

RESPONSIVE DESIGN

Adviesrapport



Inhoudsopgave

1.	Inleiding.....	1
2.	Responsive design	3
2.1.	Definitie	3
2.2.	Voor- en nadelen.....	3
3.	Werkwijze.....	5
3.1.	Product backlog	5
3.2.	Sprint backlog.....	5
3.3.	Scrummaster	6
3.4.	Scrumproces	6
4.	Grafische richtlijnen	9
4.1.	Frameworks	9
4.2.	Navigatie	10
4.3.	Kolommen	12
4.3.1.	Wireframe	12
4.3.2.	Mobile first	13
4.3.3.	Content-out design	13
4.4.	Tabellen	14
4.5.	Formulieren	14
4.6.	Touch.....	15
5.	Technische richtlijnen	17
5.1.	Frameworks	17
5.1.1.	Joomla!.....	17
5.1.2.	Wordpress.....	17
5.2.	Relatieve waardes	18
5.3.	Media queries.....	19
5.4.	Viewport meta tag	20
5.5.	Touch.....	21
5.6.	Laadtijd.....	21
5.7.	Browsersupport	21
6.	Testen.....	23

1. Inleiding

Het online landschap verandert. Er komen steeds meer apparaten waarmee mensen online kunnen en willen komen. Elk apparaat heeft zo zijn eigen kenmerken; het heeft zijn eigen schermresolutie, schermgrootte, gebruikersinterface en besturingsmogelijkheden. Al deze mogelijkheden en talloze combinaties zorgen voor een probleem voor de webontwikkelaar. Hij kan niet meer uitgaan van een vast product waarmee een website bekeken wordt. Door deze recente ontwikkeling moest de webwereld komen met een oplossing zodat websites overal op bekeken kunnen worden.

Responsive design is de oplossing waarmee de webwereld is gekomen. Deze trend is de laatste jaren in opkomst. Het geeft de webontwikkelaar de mogelijkheid om één website te maken die zich aanpast aan alle mogelijke schermresoluties. Of de website nu bezocht wordt met een tablet, smartphone, laptop of desktop, de gebruiker krijgt altijd de optimale websiteweergave te zien.

Voor Los Banditos is dit adviesrapport opgesteld om responsive design binnen hun werkwijze te implementeren. Dit rapport zal hiervoor eerst een duidelijke definitie van responsive design geven. Vervolgens zal het de nieuwe werkwijze beschrijven. Voor de ontwerpers zijn er een aantal grafische richtlijnen opgesteld. Eveneens zijn er voor de ontwikkelaars een aantal technische richtlijnen opgesteld.

Voor dit adviesrapport is een *best practices analyse* gedaan. Dit betekent dat praktijkvoorbeelden van responsive websites zijn geanalyseerd, om te kijken hoe er omgegaan wordt met specifieke onderdelen op deze websites. Daarnaast is er gebruik gemaakt van een literatuuronderzoek wat verricht is naar responsive design. Ook zijn er verschillende case studies doorlopen, waaruit verbeterpunten zijn gekomen. Deze verbeterpunten zijn in dit adviesrapport verwerkt. De case studies die doorlopen zijn, zijn het testen van twee responsive frameworks en het responsive maken van de projectmanagementtool van Los Banditos: MijnWerkers.

2. Responsive design

2.1. Definitie

Een responsive design is een website die zich aanpast aan de schermresolutie waarin hij bekeken wordt. Uit het theorieonderzoek is gebleken dat responsive design wordt herkend aan twee kenmerken:

- een vloeiende lay-out die gebruik maakt van een flexibele grid;
- afbeeldingen die in een flexibele context werken.

Om dit te realiseren wordt er gebruik gemaakt van drie technieken: het gebruik van relatieve waardes, het gebruik van media queries en het gebruik van de viewport meta tag. Deze technieken staan in hoofdstuk 5 verder beschreven.

2.2. Voor- en nadelen

Voordelen

- Eén HTML pagina betekent dat er minder onderhoud gepleegd hoeft te worden.
- Je bent altijd dezelfde tags aan het stijlen.
- Je krijgt nooit een horizontale scrollbar (ook niet als mensen de browser gebruiken in een kleiner formaat).
- Verschillende apparaten kunnen dezelfde link gebruiken om op de pagina te komen, in tegenstelling tot het gebruik van aparte subdomeinen voor mobiele websites.
- Je maakt je website 24/7 bruikbaar, doordat hij op elk apparaat toegankelijk is.
- Er wordt meer rekening gehouden met Search Engine Optimization (SEO).

Nadelen

- Het is niet op alle gebruikte browsers toepasbaar. Internet Explorer 8 (6.4% van de gebruikers) en oudere versies van Firefox (7.3% van de gebruikers) ondersteunen responsive design bijvoorbeeld nog niet.
- De laadtijd van de pagina is erg hoog in vergelijking met een mobiele website.
- Het ontwikkelen van een responsive design kost tijd en geld.

3. Werkwijze

De twee kenmerken die in de werkwijze van Los Banditos terugkomen, zijn specialisme en structuur. Bij de scrumwerkwijze staan deze twee kenmerken centraal. Scrum is een Agile softwareontwikkelmethode. Dit betekent dat er geconcentreerd wordt op het ontwikkelen in korte, overzichtelijke perioden, waarbij meerdere mensen tegelijkertijd werken aan één project. Bij scrum gebeurt dit door het project op te delen in 'sprints'. Een sprint is een vaste, herhalende periode (of iteratie) van meestal één tot vier weken, waarvan de lengte en inhoud van tevoren zijn vastgesteld. Gedurende elke sprint levert het scrumteam een functionaliteit van het eindproduct op.

Elk project heeft zijn eigen scrumteam. Dit team bestaat uit de producteigenaar, de scrummaster en het ontwikkelingsteam.

