

Vytvoření mobilní verze pro pracky-lednický.cz

Případová studie

Ing. Richard Ruibar
weto.cz 2013

Licenční ujednání:

Tento dokument může být volně sdílen, kopírován a množen pouze v nezměněné podobě ve formátu tohoto pdf dokumentu.

Citace jednotlivých částí mohou být volně prováděny s uvedením odkazu na zdroj:
<http://www.weto.cz/pripadove-studie/vytvoreni-mobilni-verze-pro-pracky-lednický.cz>

Vytvoření mobilní verze pro pracky-lednický.cz

V srpnu 2013 jsme v našem internetovém studiu řešili vytvoření mobilní verze pro web pracky-lednický.cz. Vypracovali jsme o tom tuto případovou studii.

Stručný úvod do mobilních webařských technologií

RES – Responzivní webdesign

Je plně v režii webového klienta (FF, Chrome ap.). Webové stránky jsou vytvořeny tak, že se přizpůsobují zařízení na kterém jsou prohlíženy. Tzn. zejména že se mění rozvržení stránky, velikost obrázků a grafické prvky podle velikosti displeje. To vše v závislosti na zařízení, kterému jsou stránky poskytovány.

Výhodou je, že stačí udržovat jen jeden web, který obsluhuje všechna zařízení. Tato technologie je vhodná především u menších webových sídel.

Nevýhodou je, že touto metodou nelze u větších sídel dosáhnout minimalizace datové velikosti stránky. Protože všichni klienti dostávají zcela stejný zdrojový kód, ale ani jeden nevyužívá vše co je mu posláno. Částečně lze zařídit omezení datové velikosti pomocí media queries. Některá zařízení ale načítají i ta data, která by neměla. Toto chování je známo u prohlížeče Mobile Safari na zařízeních iPad a iPhone.

„The problem with this approach is that some small screen browsers, most notably Mobile Safari on the iPhone and the iPad, will actually download both graphics, even if only one is ultimately applied to the page. While smaller screens don't always equate to lower bandwidth, we're currently punishing users on smaller screens with the download of a much heavier image than they'll ever see.“

Responsive Webdesign, Ethan Marcotte

Mobilní zařízení se často připojují přes mobilní internet, kde na jedné straně jsou datové přenosy drahé a na druhé straně pomalé. Proto se snažíme u mobilních verzí šetřit každý kilobajt.

Další nevýhodou RES je, že starší prohlížeče ji nezvládají.

Zdroje:

<http://www.abookapart.com/products/responsive-web-design>

<http://alistapart.com/article/responsive-web-design>

<http://www.techrepublic.com/blog/web-designer/how-to-create-media-queries-in-responsive-web-design/>

<http://www.zdrojak.cz/n/responsive-design/> aj.

RESS

RESS - je předchozí technologie rozšířená o to, že části zdrojových kódů jsou přizpůsobeny na straně serveru. Zdrojový kód, který se předává prohlížeči je responzivní (tedy čistě RES), ale na straně serveru se rozhoduje o tom, že některé části jsou vypuštěny nebo přidány v závislosti na tom, pro jaké zařízení je stránka posílána.

Výhody a nevýhody plynou z popisu předchozí technologie. Odstraňuje její největší nevýhodu tím, že umožní přizpůsobit zdrojový kód stránek podle prohlížečeho zařízení. Zůstává společná nevýhoda, kterou je nepodpora u starších prohlížečů. Přibývá nová nevýhoda, že řešení je finančně náročnější.

Zdroje:

<http://www.lukew.com/ff/entry.asp?1392>

Vlastní řešení

1. Samostatná verze nebo společná?

Jednou ze základních otázek je rozhodnutí zda pro mobilní zařízení vybudovat samostatný web běžící na subdoméně <http://m.pracky-lednický.cz> (příp. <http://mobil.pracky-lednický.cz>). Nebo řešit mobilní a desktopovou verzi jedním kódem (jak RES umožňuje).

U pracky-lednický.cz jsme se rozhodli takto. Pro desktopová zařízení se ponechá stávající webová verze. A pro mobilní uživatele se vybuduje nová verze na subdoméně m. Ta bude realizována technologií RES nebo RESS. Vedly nás k tomu tyto důvody:

Chybějící podpora pro media queries u starších prohlížečů

Pro RES je nejdůležitější podpora media queries, kterou nezvládají starší prohlížeče (např. IE8 a starší = všichni uživatelé WinXP, kteří používají IE).

Počátek masivního nástupu mobilních zařízení u nás lze spatřovat v roce 2010. Došlo tak k jistému paradoxu, kdy technicky méněcenná mobilní zařízení jsou vybavena modernějšími prohlížeči než mnoho desktopových řešení. Proto nám jako moudré připadá pro desktopová zařízení ponechat stávající webovou verzi, která je ošetřena i pro starší prohlížeče. A pro mobilní prohlížeče vyvinout odděleně responsivní variantu za použití moderních technologií. Tato verze potom nebude muset ošetřovat výjimky pro staré desktopové prohlížeče. Ošetřování výjimek pro starší prohlížeče je možné, ale vede k růstu datové velikosti stránek, čemuž se chceme obzvláště u mobilních zařízení vyhnout.

