig te zijn. Deze testen kunnen zeer grootschalig ondernomen worden.

Een voorbeeld van dit soort test is met behulp van opname software. Wanneer een tester een test start wordt er opname software opgestart. Deze software maakt dan een opname van alle acties die de gebruiker uitvoert. Vanaf dat de tester dan klaar is, wordt deze opname doorgestuurd naar de onderzoekers.

Niet gemodereerd + in persoon

Niet gemodereerde persoonlijke tests worden uitgevoerd in een gecontroleerde omgeving. Hier hoeft in principe geen persoon aanwezig te zijn om de test af te nemen. Dit biedt als voordeel dat de onderzoeker zelf minder invloed heeft op de test.

Een voorbeeld van dit soort test is een observatie test. Hierbij krijgt een tester een document met vooropgestelde scenario's die uitgevoerd moeten worden. Hierbij kijkt de observator naar hoe de tester de scenario's uitvoert en wat de lichaamstaal van de tester is. Ook kan de observator niet helpen tenzij de tester volledig de verkeerde kant opgaat.

Een ander voorbeeld van dit soort test is een eye-tracking test. Deze test gebruikt software om de ogen van een tester te volgen doorheen het gebruik van de website. Daarna wordt er een volledig rapport opgemaakt waar de tester het meeste gekeken heeft en wat de eerste elementen zijn die in het zicht springen.

3.2.3 UX testen binnen dit onderzoek

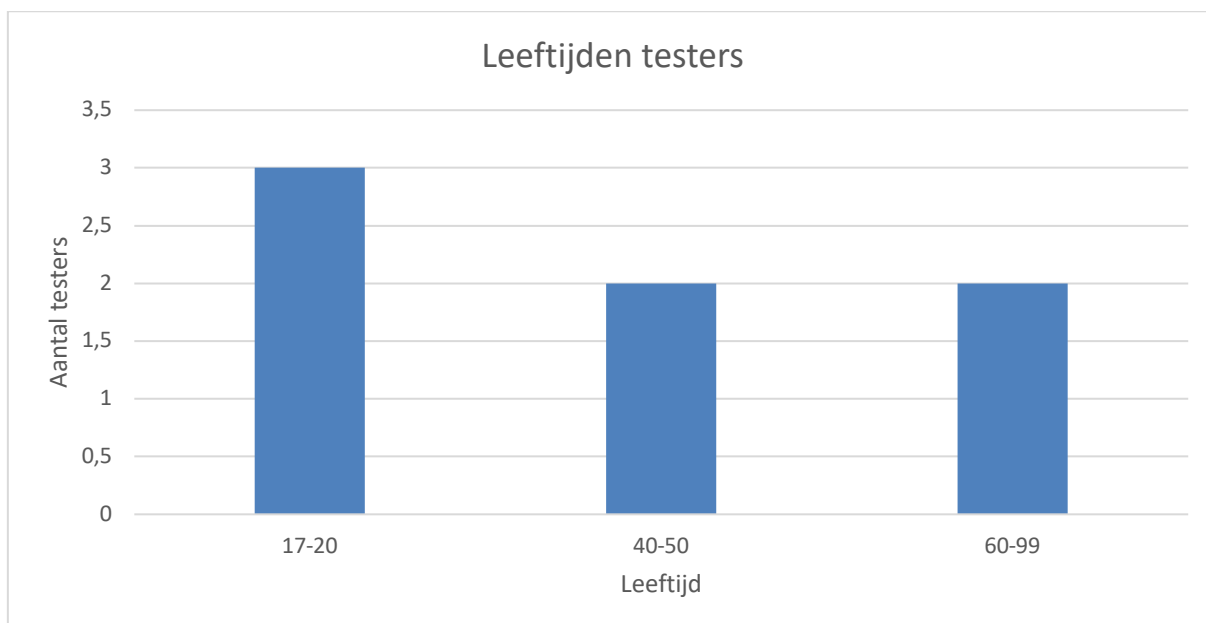
Binnen dit onderzoek is er gebruik gemaakt van de observatiemethode beschreven in het vorige hoofdstuk. De keuze om deze methode te gebruiken is gebaseerd op de mogelijkheid die het biedt om een gelijke testomgeving te creëren voor alle testen. Daarnaast was er al een gevarieerd testpubliek beschikbaar om de observaties uit te voeren.

In dit onderzoek werden er zeven personen bevraagd om deze usertesten uit te voeren. Een belangrijk aspect van usertesten is om ervoor te zorgen dat er binnen het testpubliek een grote variatie van leeftijden en beroepen is. Dit om ervoor te zorgen dat een groot deel van de bevolking gemakkelijk gebruik kan maken van de applicatie.

Binnen dit onderzoek bestaat de testersgroep uit volgende leeftijden:

- 3 personen tussen 17 en 20 jaar oud.
- 2 personen tussen 40 en 50 jaar oud.
- 2 personen boven de 60 jaar oud.

Single vs. Multi page applications
Experiment



Figuur 15 leeftijden testers

Het testpubliek bevat ook volgende beroepen:

- Student informatica;
- Student humane wetenschappen;
- Student lager onderwijs;
- Strijkhulp;
- Metser;
- Verpakkingsmedewerker;
- Magazijnier.

Binnen dit onderzoek zijn er drie scenario's opgesteld die de testpersonen moesten uitvoeren. Deze scenario's zijn de volgende:

Tabel 1 scenario's usertesten

Test scenario's	
Scenario 1	<p>Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.</p> <p>Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.</p>
Scenario 2	<p>Je hebt al een account op de website met volgende gegevens:</p>

Single vs. Multi page applications

Experiment

	<p>Gebruikersnaam: Tester</p> <p>Wachtwoord: test123</p> <p>Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje, maar niets en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.</p>
Scenario 3	<p>Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.</p>

Alle testers hadden uitgebreid de tijd om deze drie scenario's uit te voeren. Dit onder toezicht van een onderzoeker. De onderzoeker zorgde alleen voor eventuele hulp als een tester de weg kwijt was in de applicatie en niet meer verder kon in het scenario.

Om ervoor te zorgen dat er geen bevooroordeling ontstond voor een bepaalde structuur startte niet alle testers met dezelfde applicatie. De helft van de testers startte met de MPA en daarna de SPA en dan andere helft startte met de SPA.

Op het einde van de scenario's kregen de testers ook een finale vraag voorgeschoteld. Deze vraag gaat als volgt: 'Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?'

3.2.4 Performance testen

Een belangrijk aspect binnen het maken van een applicatie is het kiezen van een architectuur dat efficiënt werkt. De reden ligt hier dicht bij de gebruikerservaring. Volgens een artikel van Hostinger[17] verlaat een gebruiker een website als deze langer dan 3 seconden duurt om in te laden. [18]

Om een duidelijk beeld te scheppen over hoe de performance verschilt tussen de MPA en SPA worden er performancetesten afgenomen. De data uit deze testen kan dan naast elkaar gelegd worden om te zien op welke vlakken de MPA en SPA beter scoren in opzicht van elkaar.

3.2.5 Performance indicators

Om een website te kunnen testen op performance is het belangrijk om te weten welke indicatoren er belangrijk zijn. In dit hoofdstuk wordt ingegaan op een paar van deze indicatoren om een duidelijker beeld te scheppen over wat deze inhouden.[19]

Load time + grootte van de website

De allereerste indicatoren hangen erg nauw samen en dit zijn de load time en de grootte van de website. Hoe groter de bestanden zijn van de website, hoe langer het zal duren om als gebruiker de website te kunnen openen. Deze indicator moet zeker in de gaten gehouden worden. Zoals eerder vermeld in het vorige hoofdstuk is de aanvaarde load time maar maximum 3 seconden.

First contentful paint

Een tweede belangrijke indicator is de First contentful paint (FPC). Wat FPC concreet inhoud is de tijd die de browsers nodig heeft om de allereerste stukken inhoud in te laden. Dit zijn dan bijvoorbeeld stukken tekst of foto's.

Speed index

Een volgende indicator is de speed index. De speed index toont aan hoe snel de inhoud visueel beschikbaar is tijdens het laden van de webpagina.

3.2.6 Performance testing tools

Om performance testen uit te voeren op de beide applicaties worden er verschillende tools gebruikt.