3.1. Product backlog

De producteigenaar (de klant) is verantwoordelijk voor het samenstellen van de *product backlog*. De product backlog is een geordende lijst met eisen voor een product. Deze lijst bestaat uit functionaliteiten, op te lossen systeemfouten en niet-functionele eisen. Al deze eisen worden door de producteigenaar geordend op basis van prioriteit. De prioriteit wordt bepaald door het toekennen van de geschatte meerwaarde en het bepalen van de ontwikkelingstijd. De geschatte meerwaarde wordt bepaald door de producteigenaar. Dit gebeurt doorgaans door het toekennen van een aantal punten. De ontwikkelingstijd wordt door het ontwikkelingsteam aangegeven. Dit gebeurt tevens door het toekennen van punten, maar het is ook mogelijk om dit in uren uit te drukken. De eisen worden in verhaalformaat weergegeven. De structuur die hiervoor gebruikt wordt is: "Als een [gebruikertype] wil ik [een bepaalde actie] zodat ik [het gewenste resultaat]". In deze structuur zijn de variabelen tussen de vierkante haakjes geplaatst. Een voorbeeld voor een eis genoteerd in verhaalformaat is: "Als een medewerker van Los Banditos wil ik een takenlijst hebben in MijnWerkers, zodat ik snel kan zien welke taken er volbracht moeten worden voor een project". De scrummaster kan helpen bij het opstellen van deze eisen.

3.2. Sprint backlog

Het ontwikkelingsteam is verantwoordelijk voor de op te leveren onderdelen van het eindproduct tijdens elke sprint. Gezien de situatie bij Los Banditos, bestaat het team hier doorgaans uit minimaal één ontwerper en één ontwikkelaar. Samen stellen zij de *sprint backlog* samen. De sprint backlog is een lijst met eisen die afgewerkt moet worden tijdens de eerstvolgende sprint. Deze eisen worden geselecteerd uit de product backlog. Aan het begin van elke sprint wordt bepaald hoe lang de sprint gaat duren (dit varieert tussen minimaal een week en maximaal een maand). Het ontwikkelingsteam voegt vervolgens eisen uit de product backlog toe aan de sprint backlog, totdat er genoeg eisen zijn geselecteerd om de sprint backlog volledig te vullen. De toegekende waarde voor de ontwikkelingstijd wordt hierbij als leidraad genomen. Elke eis wordt vervolgens opgedeeld in taken. Deze taken worden bij Los Banditos ingevoerd in MijnWerkers. Dit is het centrale punt voor het ontwikkelingsteam om hun taken op te slaan. Elke stagiair krijgt een eigen specialiteit toegekend binnen een project.

3.3. Scrummaster

De scrummaster is verantwoordelijk voor het begeleiden van het scrumproces. Dit betekent dat de scrummaster verantwoordelijk is voor het verloop van elk project. Daarnaast is de scrummaster verantwoordelijk voor het begeleiden van de verschillende bijeenkomsten die plaats vinden tijdens het scrumproces. Dit is de functie van Rogier. In deze functie kan Rogier alle projecten in de gaten houden zonder zelf te hoeven ontwikkelen.

3.4. Scrumproces

Tijdens het scrumproces vinden er verschillende bijeenkomsten plaats: sprintplanning, de dagelijkse scrum, *scrum of scrums*, *backlog grooming*, projectreview en sprintreview.

Sprintplanning

Tijdens de sprintplanning wordt de eerstkomende sprint samengesteld. Dit proces duurt maximaal acht uur. Tijdens de eerste helft van dit proces wordt de product backlog geprioritiseerd en de duur van de sprint bepaald. Dit gebeurt door het gehele scrumteam. Het tweede deel wordt door het ontwikkelingsteam gedaan, dat de sprint backlog samenstelt en indeelt.

Dagelijkse scrum

Aan het begin van elke werkdag komen de scrummaster en het ontwikkelingsteam samen voor een korte bespreking van maximaal 15 minuten. In deze tijd geeft elk aanwezig lid antwoord op de volgende drie vragen:

- Wat heb je gedaan?
- Wat ga je doen?
- Ben je nog problemen tegengekomen?

Door deze sessie blijft iedereen op de hoogte van de voortgang binnen het project. Eventuele problemen worden genoteerd door de scrummaster en worden buiten deze sessie om behandeld.

Scrum of scrums

Na de dagelijkse scrum is er ruimte voor de scrum of scrums. Hier worden taken besproken waarbij er een overlap aanwezig is tussen de verschillende specialiteiten.

Backlog grooming

Backlog grooming vindt plaats tijdens de sprints. Deze bijeenkomst duurt maximaal een uur. Tijdens deze bijeenkomst wordt de product backlog gevormd. Hierbij zetten de producteigenaar en het ontwikkelingsteam de eisen om naar verhaalvorm. Ook worden ze gesorteerd op prioriteit. Backlog grooming kan meerdere malen per sprint plaatsvinden.

Projectreview en scrumreview

Deze bijeenkomsten vinden beide aan het einde van een sprint plaats. De projectreview bestaat uit het beoordelen van het gemaakte werk en duurt maximaal 4 uur. De scrumreview bestaat uit het beoordelen van de sprint en duurt maximaal 3 uur. Bij beiden is het de bedoeling om de focus te leggen op verbetering. Wat ging er fout en hoe kan dit verbeterd worden?

4. Grafische richtlijnen

Het grafische proces voor een responsive design verschilt veel van het grafische proces voor een traditionele website. Dit komt omdat er nu rekening gehouden moet worden met de verschillende schermgroottes en -resoluties. De website zal er immers anders uitzien op een smartphone dan op een desktop of tablet. Om dit verschil te overbruggen wordt er gebruik gemaakt van twee frameworks.

4.1. Frameworks

Er zijn twee verschillende frameworks aangemaakt om het ontwerp- en ontwikkelproces te vergemakkelijken: een framework voor Joomla! (<http://demo.losbanditos.nl/joomla>) en een framework voor WordPress (<http://demo.losbanditos.nl/wordpress>). Beide frameworks zijn een schil waarmee alle projecten van Los Banditos gemaakt kunnen worden. De toepasbare omgeving (Joomla! of WordPress) wordt bij elk nieuwe project gekopieerd, zodat de basis voor een responsive design al aanwezig is (wat tevens de ontwikkeltijd verlaagt).

De frameworks geven een goed voorbeeld van het gebruik van content-out design en mobile first ontwikkeling (meer hierover in hoofdstuk 4.3.1). Beide frameworks houden dan ook geen rekening met de verschillende apparaten die op de markt te krijgen zijn, maar letten puur op de inhoud van de website.