Některá mobilní zařízení mohou mít staré prohlížeče, bude jich ale málo a tak postačí když u nich ošetříme použitelnost a se vzhledem se nijak trápit nebudeme.

Optimalizace grafiky a vzhledu pro mobilní zařízení

1. Můžeme bez obav použít efekty CSS3 (přechody, stíny, radiusy). Tím se velmi sníží datová velikost grafiky, což je jedním z hlavních požadavků mobilní verze.
2. Můžeme zjednodušit grafiku a snáze minimalizovat datový objem zdrojů.
3. U mobilních webových stránek je potřeba aby dotyková plocha u tlačítek, odkazů ap. byla alespoň 44 x 44px. Tento požadavek může být v mnoha případech efektivnější řešit změnou html kódu než čistě pomocí CSS.
4. Možná se nám podaří vyhnout se nutnosti použití RESS a tím ušetřit klientovi peníze, protože taková verze je samozřejmě nákladnější než čisté RES. Nehrají zde roli jen jednorázové náklady při budování mobilní verze, ale mohou zde přibýt i udržovací náklady (nejvhodnější pro RESS je využít WURFL, která stojí 40USD na měsíc). Pokud se nám nepodaří RESS vyhnout, pak bude jistě potřeba v daleko menším rozsahu a pokusíme se obejít bez WURFLu.

2. Flexibilní rozvržení (layout)

Flexibilní layout není žádnou novinkou. Tou je ale rozsah rozlišení, pro která se řeší. Zatímco dříve jsme ošetřovali pružný layout pro možná rozlišení např. od 800px do 1280px (rozsah se samozřejmě v čase měnil dle úrovně pokroku u monitorů). Tak nyní se musíme zabývat rozsahem od 300px do 1280px i více. Proto již neřešíme jen to, že se část obsahu při nižším rozlišení přesune jinam, ale přicházejí na scénu i taková rozhodnutí jako vypuštění části obsahu u malých displejů.

Novinkou RES oproti staršímu přístupu pružných layoutů je také přizpůsobování velikosti obrázků a některých grafických prvků.

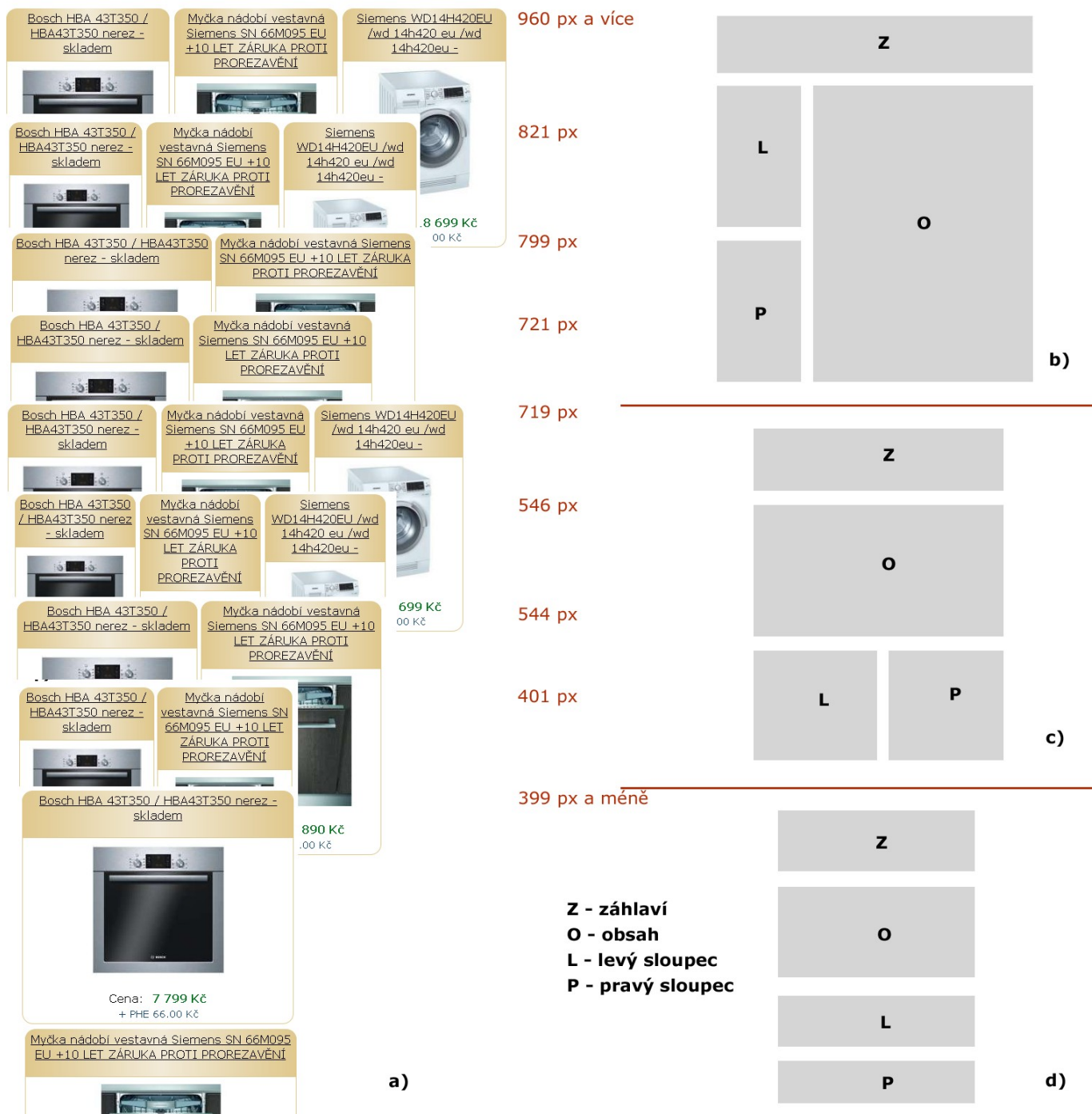
Nyní si ukážeme, jak se výsledné dílo (m.pracky-lednický.cz) chová při postupném zmenšování