Een eerste tool die gebruikt wordt tijdens dit onderzoek is WebPageTest. WebPageTest is een online website performance testing tool dat het mogelijk maakt prestaties te analyseren en te meten. Deze tool geeft ook de mogelijkheid om de testen uit te voeren op verschillende locaties, met verschillende internetsnelheden en verschillende browsers.[20]

Met deze tool is er voornamelijk data verzameld omtrent de verschillende pagina's van de MPA. Ook de SPA is getest met deze tool. Het was alleen niet mogelijk om de volledige SPA te testen met deze tool. Dit komt omdat de tool enkel in staat is om een website op te vragen via de URL. Hierdoor was het niet mogelijk om verdere requests binnen de SPA te testen.

Een tweede tool die gebruikt wordt tijdens dit onderzoek zijn de developer tools van Mozilla Firefox. Binnen de developer tools van Firefox is er de mogelijkheid om performance testen uit te voeren op de website waar een gebruiker zich op bevindt. Deze tool is voornamelijk gebruikt om de SPA te testen. Ook het aantal requests de MPA of SPA uitvoerde tijdens een volledige flow door de applicatie werd opgehaald via de developer tools.[21]

De flow die getest werd tijdens de performance testen ging als volgt:

1. Op de indexpagina wordt een menu opgevraagd voor alle maaltijdperiodes met maar één maaltijd per periode.
2. In het gegenereerde menu wordt er één recept geopend en daarna teruggekeerd naar het menu.
3. Het aangemaakte menu wordt opgeslagen en er wordt ingelogd om dit te voltooien.
4. Het opgeslagen menu wordt geopend en er wordt genavigeerd naar de ingrediëntenlijstpagina.

5. De ingrediëntenlijst wordt gedownload.

3.3 Resultaten

3.3.1 Hoe verschilt de gebruikerservaring tussen een SPA en MPA?

Uit de resultaten van de usertesten is te zien dat qua gebruiksgemak en flow van de website beide structuren bijna geen verschil aantonen. Ook op de mobiele versie werd er geen verschil opgemerkt in gebruiksgemak en flow.

Op het einde van elke usertest kreeg de tester de vraag welke van de twee websites hun voorkeur kregen. In bijna alle gevallen ging de voorkeur uit naar de SPA en de reden waarom is omdat deze volgens de testers sneller werkte dan de MPA.

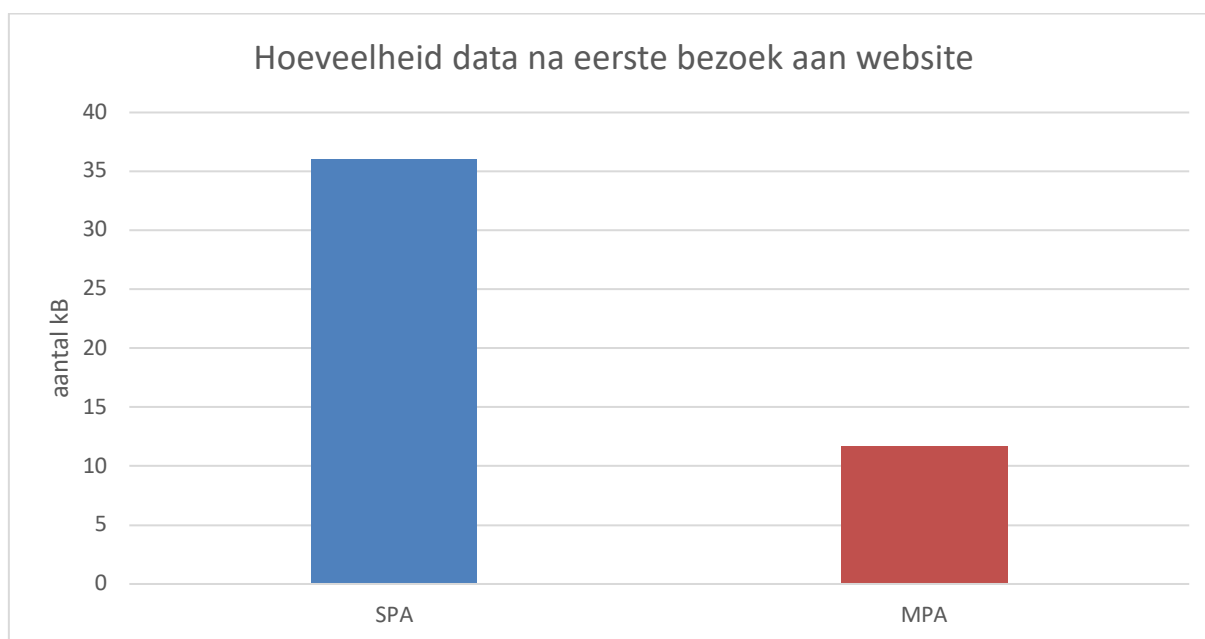
Dit verschil was duidelijk te zien wanneer een tester de usertest uitvoerde met een browser die gelimiteerde internetsnelheid had. De tester was duidelijk gefrustreerd tijdens het gebruik van de MPA wanneer een pagina er even over deed om in te laden. Bij de SPA werd al snel verduidelijkt dat deze veel beter aanvoelde om te gebruiken sinds alles veel efficiënter verliep.

Wat ook opviel tijdens de usertesten is dat de gebruikers veel vlugger kiezen om de ingebouwde browser navigatie te gebruiken. Voor de MPA is deze functionaliteit al volledig ingebouwd, maar voor de SPA moet dit zelf ontwikkeld worden door de developer.

3.3.2 Hoe verschilt de SPA en MPA qua performance?

Load time + grootte van de website

Een eerste indicator die eerder in dit onderzoek aangehaald werd is de grootte van de website. Hoe groter een website is, hoe langer deze normaal gezien duurt om in te laden. Wanneer de eerste pagina van beide applicaties worden ingeladen komen volgende resultaten uit de testen.



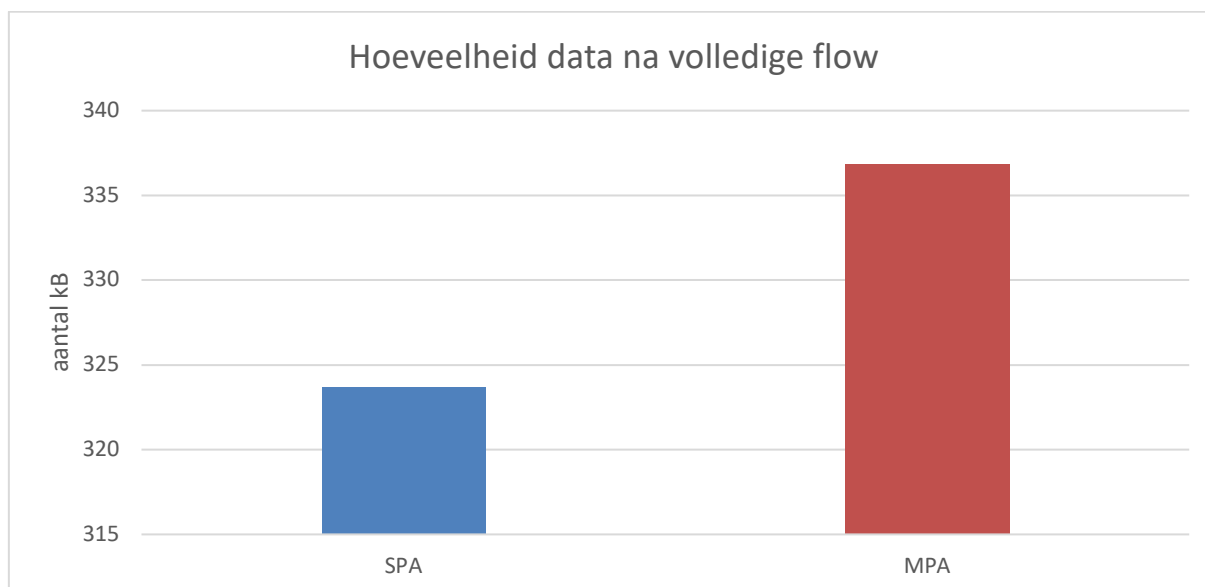
Figuur 16 grafiek vergelijking initieel ingeladen data

Single vs. Multi page applications

Experiment

Wanneer er gesurft wordt naar de beginpagina van de SPA krijgt de browser 36 kB aan files door. Dit is een groot verschil met de beginpagina van de MPA die maar 12 kB aan files bevat. Dit zorgt ervoor dat de laadtijd van de SPA in eerste instantie hoger is dan de MPA.