Beide frameworks zijn gebaseerd op het gebruik van kolommen in rijen die over de volle breedte van het scherm spannen. Zo kan de ontwerper zelf kiezen hoeveel rijen aan inhoud hij wil hebben en hoe de kolommenindeling binnen deze rijen eruit ziet.



Figuur 4.1: Voorbeeld van het Joomla! framework, hier is duidelijk de rijen- en kolommenverdeling te zien

De tekst die in de kolommen staat is voorbeeldinhoud. Hier kunnen ook tabellen, afbeeldingen of andere inhoud in terecht komen.

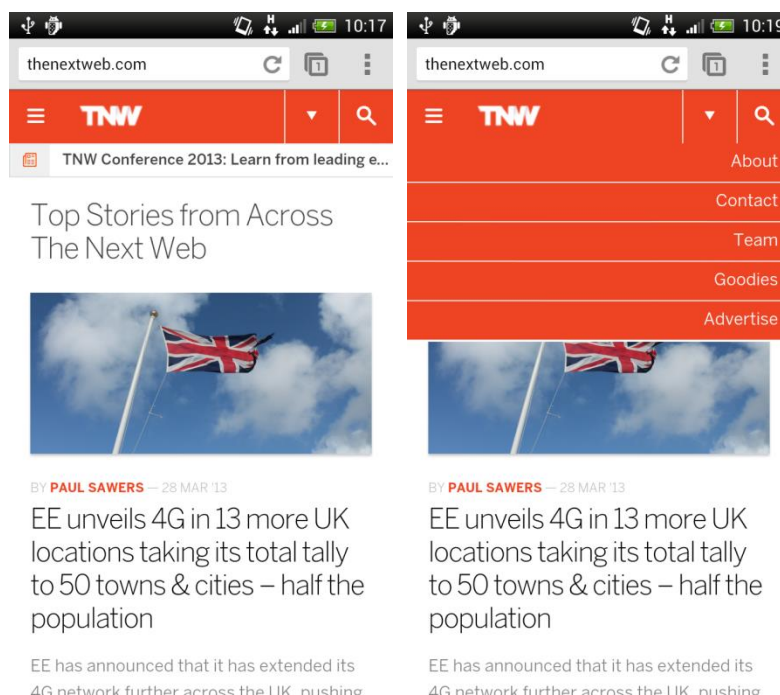
4.2. Navigatie

De navigatie in een responsive design is een apart onderdeel wat extra aandacht nodig heeft. Aangezien er veel minder ruimte is op een klein scherm, zijn er verschillende manieren om de navigatie te verwerken in het ontwerp voor de kleinste versie van de pagina.

Binnen het Joomla! framework kan er gewisseld worden tussen deze mogelijkheden in de menumodule. Tevens maakt het Joomla! framework ook gebruik van een mogelijkheid tot ondertiteling bij de menu-items. In figuur 4.1 is dit te zien. Bij het WordPress framework gebeurt dit door de code in het CSS bestand uit of aan te zetten (in comment te zetten of uit comment te halen).

Knop voor het menu

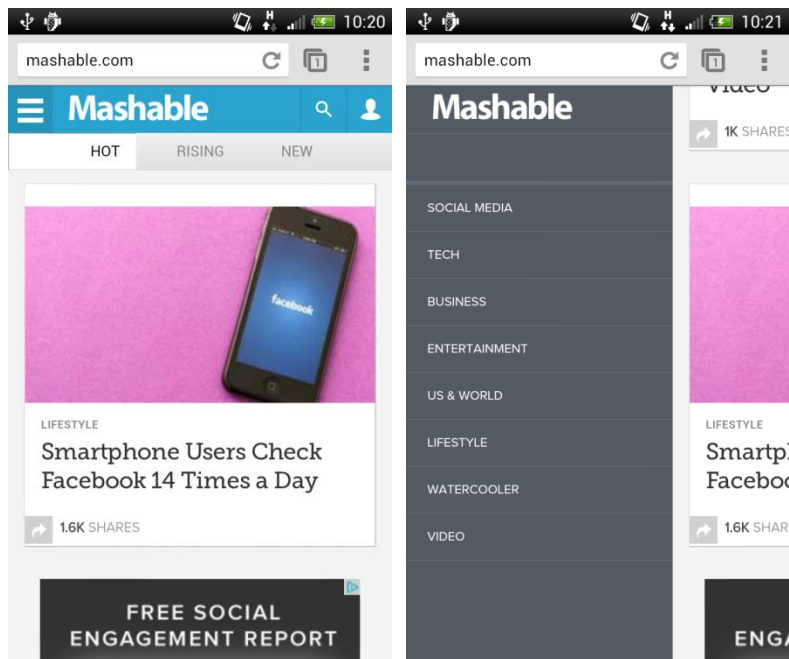
Bij kleinere schermen is dit de meest voorkomende oplossing voor het weergeven van het menu. Deze optie maakt gebruik van een knop, welke meestal links of rechts bovenin het scherm geplaatst is. Als er op deze knop geklikt wordt, dan leidt deze naar het menu. Dit wordt op verschillende manieren gedaan: het uitklappen van een accordeon, het verplaatsen naar de onderkant van de pagina of het uitschuiven van een venster over de pagina heen.



The Next Web

De website van The Next Web gebruikt een knop rechtsbovenin de pagina. Als hier op geklikt wordt, dan opent een accordeon met het menu. Deze komt over de pagina heen te liggen. Deze is ook weer te sluiten door op dezelfde knop te klikken. Opmerkelijk is dat The Next Web gebruik maakt van jQuery voor een slide-effect, terwijl dit in principe de snelheid van de pagina wel flink verlaagd.