velikosti okna. Toto můžete testovat zmenšováním velikosti okna prohlížeče nebo ve FF můžete použít tento bookmarklet: <http://lab.maltewassermann.com/viewport-resizer/>.

Celkový pohled na situaci znázorňuje obr.1. V jeho levé části jsou ve stejném měřítku znázorněny dlaždice jak se mění jejich šířka a počet sloupců při postupném zmenšování šířky obrazovky.

Prostřední sloupec udává šířku obrazovky v pixelech. Tato stupnice se vztahuje jak pro levou tak i pro pravou část obrázku 1.

V pravé části obr.1. jsou znázorněna rozložení stránky v odpovídajících případech.



Obr. 1:

Povšimněte si prosím, že při rozlišení 721px jsou dvě široké dlaždice, zatímco při zmenšení šířky obrazovky na 719px dojde k rozšíření prostoru pro dlaždice. Pak jsou tři sloupce dlaždic s menší šířkou. Je to dáno změnou rozvržení stránky, jak znázorňuje pravá část obrázku. Bloky označené L a P se přesunou z bočního sloupce dolů a tím vznikne více prostoru pro obsahovou část. Proto při

zmenšení šířky obrazovky dojde paradoxně ke zvětšení šířky obsahové části.

Následuje popis změn v mezních bodech šířky obrazovky. Mezi těmito body je uspořádání zachováno. Mění se ale šířky jednotlivých prvků a to jak celých bloků (záhlaví, obsah, levý a pravý sloupec), tak i v nich vložených obrázků a části grafiky.

Šířka obrazovky 960px

U displejů nad 960px je rozvržení podobné desktopové verzi. Jen pravý sloupec je přesunut pod levý (viz. obr.2). Obsah na titulní straně má tři sloupce dlaždic s výrobky (viz. obr.1a).

Zvažovali jsme rovněž uspořádání kdy by se pravý sloupec z desktopové verze přesunul mezi levý sloupec a obsah. Byly by pak vlevo dva úzké sloupce vedle sebe a vpravo obsah. Toto rozvržení by bylo praktické z hlediska koderských prací. Je však poněkud neobvyklé, i když se občas objevuje. Jedna ze zásad použitelnosti webu však je: „Nenuťte uživatele přemýšlet“, proto jsme od tohoto rozvržení nakonec upustili. Vzhledem k malému rozšíření tohoto typu layoutu by byl pro mnoho uživatelů nepřehledný a nezvyklý.

Další varianta, kterou jsme zvažovali by zahrnovala i stupeň s větším rozlišením a to nad 1 280px. Při něm by bylo uspořádání sloupců identické desktopové verzi. A až při rozlišení pod 1 280px by se design choval podle stejného scénáře jaký jsme zvolili. Takové řešení bychom z hlediska uživatele považovali zřejmě za nejlepší. Bohužel by pak ale nebylo snadné řešit změnu rozvržení z layoutu nad 1 280px na layout pod 1 280px. Bylo by nutné ho řešit metodou RESS a to za použití placené služby WURFL. Tento stupeň je však možné v budoucnosti přidat.

Při poklesu šířky pod 960px se vše zužuje. Přitom se rozšíří políčko pro zadání textu u vyhledávání na celý sloupec, protože vlivem zúžení levého sloupce se zalomí od tlačítka - je pak nehezké a zbytečně malé.

Šířka obrazovky 820px

Při 820px se dlaždice na titulce uspořádají do dvou sloupců, protože při třech sloupcích již vycházejí úzké. Současně se zmenšením počtu sloupců se dlaždice rozšíří.

Šířka obrazovky 720px

Při 720px se levý sloupec přesune dolů pod obsah. Je přitom rozdělen na dvě části, které odpovídají desktopovému levému a pravému sloupci. Tyto sloupce jsou vedle sebe pod obsahem stránky (viz. obr.1c). Při rozlišení nad 720px mobilní verze byly tyto sloupce nad sebou v levém sloupci (do takového uspořádání se vrátí opět při rozlišení pod 400px).