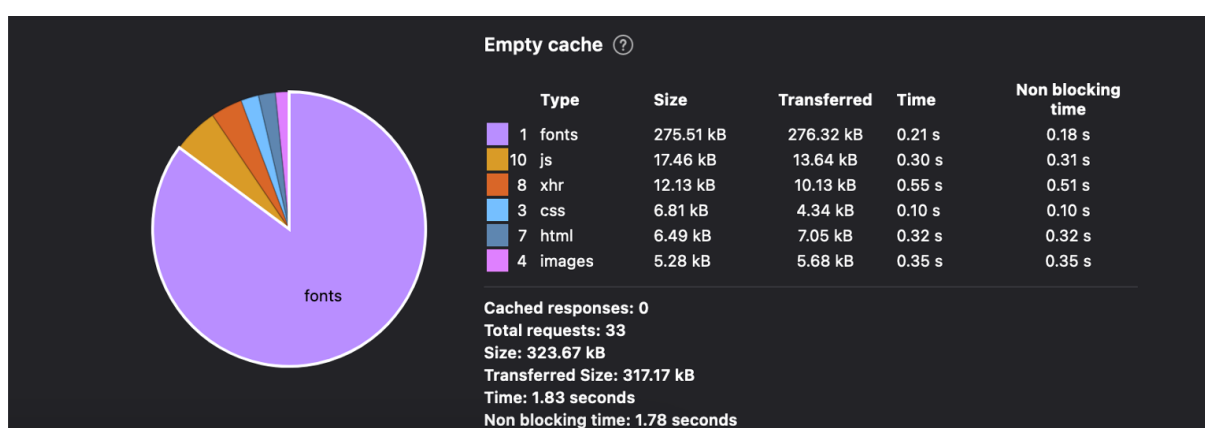
Als volgt moet er ook bij stil gestaan worden hoeveel de totale grootte van de website bedraagt. Met de totale grootte wordt hier bedoeld hoeveel data de browser heeft binnengekregen na één volledige flow door de website.



Figuur 17 grafiek vergelijking hoeveelheid data ingeladen na volledige flow door de applicatie

Bij het vergelijken van de volledige flow door de applicatie is dan uiteindelijk wel te zien dat de MPA meer data verbruikt heeft dan de SPA. Het verschil bedraagt ongeveer 13 kB. Dit is op zich niet echt veel, maar deze applicatie is natuurlijk aan de kleinere kant.

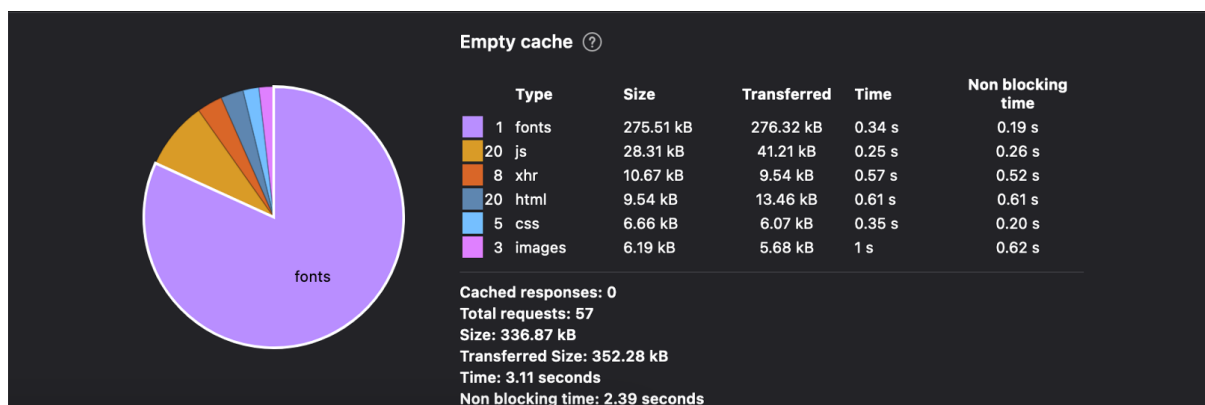
Een andere indicator is ook hoeveel requests de browser heeft moeten maken doorheen de flow van de applicaties.



Figuur 18 aantal requests verkregen SPA

Op bovenstaande afbeelding is te zien hoeveel verschillende requests de browser heeft gemaakt om door de volledige flow van de SPA te gaan. In dit geval bij de SPA zijn dat er dus 33 in totaal.

Single vs. Multi page applications Experiment

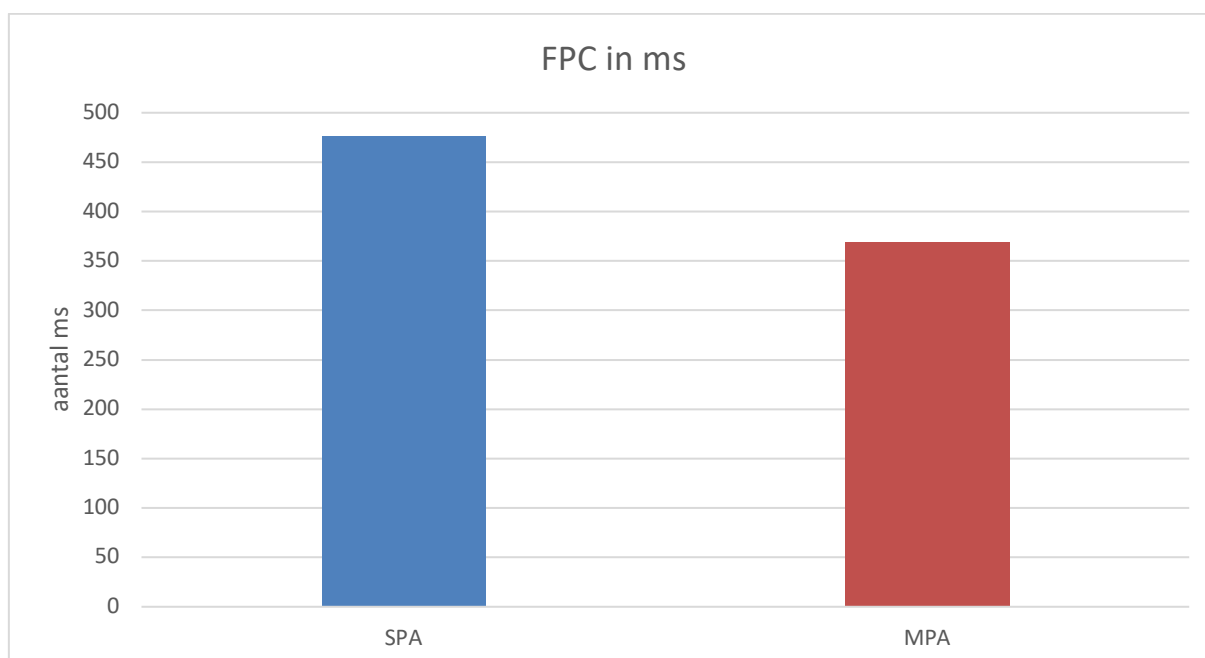


Figuur 19 aantal requests verkregen MPA

Bij de MPA kan er gezien worden dat er veel meer requests nodig zijn dan bij de SPA. Maar liefst 24 extra requests zijn nodig om hetzelfde resultaat te verkrijgen als de SPA.

First contentful paint

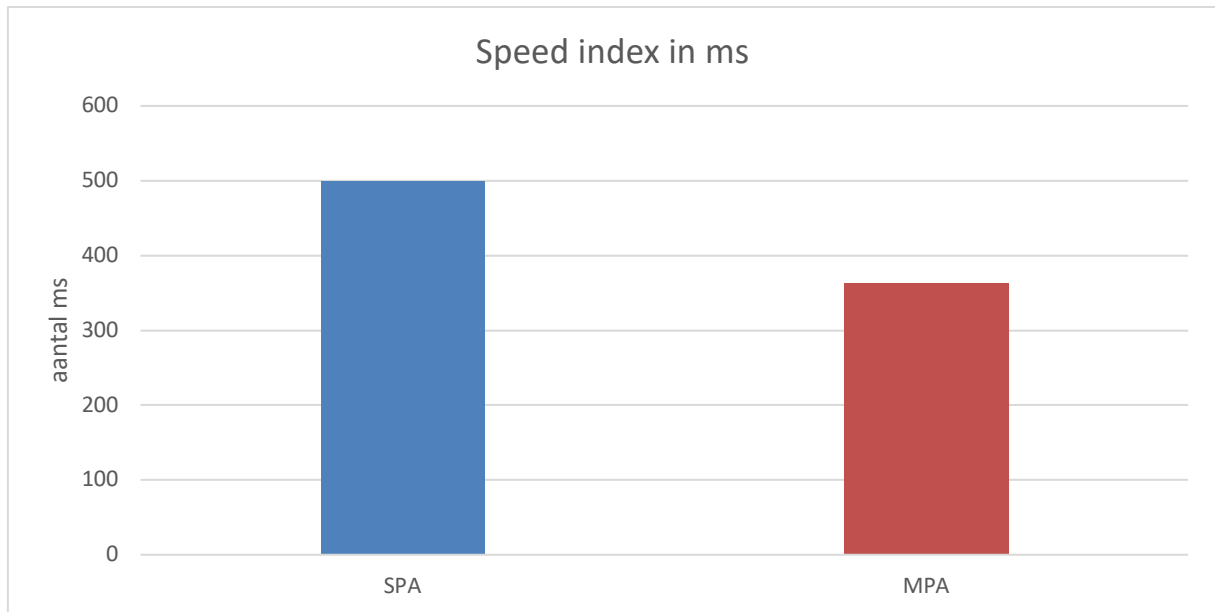
De volgende parameter die vergeleken wordt is de First contentful paint. Hieronder op de grafiek is te zien dat de MPA op dit vlak beter scoort dan de SPA. Dit met een verschil rond de 107 ms. Dit is voornamelijk te linken aan dat bij de SPA alle bestanden worden ingeladen in het begin. De MPA hoeft namelijk enkel maar de bestanden van de beginpagina op te halen.