Figuur 4.2: Voorbeeld van de werking van het menu van The Next Web



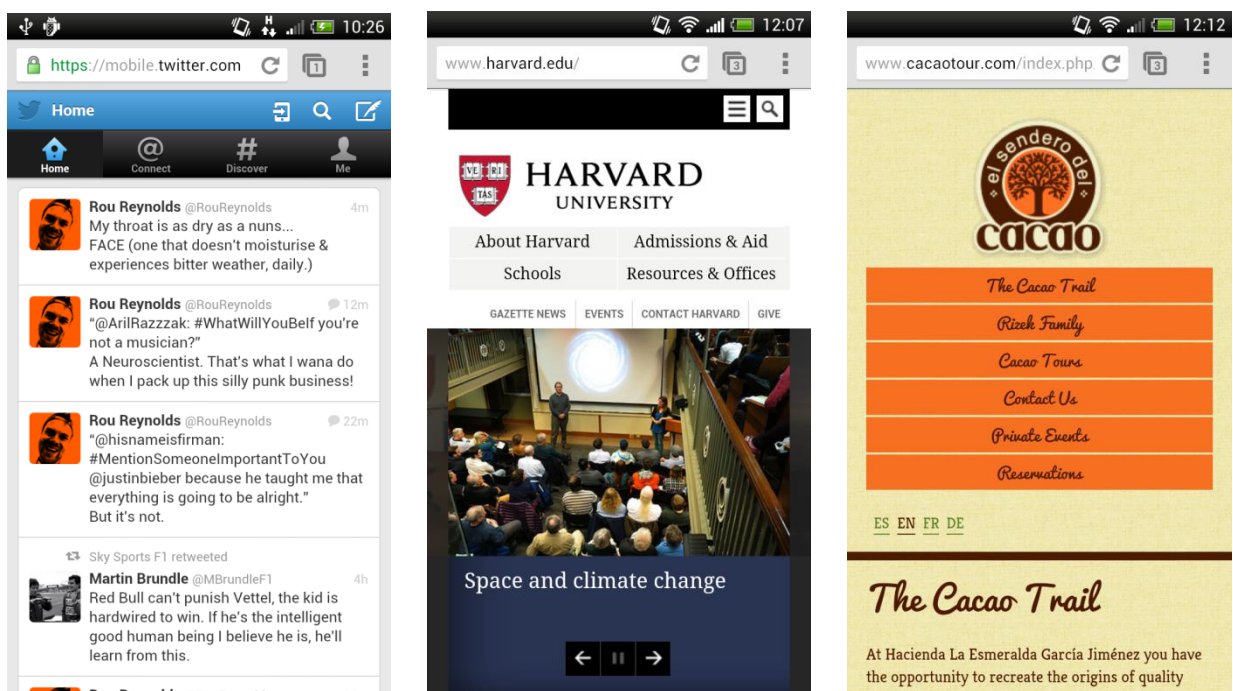
Mashable

Mashable maakt gebruik van een knop die linksbovenin de pagina staat. Deze knop zorgt ervoor dat een menu van de linkerkant van de pagina naar binnen komt 'sliden'. Dit menu neemt een groot gedeelte van de pagina in. Deze is echter op dezelfde manier weer weg te halen: door te sliden. Hier ligt de snelheid voorop als grootste slachtoffer. De meeste smartphones kunnen deze scripts niet gemakkelijk aan.

Figuur 4.3: Voorbeeld van de werking van het menu van Mashable

Menu weergeven bovenaan de pagina

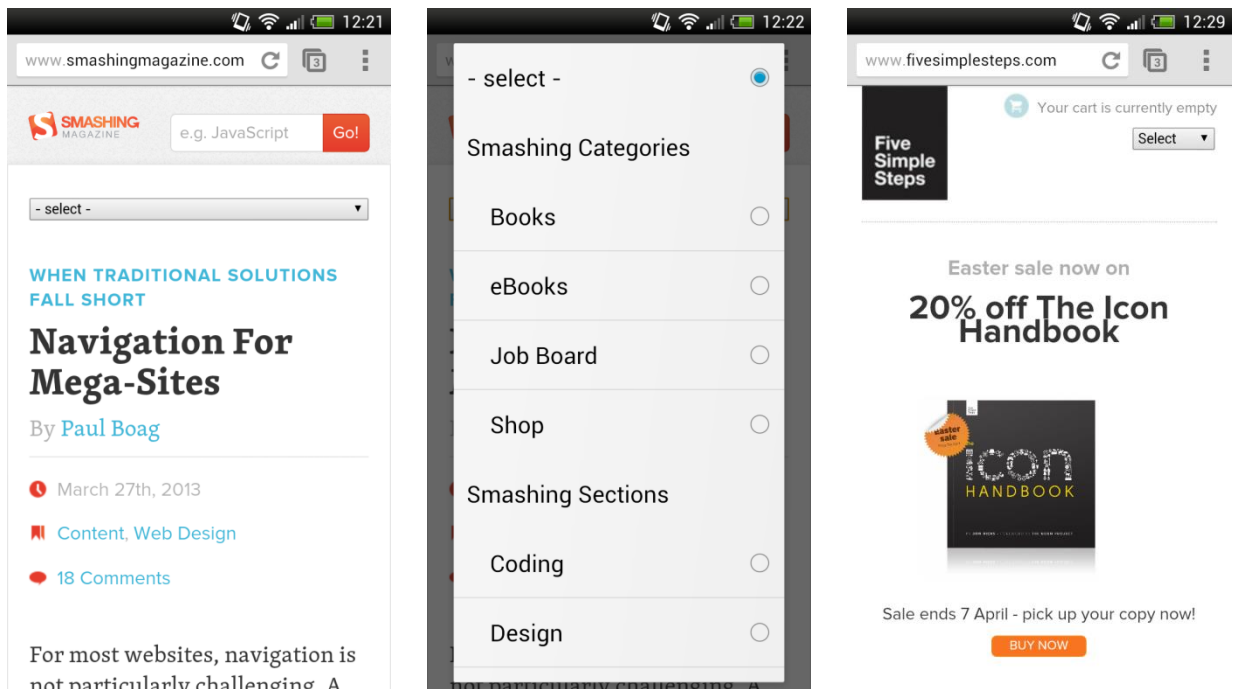
Deze stijl is het beste toe te passen bij pagina's met een simpele navigatie. Het grote voordeel hiervan is dat de navigatie niet verborgen is achter een knop. De toegankelijkheid van de website wordt hierdoor vergroot. Het nadeel is dat de pagina minder ruimte overlaat voor de inhoud van de website. Deze versie is het makkelijkste te implementeren, aangezien er niet veel veranderd ten opzichte van het menu op een traditionele website.