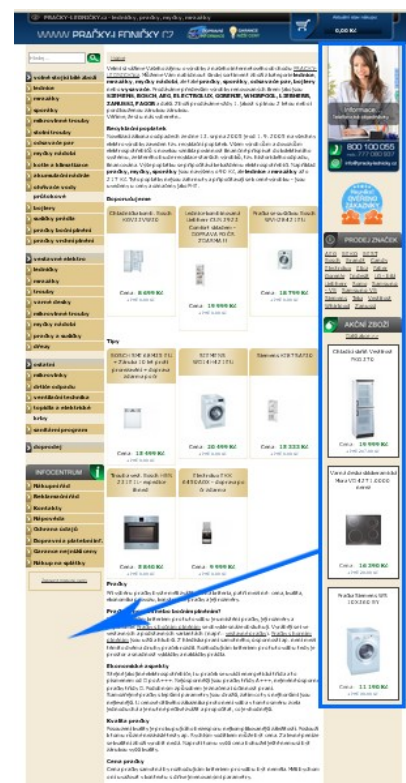
Současně s tím se rozšíří obsahová část a proto jsou nyní na titulce opět tři sloupce dlaždic.

Další změnou je vypuštění dvou informačních ikon v záhlaví vlevo.

Šířka obrazovky 545px

Při 545px se dlaždice na titulce opět přeuspořádají do dvou sloupců. Upraví se informace o košíku v záhlaví vpravo, kdy jsou nyní tyto informace na dvou řádcích místo tří.

Na detailu produktu se skryjí promoboxy, protože již jsou tak úzké, že delší slova vyčnívají přes okraje a zabírají mnoho prostoru, který by musel uživatel odrolovat.



Obr. 2

Šířka obrazovky 400px

Při poklesu pod 400px začíná jednosloupcový layout. Dlaždice na titulce jsou v jednom sloupci. Pravý a levý sloupec se opět přesunou pod sebe. Z pravého sloupce je velká část vypuštěna (Prodej značek a Akční nabídky), aby se délka stránky co nejvíce zkrátila a uživatelé nemuseli příliš rolovat. Také aby bylo dosaženo toho, že menu je na spodku stránky a za ním již je jen velmi málo obsahu. Tím je menu lépe přístupné.

Nyní již dochází i ke změnám na jiných stránkách. Na stránce s přehledem zboží je nyní zalomení u jednotlivých položek za obrázkem (viz. obr.3). Stejná změna je na detailu produktu.

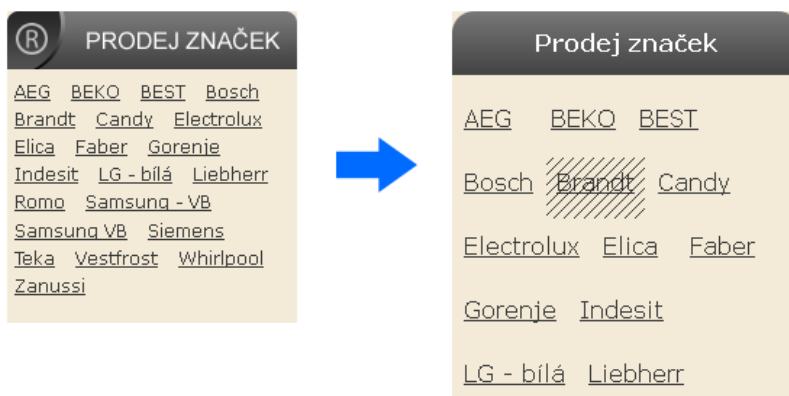


Obr. 3:

3. Tapovací plochy

Mobilní zařízení se obvykle ovládají dotykovou obrazovkou. Prsty bohužel nedokáží uživatelé „trefit“ konkrétní pixel jako myši. Přesnost tapnutí prstem je daleko menší. Společnost Apple proto doporučuje, aby minimální plocha pro tapnutí byla 44px x 44px. Tomuto pravidlu jsme museli přizpůsobit řadu prvků:

- vyhledávací políčko a jeho tlačítko,
- položky menu,
- položky drobečkové navigace,
- podsekce,
- omezení výrobců,
- tlačítka a vstupní pole,
- položky stránkovače,
- odkazy v seznamu výrobců.



Obr. 4: Zvětšení dotykových ploch u seznamu výrobců. Celá vyšrafovaná plocha je klikatelná (tapnutelná).



Obr. 5: Změna menu a vyhledávání. Celá šrafovaná plocha je klikatelná.



Obr. 6: Změna omezení výrobců. Kliknutí, či tapnutí do šrafované plochy způsobí změnu stavu checkboxu.

volně stojící bílé zboží - mrazáky

[pultové](#)

[průmyslové skříně](#)

[šuplíkové](#)

[příslušenství](#)



volně stojící bílé zboží - mrazáky

[pultové](#)

[šuplíkové](#)

[průmyslové skříně](#)

[příslušenství](#)

Obr. 7: Změna odkazů do podsekcí. I zde je možné tapnout do celé vyšrafované plochy.

1 2 3 4 5 6 7 8 9



Obr. 8: Změna stránkovače. Šrafováním je vyznačena tapnutelná plocha.

Obr. 9: Změna drobečkové navigace. Vyšrafovaná plocha je klikatelná.

Home » [volně stojící bílé zboží](#) » [mrazáky](#)



Home » [volně stojící bílé zboží](#) » [mrazáky](#)

Tyto změny mají za následek zásadní změnu vzhledu. Celý web nyní působí poněkud „hamatně“.