Figuur 20 grafiek vergelijking FPC

Single vs. Multi page applications Experiment

Speed index



Figuur 21 grafiek vergelijking speed index

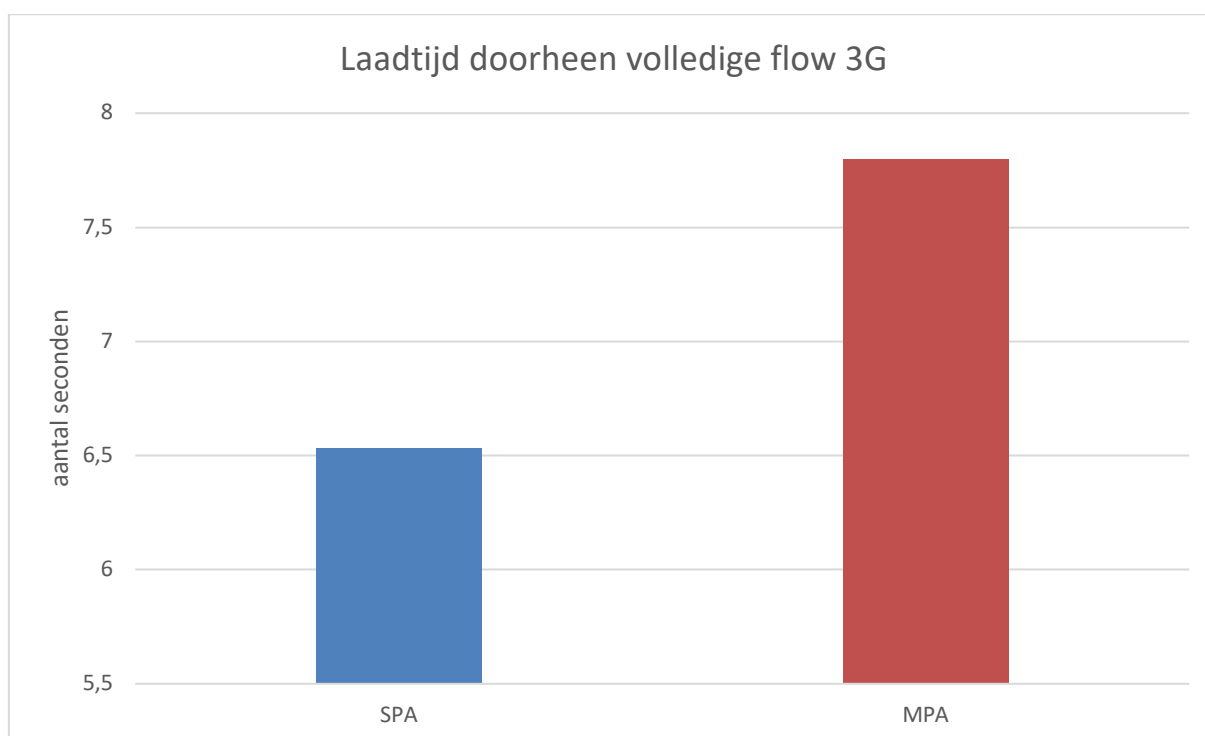
Bij de speed index is de MPA terug net dat tikkeltje sneller. Hier zit er een verschil van 136 ms. Dit is opnieuw te wijten aan het feit dat de SPA meer bestanden moet inladen tijdens het eerste bezoek aan de website.

Laadtijden bij lagere snelheden

Tegenwoordig worden websites ook veel vaker bezocht via een 3G/4G-netwerk. Dit kunnen bijvoorbeeld gebruikers zijn die onderweg zijn met het openbaar vervoer. Daarom is het belangrijk om eens dieper in te gaan hoe de MPA en SPA presteren op een lagere netwerksnelheid.

Single vs. Multi page applications

Experiment

*Figuur 22 grafiek vergelijking laadtijd op 3G-netwerk*

Op bovenstaande grafiek is te zien hoeveel tijd de user moet wachten tijdens het gebruik van de applicaties op 3G. Wat opvalt is dat de MPA in tegenstelling tot de andere testen hier niet sneller scoort dan de SPA. Wanneer er op een 3G netwerk doorheen de flow van de applicatie gegaan wordt presteert de MPA 1,30 seconden trager dan de SPA.

3.3.3 Is er een groot verschil in development ervaring?

Een belangrijk aspect bij het kiezen van een bepaalde structuur van een applicatie is het afwegen of een bepaalde structuur een goede development ervaring heeft. Dit is een zeer persoonlijk iets, maar zeker vermeldingswaardig binnen dit onderzoek.

Als er gekeken wordt naar een MPA die meerdere HTML-pagina's heeft moet op elke pagina een link gelegd worden naar bijvoorbeeld JavaScript of CSS-bestanden. Dit is een zeer repetitieve taak en één die niet echt aangenaam is. Bij een SPA aan de andere kant is er maar een enkele HTML-pagina. Hier wordt er een verwijzing gemaakt naar de nodige JavaScript of CSS-bestanden en daarna hoeft dit nooit meer opnieuw te gebeuren.

Een voordeel van meerdere HTML-pagina's te hebben is dat er veel geïsoleerder kan gewerkt worden. Er hoeft geen rekening gehouden worden met eventuele CSS die niet op deze pagina nodig zou zijn of met logica die niet moet uitgevoerd worden op een bepaalde pagina. Wanneer er maar een enkele HTML-pagina is moet er goed uitgekeken worden naar welke logica er eerst moet uitgevoerd worden. Daarna moet er ook rekening gehouden worden met welke CSS op bepaalde elementen moet zichtbaar zijn op een bepaald moment.

Een ander voordeel bij het werken van een MPA is dat de browser al volledig ingebouwd heeft hoe er gereageerd moet worden op het navigeren tussen verschillende pagina's. Dit kan door middel van de ingebouwde back- en forward knoppen in de browser. Binnen een SPA werken

deze knoppen niet zonder dit zelf volledig te programmeren. Dit kan door middel van de History API die ingebouwd zit in elke moderne browser. Het enige probleem met deze API is dat deze zeer lastig is om te debuggen en waardoor het een hele klus is om deze 100% foutloos te laten werken.

Bij een SPA is het ook zeer vervelend dat wanneer een pagina refreshed alle logica gereset wordt. Dit is lastig tijdens de development fase wanneer er aan een feature gewerkt wordt die pas na vele stappen binnen de app van toepassing komt. Dit zorgt voor een onnodige vertraging binnen het developmentproces.