Figuur 4.4: Voorbeelden van Twitter, Harvard en Cacotour, die allemaal het menu weergeven bovenaan de pagina.

Dropdownmenu

Het gebruik van een dropdownmenu is een overzichtelijke manier voor uitgebreidere navigatie. Het nadeel is dat er een extra klik nodig is voordat een pagina bereikt kan worden. Daarnaast wordt het scherm gevuld door de opties binnen het dropdownmenu, in tegenstelling tot de minder schermvullende versie die men gewend is van de traditionele website.



Figuur 4.5: Voorbeelden van Smashing Magazine en Five Simple Steps. Beiden gebruiken een dropdownmenu.

4.3. Kolommen

De breedte van de kolommen wordt bepaald door de inhoud die te zien is in een kolom. Dit varieert dus per inhoudstype. Stukken tekst met regels tussen de 45 en 75 karakters zijn voor een mens het fijnste om te lezen. Op basis van de grootte van het lettertype en deze gegevens kan dus de minimale en maximale breedte voor een tekstkolom berekend worden. Voor kolommen met andere inhoud, zoals plaatjes of tabellen, is geen vaste regel te stellen. Dit ligt aan de inhoud die gevormd wordt en hoe deze zich verspreidt over de kolom. De ontwerper kan in dit geval zijn specialisme laten spreken en hier een oordeel over vellen. In ieder geval zal elke kolom een minimale en maximale breedte hebben, op basis waarvan de kolommenindeling in de webpagina geregeld wordt.

4.3.1. Wireframe

Het bepalen van de breedte van de kolommen gebeurt tijdens het wireframeproces. Een handige tool die hierbij gebruikt gaat worden is Pencil. Met Pencil kun je makkelijk zien wanneer een kolom te breed of te smal wordt. Bij het ontwikkelen van de kolommen en het bepalen van de indeling moet er vooral rekening gehouden worden met de minimale breedte van een kolom. Een te kleine kolom is onleesbaar, ongeacht het apparaat. Een te grote kolom kan wel voorkomen. Het is immers

bijna onmogelijk dat alle kolommen op elke resolutie exact binnen hun minimale en maximale waarde vallen.

4.3.2. Mobile first

Het ontwerpen van de website (en het wireframe) gebeurt vanuit de mobile first ideologie. Dit betekent dat er eerst een mobiele pagina wordt gemaakt en dat deze vervolgens uitgebreid wordt. Voor een responsive design is dit een stuk makkelijker dan werken vanuit de volledige website en deze vervolgens verkleinen. De mobiele website is tenslotte de meest simpele versie van de website. Hier zijn de kolommen vaak onder elkaar geplaatst.

4.3.3. Content-out design

Bij een responsive design wordt gebruik gemaakt van zogenaamde optimalisatiepunten. Deze punten geven aan wanneer de indeling van een website verandert. Dit heet content-out design. De verandering zal meestal duiden op het verplaatsen van kolommen en het aanpassen van witruimte (en dus breedte van kolommen). Het eerste optimalisatiepunt wordt bepaald door de minimale breedte van de twee belangrijkste kolommen bij elkaar op te tellen. Op dit punt kunnen deze kolommen naast elkaar staan, in plaats van onder elkaar. Tot aan dat punt, blijven de kolommen altijd onder elkaar staan. Deze procedure wordt gevolgd voor alle overige kolommen. Voor beide optimalisatiepunten kan er nog gespeeld worden met de witruimtes. Door het toevoegen of verwijderen van witruimte tussen de kolommen en naast de buitenste kolommen, kunnen de optimalisatiepunten verkleind of vergroot worden. Deze witruimte kan vooral gebruikt worden voor het breder maken van kolommen, zonder dat de inhoud van de kolom te breed wordt. Het gebruik van witruimte is per ontwerp verschillend en zal grotendeels bepaald worden door de ontwerper. Bij de frameworks wordt deze handeling ook gehanteerd. Deze kan als basis gebruikt worden.

De optimalisatiepunten zijn belangrijk voor zowel de ontwerper als de ontwikkelaar. Deze zullen dan ook samen veel contact hierover moeten hebben. In principe kan de ontwerper het bepalen van de optimalisatiepunten voor zijn rekening nemen, zolang hij contact onderhoudt met de ontwikkelaar over de gemaakte keuzes. De ontwikkelaar moet immers de optimalisatiepunten weten voor het technische ontwerp. Als de optimalisatiepunten bekend zijn, kan de ontwikkelaar al aan de slag met het technische ontwerp. Hier gaan we verder op in bij hoofdstuk 5.



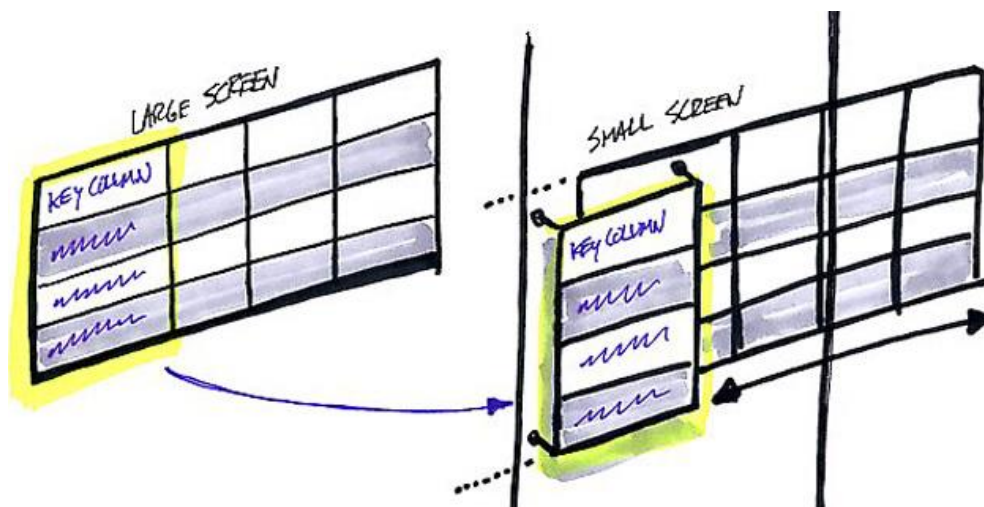
Figuur 4.6: Voorbeeld van een responsive kolommeindeling

4.4. Tabellen

Tabellen hebben een wijde variatie aan voorkomens. Ze kunnen lang zijn, veel kolommen bevatten, of juist het compleet tegenovergestelde. Het enige probleem met tabellen zit niet in de lengte, maar in de breedte. Er zijn een tweetal responsive oplossingen voor tabellen die veel kolommen bevatten.