4. Absence hover efektu

Pokud u desktopu převedeme nějaký prvek kurzorem myši, může se změnit. Například položka menu se podsvítí, odkaz se zbarví či podtrhne, při přejetí odkazu se kurzor změní na ručku ap. U dotykových obrazovek, které jsou pro mobilní zařízení typické není možné takovou událost ošetřit, protože k ní nedochází.

Je proto potřeba, aby uživatel snadno rozpoznal ovládací prvky, klikatelné obrázky a koneckonců i odkazy v textu. Za tím účelem jsme dotykové plochy na některých místech obtáhli rámečkem. Do celého tohoto rámečku je možné tapnout (viz. obr. 6,7,8) a je dosaženo cíle. Jedná se například o podsekcce, checkboxy pro výběr výrobců či položky stránkovače.

S výjimkou položek levého menu jsou všechny odkazy podtrhány. Klikatelné obrázky jsou orámečkovány.



Obr. 10: Změna promo boxů z grafiky na HTML + CSS

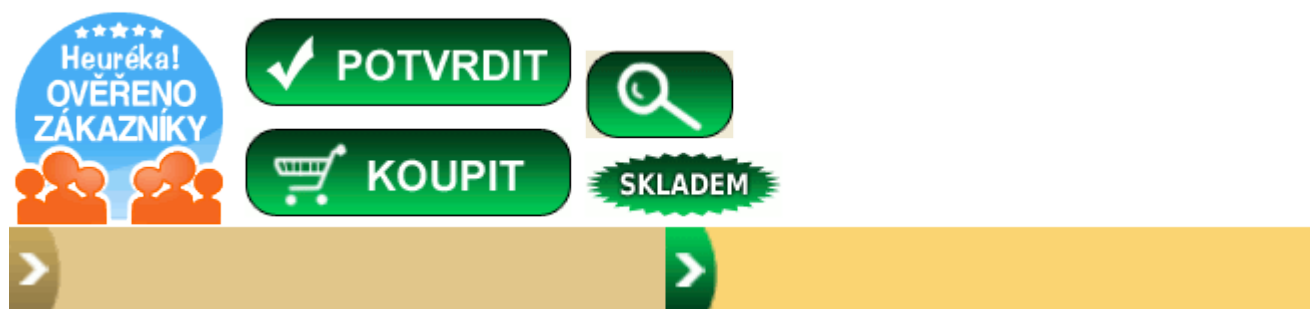
5. Duplicita

Vzhledem k tomu, že máme dva weby se stejným obsahem (mobilní a desktopová verze), museli jsme ošetřit duplicitu tím, že jsme do mobilní verze přidali meta tag canonical, který definuje jako kanonickou odpovídající URL desktopové verze.

6. Datové optimalizace

Na stránce detailu produktu se u desktopové verze pod obrázkem produktu zobrazují další obrázky s propagačními texty. Za účelem snížení datové velikosti jsme je u mobilní verze nahradili HTML+CSS verzí (viz. obr. 10). Tím jsme ušetřili 20kB.

Dalším optimalizačním krokem bylo použití spritů u grafiky. Jako obrázek se stahuje pouze logo, kvůli spolehlivějšímu ošetření responzibility (podpora pro border-size není dobrá). Web tedy, v rámci grafického layoutu, stahuje jen tři obrázky: logo a dva sprity (Pozn. sprite nezmenšuje datovou velikost, ale počet požadavků, což zejména u mobilního internetu vede ke zrychlení).



Obr. 11: Sprite

Omezili jsme používání JavaScriptů na minimum. Vzhledem k tomu, že máme verzi oddělenou pro mobily není nutné ošetřovat pomocí JS některé nefunkcionality starších prohlížečů. Obešli jsme se tak jak bez Response.js, Resolution.js, tak i bez Modernizr.js a případně další jiných potřebných podpůrných javascriptů. Ze stejného důvodu jsme se vyhnuli použití některého frameworku pro mobilní zařízení (např. Bootstrap). Jedinými JS ve stránce jsou kódy Google Analytics a TopListu a dvě drobnosti na několik řádků v košíku.

Načítání kódu Google Analytics jsme upravili tak, že k němu dochází až po načtení celé stránky. Tím se zrychlí vykreslení stránky. Při běžném způsobu vložení GA se vykreslovací jádro na kódu zasekne do doby, než stáhne celý javascript ze serveru Google. To někdy může trvat relativně dlouho.

Stylopsy parsujeme naším minimalizačním skriptem. Který jednak sloučí všechny použité css soubory, jednak z nich odstraní komentáře, mezery a znaky nových řádků. Toto jediné opatření vede k úspoře 10kB (také ošetřuje kešování v závislosti na změnách).

Na serveru je samozřejmě zapnuta komprese.

Snížením kvality jpg u generovaných náhledů jsme u stránky s přehledem zboží ušetřili 95kB. Na titulce 72kB. Na kvalitě zobrazení to není poznat. U desktopové verze má smysl ponechat vyšší kvalitu, kvůli případnému tisku.

Menší obrázky jsou vloženy do CSS jako DATA URI.