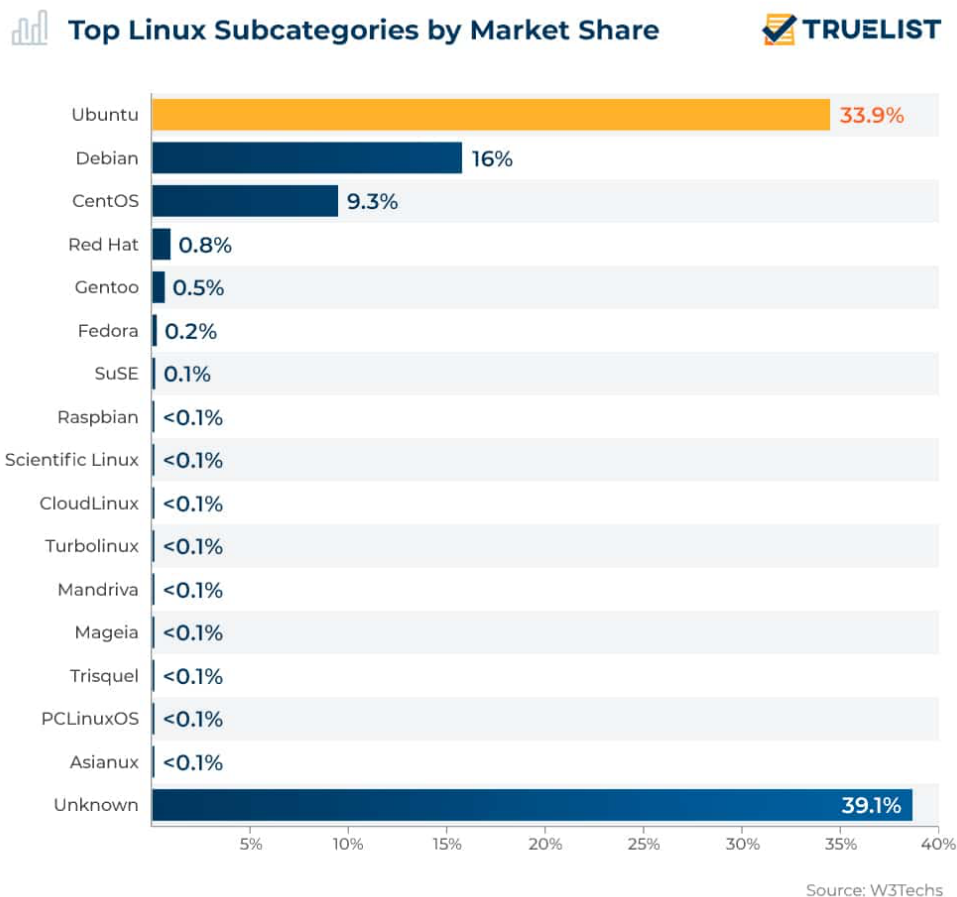
Een ander minpunt binnen het maken van een SPA is dat er rekening gehouden moet worden met welke elementen op de pagina te zien zijn. Binnen een MPA is dit gemakkelijk omdat als er genavigeerd wordt naar een andere pagina alle content van de vorige verdwijnt. Bij een SPA moet de developer zelf code schrijven om ervoor te zorgen dat content die niet zichtbaar moet zijn verwijderd of onzichtbaar wordt.

Vervolgens is het onderhouden van de code ook een belangrijk aspect. De MPA heeft als voordeel dat wanneer er zich een bug voordoet op een bepaalde pagina deze meer geïsoleerd zit. Dit heeft dan als voordeel dat deze rapper te debuggen valt. Bij de SPA zit alle HTML in één file. Dit heeft dan een impact op de uitbreidbaarheid van de applicatie. Zoals bijvoorbeeld wanneer de applicatie groter en complexer wordt het zeer overweldigend kan worden om nieuwe zaken te implementeren.

3.3.4 Hoe verloopt het deployen van een SPA en een MPA?

Beide applicaties zijn gedeployed op een Virtual Private Server (VPS). Op deze VPS draait het besturingsysteem Debian 11.[22] Debian 11 is een Linux distributie die zeer stabiel is en wordt veel gebruikt om websites mee te hosten. Volgens een artikel draaien 16% van alle Linux gebaseerde websites op een Debian besturingsysteem.[23]

Single vs. Multi page applications Experiment



Figuur 23 vergelijking marktaandeel van verschillende Linuxdistributies

Om ervoor te kunnen zorgen dat de website beschikbaar is op het internet is er ook nood aan webserver software. Op de server in dit onderzoek wordt gebruik gemaakt van NGINX om de websites toegankelijk te maken via het internet.

Wanneer NGINX op de server geïnstalleerd staat kan de code van de website op de server geplaatst worden. Hierna hoeft enkel nog de configuratie van NGINX vervolledigt te worden zodat de webserver weet welke bestanden moeten beschikbaar gesteld worden.

Voor zowel de MPA als SPA is deze configuratie volledig hetzelfde. Er moet gezorgd worden dat NGINX weet op welk bestand de website gestart moet worden en verder is er geen extra configuratieverschil.

4 Conclusie

Dit onderzoek geeft antwoord op de hoofdvraag “Single vs. Multi page applications: wanneer kies je wat?”. Om hierop een conclusie uit te schrijven wordt er teruggekeken naar de resultaten van het onderzoek.

Een allereerst element dat onderzocht is tijdens dit onderzoek was de gebruikerservaring. Uit de zeven usertesten die zijn uitgevoerd kan geconcludeerd worden dat de SPA hier beter scoort. De gehele ervaring rond de SPA voelde voor de meeste gebruikers veel sneller aan.

Een tweede element was de performance. Op veel vlakken scoort de MPA beter omdat er meerdere pagina's zijn die afzonderlijk ingeladen kunnen worden. Dit zorgt ervoor dat als een persoon enkel maar bijvoorbeeld zijn of haar menu's wilt raadplegen dit sneller zal aanvoelen. Wanneer er dan gekeken wordt doorheen de volledige flow van de applicatie scoort de SPA dan wel boven de MPA. Vooral bij een 3G/4G-netwerk is hier een groot verschil in het voordeel van de SPA. Ook op vlak van totale requests scoort de SPA beter.

Een derde element dat in dit onderzoek onderzocht werd was de development ervaring. De resultaten die in dit onderzoek naar boven komen is dat de MPA hier toch een betere development ervaring biedt. Bij een SPA zijn er namelijk veel meer zaken waar rekening mee gehouden moet worden en naar uitbreidbaarheid toe is de SPA ook minder geschikt. Daarom wordt de MPA boven de SPA gezet op het vlak van development ervaring.

Een laatste element om rekening mee te houden is SEO. Uit het literatuuronderzoek is gebleken dat een MPA veel beter scoort op vlak van SEO. Dit omwille van het feit dat de inhoud van een MPA makkelijker te crawlen is.

Het beantwoorden van de hoofdvraag blijft een lastige taak. Beide technieken hebben zo hun voor- en nadelen. In het kader van dit onderzoek komt de MPA toch boven de SPA als aanbevolen structuur. Dit is natuurlijk gebaseerd op dit onderzoek waar beide applicaties gemaakt zijn zonder externe frameworks of library's.

5 Discussie

De resultaten komen niet volledig overeen met de verwachtingen die in het begin van het onderzoek zijn opgesteld. Origineel was de verwachting dat de SPA veel beter ging scoren dan de MPA. Dit omdat SPA's meer en meer gebruikt worden door organisaties.

De data die verzameld is uit dit onderzoek is zeker valide. Voor de usertesten is er gezorgd voor een divers publiek en een goed aantal testers. Ook het verzamelen van de performance data verliep altijd op eenzelfde manier bij beide applicaties. Dit zorgt ervoor dat alle data met elkaar vergeleken kon worden.

Een eventueel discussiepunt kan het hoofdstuk over development ervaring zijn. Dit is iets wat zeer subjectief is, maar binnen dit onderzoek toch wel een punt dat moest aangehaald worden. Daarom heb ik altijd geprobeerd om alles zo objectief mogelijk te bekijken als het op de development ervaring aankwam.

Concrete suggesties om dit onderzoek voort te zetten is om verschillende frameworks of library's te introduceren. Dit kan namelijk een heel grote impact hebben op de development ervaring en op de performance van een project.