Horizontale scrollbalk

Het toevoegen van een horizontale scrollbalk is een optie indien de tabel veel gegevens bevat. Op deze manier kunnen de gegevens nog steeds vergeleken worden met elkaar. Daarnaast zal de tabel in lengte niet toenemen, ondanks de schermgrootte. Een belangrijk gegeven is dat de eerste kolom wel altijd vast blijft staan, aangezien deze meestal de naam van de rij bevat.



Figuur 4.6: Voorbeeld van een horizontale scrollbalk bij tabellen

Rijgegevens onder elkaar

Dit betekent dat de verticale ruimte optimaal benut zal gaan worden. De rijgegevens zullen onder elkaar gaan staan, in plaats van dat ze horizontaal naast elkaar gaan staan. Dit geeft veel overzicht per rij, maar geeft geen mogelijkheid om rijen met elkaar te vergelijken. Ook zullen uitgebreide tabellen een veel te lange pagina opleveren.

4.5. Formulieren

Het gebruik van formulieren zorgt voor het toevoegen van invoervelden. Invoervelden zullen bij een grotere resolutie anders voor de dag komen dan bij een kleinere resolutie. Zo zal er bij een grote resolutie ruimte zijn om het label en het invoerveld naast elkaar te plaatsen. Bij een kleinere resolutie is dit niet mogelijk. In dit geval zal er gebruik gemaakt worden van **placeholders**. Hierbij staat het label in het invoerveld indien er niks is ingevuld. Als er iets wordt ingevuld, zal de placeholder verdwijnen.

First name
Last name
Submit

Figuur 4.7: Voorbeeld van placeholders in een invoerveld

4.6. Touch

Hoewel er geen specifieke technieken zijn om de touchschermen te ondersteunen, is het wel belangrijk om rekening te houden met de verschillende besturingsmogelijkheden van de apparaten. Zo moet er goed aan de grote van knoppen en links gedacht worden bij de kleinere weergaves, aangezien deze vaak door middel van touch gebruikt worden. Het gebruik van de vingers is veel minder nauwkeurig dan het gebruik van een muis. Het is aan te raden om klikbare elementen minimaal 7 mm hoog en breed te maken. Daarnaast zal er minimaal 2 mm tussen de verschillende elementen moeten zitten.

5. Technische richtlijnen

Voor het technische ontwerp van de pagina wordt grotendeels gebruik gemaakt van de relatieve eenheden em en procent. Deze eenheden passen zich aan de omstandigheden aan en zijn niet statisch, in tegenstelling tot pixels.

5.1. Frameworks

5.1.1. Joomla!

Voor het Joomla! framework is een aparte module gemaakt waarin verschillende opties gekozen kunnen worden voor de responsive wijze van het menu. Deze module (responsive menu), staat standaard ingeschakeld en beschikt over vier opties: select, dropdown, list, none. Select staat voor het gebruik van een selectbox. Dropdown geeft een knop, met daarbij een menu. List laat alle menu-items staan, maar deelt ze in op breedte. None geeft geen enkele optie.

Het Joomla! framework is ook uniek in zijn indeling in rijen en kolommen, zoals al te zien was in figuur 4.1. Het aantal rijen kan aangepast worden door de broncode van het template aan te passen. Het gebruik van kolommen gaat op een speciale manier. Dit kan de ontwikkelaar inrichten door middel van modules of componenten te binden aan een positie. Elke rij heeft 3 posities: A, B en C. Daarnaast wordt per positie aangegeven bij welke rij de positie hoort door middel van de rijnummer. Een template met 3 rijen zal dus 9 posities hebben: 1A, 1B, 1C, 2A, 2B, 2C, 3A, 3B en 3C. De breedte die een positie inneemt wordt bepaald op basis van de geactiveerde posities. Als er geen module of component gebonden wordt aan een positie, wordt deze niet geactiveerd (en dus niet weergegeven). Door middel van een **if - else statement**, wordt de breedte van elke actieve positie bepaald:

- Als positie A en B samen zijn geactiveerd, is A 33% breed en B 67% breed.
- Als positie B en C samen zijn geactiveerd, is B 67% breed en C 33% breed.
- Als positie A en C samen zijn geactiveerd, is A 50% breed en C 50% breed.
- Als positie A, B en C samen zijn geactiveerd, zijn A en C 33% breed en is B 34% breed.
- Als er maar één positie is geactiveerd (A, B of C), neemt deze 100% van de breedte in.

Deze kolommenindeling is al volledig responsive gemaakt. De media queries die hiervoor gebruikt zijn kunnen uiteraard wel aangepast worden op basis van het aangeleverde ontwerp.

5.1.2. Wordpress

Bij het WordPress framework zijn er al verschillende opties voor het menu ingesteld. Deze zijn verwerkt in het CSS bestand van het template. Door het in of uit comment halen van bepaalde code in het CSS bestand, kunnen er verschillende responsive oplossingen gekozen worden. Door middel van opmerkingen in het bestand, staat aangegeven welke code bij welke oplossing hoort.

Het WordPress framework kan gebruik maken van tags voor de inhoudskolom. Hier kan gekozen worden tussen de tags **[one_half]**, **[one_third]**, **[two_third]**, **[one_fourth]**, **[one_fifth]** en **[one_sixth]**. Deze tags geven de breedte aan die gebruikt wordt voor de tekst die tussen de tags staat. Als er dus

twee stukken tekst zijn met de tag `[one_half]`, dan wordt de kolom verdeelt in twee tekstkolommen van 50% breedte. In `shortcode.lib.php` in de template kunnen tags naar wens toegevoegd worden.