Jednou z komplikací byly ikonky Doprava zdarma a Garance nižší ceny, které se zobrazují při vyšších rozlišeních vlevo nahoře. Tyto ikonky jsou umístěny na pozadí s přechodem, takže jako průhledný gif nevypadají dobře a je zde nutnost použití formátu png (Pozn.: Png má 256 úrovní průhlednosti, zatímco gif jen jednu. Proto průhledný png vypadá vždy lépe než gif a v případě nestejnobarevného pozadí jsou gify obvykle nepoužitelné). Zařazení těchto dvou png obrázků do spritu by znamenalo, že by se musel převést na png, zatímco bez nich může být gifem. Jako gif však má poloviční datovou velikost než by měl jako png. Proto jsme se tomuto chtěli

vyhnout. V úvahu přicházely dvě jiné možnosti. Zaprvé vytvořit druhý sprite ve formátu png, který by obsahoval jen tyto dvě ikony. To by ale znamenalo požadavek na server navíc. Druhou možností bylo převést ikony do DATA URI a vložit je přímo do CSS - tuto možnost jsme nejprve zvolili. Při následné optimalizaci pro retinové displeje jsme ale museli přikročit k prvnímu řešení, kterým je samostatný sprite.

Použití moderní technologie CSS3 umožnilo výrazně snížit nároky na datovou velikost grafiky i na počet nutných požadavků na server. Stíny a radiusy již není nutné řešit pomocí obrázků, ale lze je snadno (a datově extrémně úsporně) nastylovat v CSS.

Výše uvedenými (ale i dalšími) opatřeními se dostáváme k datovým velikostem stránek okolo 100kB. Zčehož 15kB (v komprimované podobě) je Google Analytics a 45kB grafika (obrázky + CSS).

Níže jsou uvedeny velikosti tří stránek pro srovnání desktopové a mobilní verze. Rozhodující je „Transportní velikost“, což je „Velikost souborů“ po zkomprimování, ke kterému dochází před transportem.

Cesta	Desktopová verze		Mobilní verze	
	Velikost souborů	Transportní velikost	Velikost souborů	Transportní velikost
/ (Home page)	281 kB	231 kB	146 kB	91 kB
/bile-zbozi/lednice/	374 kB	302 kB	200 kB	118 kB
/bile-zbozi/lednice/ostatni/chladici-skrin-vestfrost-fkg-370	270 kB	224 kB	122kB	71 kB

7. Přepínání verzí

K přepínání mezi desktopovou a mobilní verzí dochází automaticky na základě detekce User Agenta. Nicméně je dobrým zvykem dát uživateli možnost manuálního přepnutí. Jednak pro případ, kdy automatický mechanismus selže. Jednak pro případ, kdy by si uživatel přál používat jinou verzi než která je mu určena.

8. Retinové a jiné displeje s vysokým rozlišením

Retina je latinský název pro sítnici oka. Pojem retinové displeje zavedl Apple v roce 2010 pro displeje s tak vysokým rozlišením, že při jejich držení v obvyklé vzdálenosti nedokáže lidské oko (sítnice) rozlišit jednotlivé pixely.

Např. u iPhone, který se běžně drží v průměrné vzdálenosti 25 cm je hraniční hodnota rozlišení 300 ppi (300 pixelů na čtvereční palec). U iPadu, který se drží ve vzdálenosti 38 cm se jedná o hodnotu o něco nižší.

Tato zařízení mají velký počet pixelů. Například iPhone 4, který má maličký displej o uhlopříčce 3,5 palce (cca 9cm) má rozlišení 640 x 960 px. Tzn., že při držení v režimu landscape (na ležato) má šířku 960px.

Pokud se náš responsivní design opírá o šířku displeje, nedostaneme se u takto malých displejů do potíží? Šířka 960px odpovídá naší nejširší variantě určené pro ta největší zařízení.

Naštěstí není pixel jako pixel.

Již v roce 2001 definovala W3C (organizace definující standardy pro www) webový pixel velmi prozíravě nezávisle na fyzickém rozlišení displeje. A to jako úhel, pod kterým je viděn pixel na zařízení s 90 ppi (později změněno na 96 ppi) na vzdálenost 71cm. Ve standardu je pro tuto vzdálenost vyčíslen jako 0,28mm (na každém zařízení). Tedy webový prohlížeč by měl jednotku 1px vždy zobrazovat tak, aby na příslušném zařízení měl asi 0,28mm - je-li zařízení používáno ve vzdálenosti 71cm. Pokud je dále, pak je velikost pixelu větší, např. při vzdálenosti 3,5m má pixel velikost 1,4mm. Při nižší vzdálenosti se naopak pixel zmenšuje. To vše je nezávislé na tom, jaké je skutečné fyzické rozlišení displeje. U zařízení s větší hustotou W3C doporučuje převzorkování. Více viz. zde <http://www.w3.org/TR/2001/WD-css3-values-20010713/>

Šťěstí máme ještě jednou a to v tom, že výrobci zařízení s vysokou hustotou rozlišení toto doporučení skutečně dodržují. A tak výše uvedený iPhone4 s fyzickým rozlišením 640x960 zobrazuje webové stránky jako by měl rozlišení jen 320x480, protože jeden webový pixel je realizován čtyřmi hardwarovými (2x2). V souvislosti s tím došlo k zavedení veličiny Device pixel ratio (dále jen DPR), která udává poměr mezi pixelem zařízení a webovým pixelem. Nejčastější hodnota dnes je 2 (týká se i iPhone4). U zařízení s nejhustšími displeji má dnes hodnotu 3 (HTC Butterfly, Samsung Galaxy S4, Sony Xperia Z). V případě DPR 3 to znamená, že jeden webový pixel je vykreslen 9ti hardwarovými pixely (3x3 = 9).