Referentielijst

- [1] 'Single Page Application (SPA) vs Multi Page Application (MPA) – Two Development Approaches | ASPER BROTHERS'. <https://asperbrothers.com/blog/spa-vs-mpa/> (geraadpleegd 4 april 2023).
- [2] 'Single Page vs Multi-Page Application – What are The Main Differences? - Lightrains'. <https://lightrains.com/blogs/single-page-application-vs-multi-page-application/> (geraadpleegd 4 april 2023).
- [3] 'Single-page application vs Multi-page application: What Is Better for Your Project? | LinkedIn'. https://www.linkedin.com/pulse/single-page-application-vs-multi-page-what-better-your-/?trk=organization_guest_main-feed-card_feed-article-content (geraadpleegd 4 april 2023).
- [4] 'What is a Single Page Application? Pros and Cons of the SPA Technology - HUSPI'. <https://huspi.com/blog-open/definitive-guide-to-spa-why-do-we-need-single-page-applications/#1-client-side-rendering-csr> (geraadpleegd 4 april 2023).
- [5] '(95) What is a single page application? SPA - YouTube'. <https://www.youtube.com/watch?v=elxDHgzGCnY> (geraadpleegd 4 april 2023).
- [6] 'Pros and Cons of Single-Page Applications'. <https://www.spiceworks.com/tech/devops/articles/what-is-single-page-application/> (geraadpleegd 4 april 2023).
- [7] 'SPA (Single-page application) - MDN Web Docs Glossary: Definitions of Web-related terms | MDN'. <https://developer.mozilla.org/en-US/docs/Glossary/SPA> (geraadpleegd 4 april 2023).
- [8] K. Lawson, 'What Is a Single Page Application? | Bloomreach', 2022. <https://www.bloomreach.com/en/blog/2018/what-is-a-single-page-application> (geraadpleegd 4 april 2023).
- [9] 'MDN Web Docs'. <https://developer.mozilla.org/en-US/> (geraadpleegd 4 april 2023).
- [10] 'Wat is een API en wat kan je ermee? - Salesforce NL Blog'. <https://www.salesforce.com/nl/blog/2019/10/wat-is-een-api.html> (geraadpleegd 28 mei 2023).
- [11] '<template>: The Content Template element - HTML: HyperText Markup Language | MDN'. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/template> (geraadpleegd 28 mei 2023).
- [12] 'History API - Web APIs | MDN'. https://developer.mozilla.org/en-US/docs/Web/API/History_API (geraadpleegd 29 mei 2023).
- [13] 'History: pushState() method - Web APIs | MDN'. <https://developer.mozilla.org/en-US/docs/Web/API/History/pushState> (geraadpleegd 29 mei 2023).
- [14] 'Fetch API - Web APIs | MDN'. https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (geraadpleegd 29 mei 2023).

Single vs. Multi page applications
Referentielijst

- [15] 'Why UX testing is so important for your product in 2023 - UX Design Institute'. <https://www.uxdesigninstitute.com/blog/why-ux-testing-is-so-important/> (geraadpleegd 29 mei 2023).
- [16] '8 Usability Testing Methods That Work (Types + Examples)'. <https://www.hotjar.com/usability-testing/methods/> (geraadpleegd 29 mei 2023).
- [17] 'Everything You Need to Know About Website Performance'. <https://www.hostinger.com/tutorials/website-performance> (geraadpleegd 28 mei 2023).
- [18] 'Why Site Speed Is So Important: Conversions, Loyalty, and Google Search Ranking (2023)'. <https://www.shopify.com/blog/site-speed-importance> (geraadpleegd 14 mei 2023).
- [19] 'Top 10 Key Website Performance Metrics You Should Measure - Sematext'. <https://sematext.com/blog/website-performance-metrics/> (geraadpleegd 1 juni 2023).
- [20] 'WebPageTest - Website Performance and Optimization Test'. <https://www.webpagetest.org/> (geraadpleegd 7 mei 2023).
- [21] 'Performance Analysis — Firefox Source Docs documentation'. https://firefox-source-docs.mozilla.org/devtools-user/network_monitor/performance_analysis/ (geraadpleegd 1 juni 2023).
- [22] 'Debian -- Reasons to use Debian'. https://www.debian.org/intro/why_debian (geraadpleegd 14 mei 2023).
- [23] 'Linux Statistics 2023 - TrueList'. <https://truelist.co/blog/linux-statistics/> (geraadpleegd 14 mei 2023).

Lijst van bijlagen

Bijlage 1: GitHub repository	44
Bijlage 2: Usertest magazijnier MPA SLOW	44
Bijlage 3: Usertest magazijnier SPA SLOW	46
Bijlage 4: Usertest verpakkingsmedewerker SPA iPad	47
Bijlage 5: Usertest verpakkingsmedewerker MPA iPad	48
Bijlage 6: Usertest student informatica SPA	50
Bijlage 7: Usertest student informatica MPA	51
Bijlage 8: Usertest student lager onderwijs SPA	53
Bijlage 9: Usertest student lager onderwijs MPA	54
Bijlage 10: Usertest metser SPA	56
Bijlage 11: Usertest metser MPA	57
Bijlage 12: Usertest student humane wetenschappen SPA	59
Bijlage 13: Usertest student humane wetenschappen MPA	60
Bijlage 14: Usertest strijkhulp SPA	62
Bijlage 15: Usertest strijkhulp MPA	63

Bijlagen

Bijlage 1: GitHub repository

URL naar de GitHub repository met alle data en bestanden uit dit onderzoek:

<https://github.com/wittevrongelthibaut/BAP>

Bijlage 2: Ustest magazijnier MPA SLOW

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	40-50
Beroep	Magazijnier

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester gaat eerst naar de login pagina.

De tester ging eerst inloggen zonder account, maar hij heeft ontdekt dat hij moet registreren.

Registreren op slow mode toont niet aan of hij effectief iets aan het doen is bij het registreren.

Het menu aanmaken gaat vlot.

De tester gaat direct naar save menu en slaat het menu op.

Bij het saven van het menu is het niet duidelijk of de website effectief het aan het saven.

De tester vindt snel zijn menu terug en opent de ingrediëntenlijst.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester gaat eerst naar Menu's en logt niet direct in als andere gebruiker.

De tester vindt na even uit dat hij moet switchen van account.

De tester logt gemakkelijk in met de nieuwe credentials die opgegeven staan.

De website zou ook meer mogen tonen of hij bezig is met inloggen.

De tester maakt rap zijn menu terug aan.

De tester vindt het makkelijk om te recepten te refreshen.

De tester zit vast op de overlay van de pagina omdat de confirmation box niet op het scherm komt.

De tester vindt het uiteindelijk en savede het menu

De tester vindt makkelijk zijn menu terug en download zeer snel de ingrediëntenlijst.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester keert terug naar de main page met de plan to plate knop.

Hij maakt rap een menu aan volgens de criteria die er staat.

De tester klikt op het openen van een recept.

De tester gebruikt de back arrow van browser om terug te keren.

De tester refreshed tot hij de juiste menu structuur heeft.

Qua UX zou het refreshen van een recept beter mogen om het soms onduidelijk is of het recept aan het refreshen is.

De tester savet het menu rap.

En vind snel zijn menu terug en download zijn ingrediënten.

Bijlage 3: Usertest magazijnier SPA SLOW

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	40-50
Beroep	Magazijnier

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester logt terug direct in met zijn account.

Na dat de tester is ingelogd gaat hij direct naar de main pagina.

De tester maakt terug rap een menu aan en savede de menu.

De tester opent de saved menus pagina en opent zijn net aangemaakte menu.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

Eerst logt de tester uit.

Inloggen met verkeerde credentials toont geen error.

De tester maakt rap terug een menu aan zonder probleem.

De tester zat even vast op de overlay maar herinnerd dat hij omhoog moest scrollen.

De tester savede zijn menu en klikt op saved menu en opent zijn ingrediëntenlijst zonder problemen.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester maakt terug zeer snel een menu aan.

De tester refreshed alles tot hij de juiste structuur heeft.

De tester savede het menu correct en opent zijn gesavede menu via de saved Menus screen.

De tester vindt hij terug zijn menu en opent zijn ingrediëntenlijst.

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

De SPA is het beste omdat deze het snelste werkt. Qua gebruik van de website wordt er weinig verschil gemerkt.

Bijlage 4: Usertest verpakkingsmedewerker SPA iPad

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	40-50
Beroep	Verpakkingsmedewerker

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester logt in op de website.

Ze klikt op Plan to Plate en begint met een nieuwe menu aan te maken.

Zeer vlot savede ze dit menu.

Ze opent de saved menu page en opent daarna terug haar ingrediëntenlijst.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester logt direct uit. En logt in met het juiste account

Ze maakt zeer snel een nieuwe menu.

Ze refreshed receptjes zeer goed en leest de verschillende receptjes door en savede het menu perfect.

Ze opent de saved menu page en opent daarna perfect de ingrediëntenlijst.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester maakt rap het menu aan.

Ze refreshed zeer enthousiast alle receptjes tot de juiste structuur bereikt is.

Ze savede het receptje met de juiste naam.

Daarna vindt ze direct de ingrediëntenlijst

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

De tester vindt geen verschil in beide.

Bijlage 5: Usertest verpakkingsmedewerker MPA iPad

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	40-50
Beroep	Verpakkingsmedewerker

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester vindt redelijk rap hoe ze een menu moet genereren.