Daarnaast kan er gekozen worden tussen de hoeveelheid sidebars. Dit gebeurt op dezelfde manier als met het menu: door het in of uit comment halen van bepaalde code. Er is een keuze tussen één sidebar links, één sidebar rechts, één sidebar aan elke kant of helemaal geen sidebars.

5.2. Relatieve waarden

Een flexibele grid bestaat uit vier elementen: flexibele typografie, flexibele containers, flexibele margin en flexibele padding. We gebruiken hierbij de onderstaande formule:

$$\text{target} \div \text{context} = \text{result}$$

Target: de grootte die we in willen stellen.

Context: de grootte op basis waarvan we het element af willen stellen: de basiswaarde.

Flexibele typografie

De techniek achter flexibele typografie zit in het gebruik van de eenheid voor de grootte van het lettertype. Tot nu toe drukken webontwikkelaars dit meestal in pixels (px) uit. Voor een flexibele typografie zal er echter gebruik gemaakt moeten worden van em. Stel dat er een koptekst in het ontwerp staat met een grootte van 32px. De target zou in dit geval 32px zijn, de context 16px (font-size is standaard 16px in de meeste browsers). Het result wat hieruit komt is 2em. Het kan zijn dat het result op veel decimalen eindigt. Het is belangrijk om in dit geval zoveel mogelijk decimalen in te vullen. De comment is puur voor overzichtelijkheid.

```
font-size: 2em; /* 32px/16px */
```

Figuur 5.1: CSS code voor de grootte van het lettertype na het toepassen van de formule

Flexibele containers, margin & padding

Flexibele containers komen tot stand door het gebruik van procenten voor de breedte van een element. Over het algemeen zal er echter wel met een maximale breedte voor de pagina gewerkt worden. Er is geen meerwaarde aan het strekken van kolommen, als de inhoud dit niet toelaat. Voor het werken met procenten is het ook noodzakelijk om CSS3's nieuwe functie box-sizing op border-box te zetten. Hierdoor kan er nog steeds met padding en borders gewerkt worden, zonder dat deze bij de procenten opgeteld wordt. Dit maakt het een stuk makkelijker om de breedte van de kolommen te bepalen. Hier zal opnieuw het ontwerp gevolgd worden.

*(breedte element in pixels ÷ breedte container in pixels) * 100 = breedte in procenten*

Dit zal ook gebruikt worden voor het berekenen van de margin. De padding kan echter gewoon in een andere waarde uitgedrukt worden, aangezien deze door de box-sizing waarde niet optelt bij het geheel. Het berekenen van de padding met em wordt te ingewikkeld (aangezien em zich continu

aanpast aan het parent element, dus continu verandert). De andere optie is om de padding met rem in te stellen, een nieuwe waarde van CSS3. Rem werkt hetzelfde als em, alleen stelt zich af aan het HTML element, niet aan het parent element. Rem wordt momenteel echter nog niet door alle browsers ondersteund (Internet Explorer 8 ondersteunt het bijvoorbeeld nog niet). Zolang rem nog niet algemeen ondersteund is, zal de padding dus nog gewoon in pixels uitgedrukt worden.

```
* {
    -webkit-box-sizing: border-box; /* Safari 5.0-, Chrome 9.0- */
    -moz-box-sizing: border-box; /* Firefox */
    box-sizing: border-box; /* Internet Explorer 8.0+, Safari 5.1+,
                                Chrome 10.0+, Opera */
}
.container {
    max-width: 960px;
    margin: 0 auto;
}
#links {
    width: 48.958333%; /* (470/960) * 100 */
    margin-right: 2.604166%; /* (25/960) * 100 */
}
#midden {
    width: 32.8125%; /* (315/960) * 100 */
    padding-right: 15px;
}
#rechts {
    width: 15.625%; /* (150/960) * 100 */
}
```

Figuur 5.2: Voorbeeld van een template voor een ontwerp van 960 pixels breedte

5.3. Media queries

De stylesheets hebben altijd al een soort van apparaatherkenning gehad, in de vorm van mediatypes. Bij het laden van een stylesheet in de **head** van de HTML pagina, kon dus duidelijk gemaakt worden voor welk apparaat die specifieke stylesheet bedoeld was.

```
<link rel='stylesheet' type='text/css' href='screen.css'
      media='screen' />
<link rel='stylesheet' type='text/css' href='print.css'
      media='print' />
```

Figuur 5.3: HTML code voor een stylesheet alleen voor op het scherm en alleen voor een geprinte pagina

Het W3C is met de ontwikkeling van CSS3 hier verder op door gaan bouwen en heeft de media queries ontwikkeld. Door middel van media queries kun je specifieke karakteristieken van een pagina inspecteren.

```
<link rel='stylesheet' type='text/css' href='screen.css'
      media='screen and (max-width: 480px)' href='phone.css' />
```

Figuur 5.4: HTML code voor een stylesheet met de media query max-width

De bovenstaande stylesheet werkt dus alleen als het scherm een maximale breedte van 480px heeft. Dit betekent dat deze wel op een iPhone te zien is, maar niet op een normale desktop. Door middel hiervan kunnen er verschillende styles toegepast worden voor apparaten met verschillende kenmerken.