Díky tomu není nutné provádět speciální opatření pro rozvržení (layout) pro zařízení s vysokým rozlišením displeje (DPR > 1).

V úvahu však přichází jeden aspekt a to jsou obrázky, grafické prvky, fotografie. Všechny tyto prvky musejí být u těchto zařízení softwarově zvětšeny. V případě DPR 2 dvakrát, v případě DPR 3 třikrát. Takto softwarově zvětšované obrázky samozřejmě obvykle nevypadají dobře. Jsou rozmazané, či kostrbaté.

Princip řešení

Tato situace se řeší tak, že se upraví definice vzhledu pro takové zařízení tak, aby dostávala větší obrázky (např. 2x či 3x). Paradoxně se u nich ale musí poté definovat i zmenšení ve stejném poměru, aby měly správnou velikost. Tím se zajistí obvyklá kvalita grafiky.

Potíž je v tom, že každý obrázek pro DPR 2 musí být zvětšen 2x. To ale u plošného obrazce znamená 4x větší plochu (zvětšuje se 2x ve dvou směrech, vodorovně a svisle, a $2 \times 2 = 4$). U zařízení s DPR 3 se pak jedná o devíti násobek.

To znamená, že veškeré grafické prvky a fotografie mají v jednom případě 4x větší datovou velikost a ve druhém dokonce 9x. Což je velký problém z hlediska datových přenosů a rychlosti stránek.

Bohužel to vypadá, že hustota rozlišení je velmi dobrým prodejním argumentem výrobců těchto zařízení. Přestože již se dostali pod rozlišovací schopnost lidského oka, zvyšují hustotu bodů dále a dále. Je proto logické očekávat, že budoucí zařízení mohou mít DPR i vyšších hodnot než 3. V případě DPR 4 by grafika musela být 16x datově větší a u DPR 5 by se jednalo o 25ti násobek. Je smutné, že za tím vším stojí iracionální souboj o hustší a hustší displeje, kdy zvyšování hustoty již nemá praktický dopad.

Ošetření běžné grafiky

Za běžnou grafiku v tomto kontextu považujeme obsah spritů a logo. Jeden ze spritů je na obr.11, druhý vzniká složením ikon na obr. 12.

Všechny tyto obrázky jsme vytvořili zvětšené v následujících poměrech: 1,3x; 1,5x; 2x a 3x. Pro jejich názvy jsme použili appleovskou konvenci, kdy pokud se původní obrázek jmenuje `sprite.png`, pak jeho zvětšeniny se jmenují `sprite@2x.png`, `sprite@3x.png` atd. pro dvoj-, resp. trojnásobnou zvětšeninu.

Potom se ve stylpisu pomocí media queries předefinují pozadí jednotlivým prvkům. Například takto pro DPR 2

```
@media only screen and (-webkit-min-device-pixel-ratio: 2),
only screen and ( min--moz-device-pixel-ratio: 2),
only screen and ( -o-min-device-pixel-ratio: 2/1),
only screen and ( min-device-pixel-ratio: 2),
only screen and ( min-resolution: 192dpi), /* 2 x 96 */
only screen and ( min-resolution: 2dppx) {
  #mmenu li li, #imenu li, #mmenu li li:hover, #imenu li:hover,
  #box-vyrobci button, #searchform button, #produkty button,
  p.heu a, span.skladem {
    background-image: url(sprite@2x.gif);
    background-size: 656px 155px;
  }

  a.ico-doprava, a.ico-garance {
    background-image: url(sprite2@2x.png);
    background-size: 94px 49px;
  }
}
```

Totéž se zopakuje pro každé ošetřované DPR.

Pozn.: V případě spritu musí být u background-size uvedena velikost nezvětšeného obrázku. Není zde možno použít 50% 50% ap.

Ošetření fotografií

Zatímco ošetření grafiky je relativně jednoduché. Problémem je ošetření obrázků vkládaných přes tag `img`. V našem případě se jedná výhradně o fotografie produktů.

Jádro problému leží v tom, že toto nelze ošetřit pomocí CSS (které umí rozpoznat DPR). Dále je nutné zajistit, aby ve výchozím stavu byl v parametru `src` u obrázku odkaz na fotografii v původním rozlišení. Zatímco v případě displeje s vysokým rozlišením je třeba načíst obrázek s vyšším rozlišením. Toto by bylo možno řešit pomocí JavaScriptu. Taková řešení, ale mají své nevýhody.

Nejlépe by bylo, kdyby přímo server poznal jaký displej uživatel má a poslal mu správnou variantu obrázku. Problém je ale v tom, že toto by se muselo ošetřit na straně serveru. Ten ale nemá možnost jak rozpoznat DPR klienta.

Existují určitá řešení jak situaci řešit pomocí markapu (html tagů). Ta jsou ale zatím jen ve stadiu návrhu a jejich podpora je problematická.