De tester klikt op save menu en vraagt wat ze moet invullen, ik geef aan dat ze zelf kan kiezen.

Daarna komt ze op de login pagina maar ze weet niet goed wat ze moet doen. Na even helemaal opnieuw te beginnen heeft ze gezien dat er een register here knop is.

Daarna maakte ze een nieuw account opent ze een menu een vindt ze haar ingrediëntenlijst.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester zoekt hoe ze terug kan inloggen. Ze vindt na even zoeken dat er een logout knop zit op de profile page.

Nu logt ze opnieuw in en maakt ze een menu met de criteria.

De tester maakt veel gebruik van de back button.

Ze vindt hoe ze een recept opent en hoe ze deze kan aanpassen.

De tester had rap door dat ze naar boven moest scrollen en maakte het recept aan.

Ze vindt rap terug het ingrediëntenlijstje en ze downloadt deze.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

Ze gaat naar de main pagina met de plan top late knop.

Ze maakt een receptje volgens de criteria en refreshed de receptjes tot ze de juiste structuur heeft.

En ze slaat dit receptje juist op.

Ze vindt dit receptje direct terug en opent de ingrediëntenlijst direct.

Bijlage 6: Ustertest student informatica SPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	17-20
Beroep	Student informatica

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester maakt een menu met de criteria, zonder in te loggen.

De tester klikt op save menu en komt op de inlogpagina. Hier maakt de tester een account.

De tester klikt op saved menus en opent zijn menu en opent daarna zijn ingrediëntenlijst zonder problemen.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester logt uit zonder lang te zoeken.

De tester logt daarna in met de gegeven credentials.

De tester klikt op plan to plate om naar de index pagina te gaan.

Hij maakt nu een recept met de criteria en dit gaat vlot.

De tester lacht om de overlay zonder confirmation box maar maakt hierna zonder problemen de menu aan.

Nu klikt hij op saved menus en opent hij zijn menu en download hij alles direct.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester raakt zonder problemen terug op het beginscherm en maakt een nieuw recept aan.

De tester refreshed de recepten zonder problemen tot hij de juist structuur gevonden heeft.

Hij klikt op SavedMenus en opent zijn menu en download dit terug.

Dit alles ging vrij vlot.

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

De tester vond dat de SPA over het algemeen het snelst was qua laadtijden.

Qua werken met de site zit er geen verschil tussen de twee vindt de tester.

Bijlage 7: Usertest student informatica MPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	17-20
Beroep	Student informatica

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester maakt een menu zeer vlot aan zonder veel na te denken

De tester klikt na even kijken op Save menu maar klikt daarna op cancel om een receptje te bekijken.

De tester klikt hierna terug op back to menu en slaat het recept op en komt op de login pagina. Hier loopt hij even vast. Hij klikt op de back button en bevindt zich terug op de menu pagina.

Nu klikt hij terug op save menu en ondervindt dat hij nog een account moet maken. Dit duurt wel even voor hij het doorheeft. Hij registreert een account.

De tester opent zijn opgeslagen menu.

En nu heeft hij de ingrediëntenlijst gevonden.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester logt uit

Hierna logt de tester in met de gegeven credentials

De tester maakt terug vlot een recept aan.

De tester keert terug met de back to menu knop en refreshed een receptje.

De tester klikt op save menu en loopt even vast omdat hij terug vastloopt op het eerst de overlay en scrolled naar de confirmation box.

De tester vindt rap zijn menu terug en opent de lijst en downloadt deze.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester maakt een menu

Hij vraagt zich af of je een recept kon zoeken, hierop antwoord ik dat je enkel maar kan refres-
hen.

De tester maakt rap de structuur na van het recept en savede het menu correct.

Daarna klikt de tester op zijn recept en download hij zijn ingrediëntenlijst direct.

Bijlage 8: Usertest student lager onderwijs SPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	17-20
Beroep	Student lager onderwijs

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester maakt direct een menu aan zonder problemen.

De tester saved de menu direct met een naam. De tester maakt een nieuw account aan.

En opent hierna de saved menus. Hierna klikt ze op de aangemaakte menu en vind ze direct haar ingrediëntenlijst.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester vindt direct hoe ze terug kan keren naar de main pagina met de Plan to plate knop.
Hier maakt ze direct het juiste menu met de opgegeven criteria.

Ze opent een recept direct.

Ze refreshed een paar receptjes

Bij het saven raakt de tester geïrriteerd door de confirmation box die te ver omhoog staat.

Ze klikt direct na het aanmaken op de saved menus knop en ziet haar menu direct staan.

Ze had mis geklikt op een pijltje maar met de back to menu knop kwam ze terug op de pagina waar ze moest zijn.

Ze downloadt nu direct de lijst.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester maakt een menu aan zonder problemen.

Ze refreshed alles in het recept tot ze de structuur uitkomt die opgegeven staat.

Ze savede de menu zonder problemen met de juiste naam en klikt daarna op saved menus om haar menu te bekijken en download zonder problemen de lijst.

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

Volgens de tester ging de tweede website sneller.

Qua gebruiksgemak waren beide websites hetzelfde.

Bijlage 9: Usertest student lager onderwijs MPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	17-20
Beroep	Student lager onderwijs

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester klikt breakfast en lunch aan en kiest voor 2 receptjes. Dit gaat vlot

De tester klikt op een recept en leest het, daarna klikt ze op save menu en ze geeft een naam in.

Ze klikt op save en komt op de login pagina. Hier moet ze even denken en klikt op register. Hierin maakt ze dan een eigen account aan.

Na het maken van een account heeft ze direct haar menu gevonden en kwam ze direct uit op het ingrediëntenlijstje zonder haperingen.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester vraagt zich af of ze moet uitloggen.

De tester voert de inlog credentials in.

De tester maakt direct een menu aan voor 3 dagen en alle periodes.

De tester keert terug met de back to menu knop op de website en niet met de back button

Bij het saven van de menu vroeg de tester of de site vast liep omdat ze de overlay te zien krijg en niet de confirmation box. Na even ondervindt ze dat ze naar boven kan scrollen en vermeld dat ze dat niet fijn vond dat die box niet direct tevoorschijn kwam bij het klikken op save menu.

Ze heeft direct haar menu gevonden en de ingrediëntenlijst en ze heeft direct haar lijst gedownload.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

Ze maakt een menu aan zonder problemen.

Ze vindt zonder problemen hoe ze de receptjes kan refreshen

Hierna saved ze het menu heel vlug en vind ze terug de ingrediëntenlijst en download ze deze zonder probleem.

Bijlage 10: Usertest metser SPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	60-99
Beroep	Metser

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

Het aanmaken van het recept het recept ging vlot, daarna ging het save van het menu ook zeer vlot.

Daarna kwam het inlog scherm terug, hierin heb ik kleine tip moeten geven om te registreren.

Daarna deed hij perfect het saved menus screen open en opende hij zijn menu en toen kreeg hij ook zijn ingrediëntenlijst te zien.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

Het uitloggen ging iets trager, hij maakte gebruik van de back arrows van chrome om terug te keren naar het profile screen om uit te loggen.

Het inloggen ging wel vlot en het aanmaken van het menu ging ook vlot.

De overlay is terug een hindernis omdat deze niet juist gepositioneerd staat.

Het openen van de saved menus ging goed en daarna het downloaden was ook okay.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

Het terug keren naar de main menu was heel vlot.

Het aanmaken van het menu ging vlot.

Daarna het refreshen van de recepten ging ook vlot en het saveen ook.

Het downloaden ging dan ook goed alleen zag ik vaak dat hij wou klikken op het record en niet op de knop met het pijltje

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

Beide sites hadden geen verschil kreeg ik als antwoord.

Bijlage 11: Usertest metser MPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	60-99
Beroep	Metser

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester vond na even zoeken hoe hij een menu kon genereren.

Daarna vond hij redelijk rap hoe je een recept moest opslaan.

Toen hij bij het aanmaken een account moest maken was het even denken wat er moest gebeuren, daarna heb ik een paar kleine tips gegeven zodat hij op weg kon.

Hij savede zijn menu en vond redelijk rap dat hij kon klikken op zijn menu en daarna dat hij zijn ingrediëntenlijst kon openen.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

Bij het uitloggen was er wat hulp nodig om dit te verwezenlijken.

Daarna is het gelukt om in te loggen via de credentials.

Bij het maken van het gerecht lukte het over het algemeen wel redelijk rap om een menu te maken. Iets trager dan anders.