Voor een responsive design wordt meestal alleen gebruik gemaakt van de max-width en min-width media queries, aangezien deze kunnen beïnvloeden voor welke schermresolutie bepaalde opmaak gebruikt wordt. De media query is tevens toepasbaar binnen de stylesheet, als toevoeging van een @media of een @import regel.

```
@media screen and (max-width: 480px) { /* inhoud */ }
@import url('iphone.css') screen and (max-width: 480px);
```

Figuur 5.5: Een media query toegevoegd aan een @media en een @import regel

In verband met de laadsnelheid van de pagina, zullen we gebruik maken van @media regels om de media queries toe te passen zodat alle media queries in één document staan. De ontwerper heeft de verschillende optimalisatiepunten al berekend en aangeleverd. Deze punten kunnen in een media query verwerkt worden. Als voorbeeld zullen we een lay-out nemen waar de ontwerper de optimalisatiepunten op 500px, 650px en 780px heeft. We werken vanuit de mobile first ideologie, wat betekent dat de styles buiten de media queries voor de kleinste lay-out zijn, alle grotere lay-outs krijgen nieuwe styles (en dus eigen media queries). Mobile first wordt vooral gebruikt om de laadtijd van een pagina drastisch te verminderen. Aangezien het mobiele internet het langzaamste is, is het verstandiger om zo min mogelijk styles in te laden voor de kleinste lay-out. Een vuistregel die hierbij gebruikt wordt is **er mogen nooit styles overschreven worden**. De media queries mogen alleen styles **toevoegen**, niet **overschrijven**. Voor de waardes voor de media queries zullen we em gebruiken. Hierdoor zal er geen vervorming van de site plaatsvinden indien de gebruiker inzoomt op de pagina. Opnieuw wordt dezelfde formule toegepast als voor het berekenen van de font-size.

```
@media screen and (min-width: 31.25em) and (max-width: 40.625em) {
    /* content */
}
@media screen and (min-width: 40.625em) and (max-width: 48.75em) {
    /* content */
}
@media screen and (min-width: 48.75em) {
    /* content */
}
```

Figuur 5.6: Voorbeeld van het toepassen van de media queries

5.4. Viewport meta tag

De viewport meta tag zorgt ervoor dat de browser herkent met welke viewport je de website bekijkt. Als deze niet ingesteld wordt, geeft hij standaard de viewport van de desktopversie. Dit wil zeggen dat hij de breedte van de pagina (viewport) niet op het juiste aantal pixels instelt. Een responsive design zal dan op de iPhone dus nog steeds de desktopversie laten zien. Om dit te

voorkomen moet altijd de viewport meta tag toegevoegd worden aan de head tag van de HTML pagina. Deze tag is vereist in het geval van een responsive design. Het flexibele grid zal anders niet optimaal werken en zich niet aanpassen aan het apparaat. Voor websites van Los Banditos zal deze tag ook gebruikt worden.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Figuur 5.7: De viewport meta tag

De code is opgebouwd uit drie onderdelen. Allereerst geeft het aan dat de inhoud van de meta tag bedoeld is voor de viewport van de pagina, door middel van het name-element. Vervolgens geeft het nog twee kenmerken mee door middel van het contentattribuut. Allereerst zorgt het dat de breedte van de pagina ingesteld wordt op de breedte van het apparaat waarop de pagina bekeken wordt (`width=device-width`). Vervolgens zorgt hij er voor dat de pagina altijd met een schaal van 1:1 geopend wordt (`initial-scale = 1`).

5.5. Touch

Voor formulieren moet er altijd rekening gehouden worden met de `type`, `attribute` en `mask` attributen, zodat er verschillende toetsenborden zichtbaar worden op touchapparaten. Daarnaast moet er zo min mogelijk gebruik gemaakt worden van `:hover`, aangezien dit niet beschikbaar is op een touchapparaat.

5.6. Laadtijd

De laadtijd van een responsive design is altijd een groot probleem. Aangezien de smartphones en tablets niet over dezelfde internetsnelheid als een laptop of dekstop beschikken, zullen de websites zo minimaal mogelijk gebouwd moeten worden. Dit wordt allereerst al gedaan door de mobile first ideologie. Hierdoor worden er geen styles overschreven. Een aantal andere aanpassingen zijn het zo min mogelijk gebruiken van JavaScript en jQuery. Deze technieken eisen veel van de hardwaremogelijkheden en de internetsnelheid. Ook zal de CSS, JavaScript en jQuery altijd minified (het verwijderen van onnodige spaties en enters) moeten worden (dit is gemakkelijk te doen op <http://refresh-sf.com/yui/>), aangezien dit veel kan schelen in de grootte van de bestanden.

Een populaire techniek voor websites welke afbeeldingen bevatten is lazy loaden. Hierdoor worden de afbeeldingen als laatste geladen. Een goede jQuery plugin die hiervoor gebruikt kan worden is Lazy Load (<http://www.appelsiini.net/projects/lazyload>).

5.7. Browsersupport

Media queries worden ondersteund door Internet Explorer 9+, Firefox 3.5+, Chrome, Safari, Opera 9+ en alle mobiele browsers. Om toch volledige browserondersteuning voor de oudere versies van Internet Explorer te krijgen, kan er gebruik gemaakt worden van Respond.js (<https://github.com/scottjehl/Respond>).

6. Testen

Een ander groot punt is het testen van een responsive design. Er zijn namelijk veel apparaten, browsers en besturingssystemen waar het responsive design op moet werken. De enige mogelijke manier om een responsive design uitvoerig te testen is om deze op zoveel mogelijk apparaten en in zoveel mogelijk browsers en besturingssystemen te testen. De initiële test kan het makkelijkste gebeuren door de browser te vergroten en te verkleinen, om te zien hoe de website reageert op verschillende schermgroottes. Echter zal er pas een volledige test gedaan zijn als de website op meerdere apparaten met verschillende eigenschappen bekeken is. Er zijn twee oplossingen voor het testen op meerdere apparaten. De meest simpele manier is om een emulator te gebruiken. Deze bootst de eigenschappen van verschillende apparaten na, zodat er getest kan worden. Een tweede oplossing is het gebruik van een openbaar testlab. Dit is een locatie waar een verzameling aan fysieke apparaten aanwezig is, waar iedereen gratis zijn responsive web designs op kan testen. Netvlies heeft zo'n locatie beschikbaar gesteld in Breda.

Los Banditos heeft niet vaak hele grote projecten. Dit betekent dat de projecten over het algemeen niet meer dan één maand duren. Aangezien Los Banditos gelegen ligt in Rotterdam, is er gekozen om de kleine projecten te testen met het gebruik van een emulator. Responsinator (<http://www.responsinator.com>) is gekozen, aangezien deze een goed overzicht geeft van een responsive design op de meestgekochte apparaten. Eventuele grotere projecten kunnen getest worden in het openbare testlab van Netvlies in Breda.