Dále je k dispozici mnoho pluginů, které se dokáží s problematikou více (ale spíš) méně dobře poprat. Ta co jsme zkoumali trpěla vždy nějakým nedostatkem ať již prohřeškem proti přístupnosti, použitelnosti nebo velkým objemem dat. Nakonec jsme vymysleli vlastní řešení, později jsme zjistili, že na stejném principu funguje plugin Adaptive Images. Pro naše potřeby ale bylo jednodušší použít vlastní řešení popsané níže.

Obecné možnosti řešení

Situaci lze obecně řešit pouze za použití Javascriptu (dále jen JS). Což už samo o sobě je řešení, kterému se dobrý webdesigner snaží vždy vyhnout z toho důvodu, že určitá (byť malá) skupina uživatelů má podporu pro JS vypnutou nebo ji nemá. Zde ale bohužel není zbylí a je nutno takto postupovat.

Jednoduché JS řešení spočívá v tom, že se do stránky umístí kód, který dynamicky (v závislosti na DPR klienta) změní obsah parametru src u obrázků. Zde je ale problém v tom, že není možné tento JS spustit před načtením celé stránky, ale až v okamžiku kdy se již bohužel začínají obrázky stahovat. Toto řešení je sice možné, ale část obrázků se nejprve stáhne v původním rozlišení a je nahrazena. Což je zbytečně nekomfortní a vede to ke zbytečnému stahování obsahu, který se nepoužije (Když jsme tuto cestu testovali, stihl se stáhnout jen první obrázek v nižším rozlišení.).

Naše řešení

My jsme zvolili poněkud elegantnější řešení. Museli jsem se ale smířit s dalším ústupkem, který webdesignerů neradi dělají a sice použití cookies.

Použití JS a cookies není v tomto případě tak moc problematické, protože jediné co hrozí nemnoha uživatelům bez těchto technologií je to, že pokud mají displej s vyšším rozlišením, tak budou mít o něco méně kvalitní fotografie. Nehrozí zde žádná ztráta funkcionality, schopnosti navigace ap.

Pozn.: Všechny pluginy, které se k ošetření tohoto problému používají také používají JS a některé i cookies.

Naše řešení využívá toho, že všechny náhledy fotografií v tomto projektu (stejně jako ve většině našich projektů) jsou generované php skriptem „za letu“.

Na začátek sekce HEAD html stránky jsme umístili jednořádkový JS, který nastaví do cookie hodnotu DPR. Když v těle stránky dojde k načítání obrázků (voláním php skriptu pro generování thumbnailů) je již cookie uložena a pro tento skript přístupná. V php skriptu jsme pak provedli úpravu, aby podle hodnoty DPR zvětšil generovaný náhled. Vůbec se tedy nemění obsah atributu src a o tom jak velký obrázek se ze serveru pošle rozhoduje server na základě hodnoty DPR uložené v cookie.

Zároveň jsme výrazně snížili kvalitu jpg u těchto náhledů, které se generují pro zařízení s DPR > 1. Je totiž obecně známo, že to je možné. Metodou pokus/omyl jsme zjistili, že je v tomto případě únosné snížení na 30%. Zatímco obrázky pro DPR=1 u mobilní verze mají kvalitu 60% a pro desktop 100%. Bohužel toto snížení nevedlo k výraznějšímu snížení datového objemu. Přesto se snížil asi o 25%, což není vůbec zanedbatelné.

9. Testování

Testování jsme provedli pomocí online služby, která provádí testování na reálných zařízeních.

Testováno bylo na 12-ti zařízeních v režimu portrait i landscape: Android Nexus 7, Android Nexus S, BlackBerry PlayBook Tablet, Blackberry Torch 9800, Android - Fennec, Nook Color, iPad2, iPad Mini, iPad 3, iPhone 3, iPhone 4 a iPhone 5.

V rámci testování jsme zjistili potřebu další změny košíku pro mobilní uživatele. Provedli jsme v něm ještě následující dodatečné změny.

Vstupní políčka pro zadání fakturační adresy většina uživatelů nepotřebuje, protože se využijí jen tehdy, když je fakturační adresa odlišná od adresy dodání. Na malém displeji se přitom musí odrolovat. Nebo při vyplňování formuláře postupně přeskakovat jedno za druhým. Proto jsme ve výchozím stavu tato políčka ukryli a uživatel, který je potřebuje si je rozbalí.

Textové pole s poznámkou jsme změnili z TEXTAREA na INPUT typu text z následujícího důvodu. U malého displeje, kde při zadávání údajů překryje $\frac{3}{4}$ obrazovky klávesnice zbývá z webové stránky jen uzoučká nudle. Která byla vyplněna celou textareou a těžko se posouvalo jinam. Na Androidu, kde je možné se přesouvat mezi políčky INPUT[type=text] pomocí tlačítka se symbolem tabelátoru toto tlačítko u TEXTAREA zmizí. A uživatel nemá jednoduchou možnost jak pokračovat v zadávání údajů.

10. Výsledek

Na závěr přikládáme několik printscreenů výsledného díla. U mobilních verzí je zachován vzájemný poměr velikostí pro srovnání.

■ Desktopová verze

Obr. 13: Desktopová verze



Obr. 14: 320px

Obr. 15: 480px

Obr. 16: 720px

Obr. 17: 768px

Obr. 18: 1024px