Het scrollen van de overlay was iets dat ik zeker moest vermelden.

Het was even zoeken naar de naam die moest ingevoerd worden en daarna was het vinden van de download knop zeer snel gevonden.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

Het vergde wat hulp om terug te keren naar de homepagina.

Het invullen van het menu ging zeer vlot en het opslaan ook.

Het openen van het menu ging iets trager.

Het downloaden ging ook, maar vergde iets meer zoek werk.

Bijlage 12: Usertest student humane wetenschappen SPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	17-20
Beroep	Student humane wetenschappen

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester maakt een recept zeer snel terug aan.

Ze savede dit recept direct en logt in met haar account die ze daarnet heeft aangemaakt.

Ze opent het menu en ziet de ingrediëntenlijst direct.

Deze flow ging nu zeer snel omdat ze de MPA al had onderzocht.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester logt terug uit en ging direct terug naar de login pagina.

Nu zat ze op de profile page en wou ze terug keren naar de index pagina. Alleen in de SPA lukt dit niet met de back buttons. Ze is aan het zoeken om terug te keren maar vind de Home knop niet.

Na een kleine tip dat ze overal al geklikt had behalve op 1 plaats vond ze dat ze op 'Plan to Plate' kon klikken om terug te keren naar de index pagina.

Hierna kon de tester direct terug alle menu's aanmaken. Ze past een receptje direct aan en weet van de vorige keer dat ze bij het save terug naar boven moet om het te kunnen save. Ze vindt direct de menu's terug en vindt haar net aangemaakte menu terug. Hierna vindt ze terug de ingrediëntenlijst en download ze die.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester maakt terug een menu aan en ik zie dat ze gebruik maakt van enter om de form door te sturen.

De tester vindt rap hoe ze de receptjes terug aan past en maakt de structuur terug zoals het hoort.

Hierna slaat ze direct het menu op en vind ze direct haar weg naar de menu en download ze de ingrediënten.

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

Ik vond dat beide websites hetzelfde waren en dat er weinig verschil qua snelheid of UX zat.

Bijlage 13: Usertest student humane wetenschappen MPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	17-20
Beroep	Student humane wetenschappen

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester heeft heel rap een menu kunnen generaten zonder ergens te misklikken, wel niet ingelogd. Heeft geklikt op een receptje en vroeg daarna of de user test klaar was omdat ze bij het recept zat. Heb even moeten zeggen dat het nog niet klaar was. Hier na heeft de tester

zelf gevonden dat ze moest terugkeren naar het menu en heeft ze de knop save menu gevonden.

Hierna heeft ze erop geklikt en werd zo doorgestuurd naar de login pagina. Hier zat ze dan even vast omdat ze niet wist met wat ze moest inloggen. Hierna maakte ze zelf een account na dat ik een tip gegeven had dat ze een volledige nieuwe gebruiker was.

Hierna heeft de tester zelf gevonden waar de ingrediënten stonden na het klikken van het aangemaakte menu en hiermee is dit scenario afgerond.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester logt direct uit en logt in met de nieuwe gegevens zonder dat ik een tip moest geven dat ze was ingelogd.

De tester ging direct gaan kijken naar menus maar zag dat er geen menus in dit account zat. De tester zat daarna even vast met het terug keren naar de main pagina. Hiervoor klikte ze op back arrow om terug te keren. Hierna maakte ze direct het menu aan voor de 3 dagen die opgegeven staan in het scenario.

Qua UX zou het goed zijn om een spinner in de receptpagina te steken deze duurde even om in te laden.

De tester vond direct hoe ze een recept kon refreshen

UX: Als er geklikt wordt op save menu dan moet de pagina ook effectief het menu box tonen, nu bleef de site gewoon staan en zag de tester een overlay en was ze aan het wachten. Na tussen te komen en te vertellen dat ze naar boven kon scrollen om de box te zien kon ze terug verder. Ze vond zeer snel terug de ingredients pagina waar ze daarna de ingrediëntenlijst direct gedownload heeft.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

De tester heeft nog altijd niet gevonden hoe ze terug kan keren naar de main menu. Hiervoor gebruikt ze steeds de back buttons om terug te keren naar de main pagina.

De tester maakt direct terug juist een menu aan en begint direct te refreshen om de juiste structuur te krijgen.

Daarna slaat de tester het menu direct juist op. De tester klikt op de article zelf om door te gaan naar een menu en niet op het pijltje, Hierna vindt ze dat ze op het pijltje moet klikken en kon ze direct terug het lijstje downloaden.

Bijlage 14: Usertest strijkhulp SPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	60-99
Beroep	Strijkhulp

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

Het aanmaken van een menu verliep iets trager.

Het was iets moeilijker om een menu aan te maken ze was vooral aan het kijken om het receptje te openen in plaats van op te slaan.

Daarna wanneer ze op het inlog menu was duurde het even voor ze doorhad dat ze eerst moest registreren. Hiervoor had ze een tip nodig om dit te vinden.

Na het registreren kwam ze op de profielpagina en kwam ze al zeer snel op de savedmenu pagina.

Hier was ze dan even aan het zoeken hoe ze een menu kon openen, hier zag ik dat ze wou klikken op het article zelf in plaats van op het pijltje zelf.

Hierna vond ze wel snel de ingredients knop.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

Het uitloggen had ze vlug gevonden via de profile page knop.

Het navigeren naar de main page vond ze na een tip.

Het aanmaken van een menu ging vlot.

Het opslaan van het menu ging iets minder goed omdat de confirmation box terug te hoog stond.

Het opslaan zelf ging dan wel goed.

Daarna het ingrediëntenlijstje zoeken ging ook direct, samen met het downloaden.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

Het terug keren naar de main page ging direct.

Het refreshen van de receptjes gaat ook goed.

Daarna het opslaan ging ook zeer vlot.

Daarna klikte ze op saved menus.

Haar menu vond ze ook snel terug en het downloaden van ingrediëntenlijstje ging ook goed.

Bijlage 15: Usertest strijkhulp MPA

UX-tests BAP Thibaut Wittevrongel

Tester info	
Leeftijdsgroep	60-99
Beroep	Strijkhulp

Scenario's

Scenario 1:

Je wilt een nieuw menu samenstellen om dit weekend klaar te maken. Graag zou je 2 receptjes hebben voor het ontbijt en 2 receptjes voor de lunch.

Na het maken van het menu wil je naar de winkel en zou je een boodschappenlijstje willen raadplegen.

Findings:

De tester maakt direct een account aan.

Daarna ging het aanmaken van een menu zeer vlot.

Het save van een menu lukt ook zonder haperingen.

Het openen van een menu gaat ook super vlot zonder problemen en daarna de lijst openen ging ook vanzelf.

Scenario 2:

Je hebt al een account op de website met volgende gegevens:

Gebruikersnaam: Tester

Wachtwoord: test123

Je zou graag een menu aanmaken voor de komende 3 dagen die voor alle maaltijdperiodes een maaltijd bevat. In de lijst staat er een receptje waarvan je niet goed weet hoe je het zou moeten klaarmaken. Uiteindelijk vind je het receptje maar niks en kies je ervoor om deze te veranderen. Je beslist om dit menu de naam 'HalfWeekMenu' te geven. Graag had je ook een lijstje van de ingrediënten gedownload om mee te nemen naar de winkel.

Findings:

De tester gaat direct naar de inlog pagina.

Hier voert ze juist de credentials in.

Het maken van het recept gaat ook zeer vlot.

Het save ging iets trager omdat de confirmation box niet voor de user tevoorschijn kwam.

Daarna kon ze wel goed save en vond ze makkelijk de contents van het menu en kon ze de ingrediëntenlijst raadplegen.

Scenario 3:

Je wilt graag voor vandaag een menu aanmaken. Je hebt zin om een als ontbijt pannenkoeken te eten, als lunch zou je graag het recept hebben om een 'Creamy Tomato Soup' te maken en als avondmaal wil je graag 'Black Bean Tacos' maken. Dit menu geef je de naam 'DagMenu'. Hiervan zou je ook graag de ingrediëntenlijst downloaden.

Findings:

Het aanmaken van het menu ging vlot.

Het refreshen van de ingrediënten ging ook super vlot.

Het Saven van het menu ging ook vlot.

Daarna het openen van het menu ging vlot en het raadplegen van de ingrediëntenlijst.

Q: Welke website van de twee die je nu getest hebt, krijgt uw voorkeur?

Alle websites waren hetzelfde volgens de tester.

