

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Bakalářská práce

2014

Zdeňka Hatnianská

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra systémové analýzy

Studijní program: Aplikovaná informatika

Obor: Informatika

Flexibilní design webových stránek pro různá zobrazovací zařízení

BAKALÁŘSKÁ PRÁCE

Student : Zdeňka Hatnianková

Vedoucí : Mgr. Ing. Zdeněk Smutný

2014

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze které jsem čerpala.

V Praze dne 11. května 2014

.....

Zdeňka Hatnianková

Poděkování

Chtěla bych poděkovat především vedoucímu této práce, panu Mgr. Ing. Zdeňkovi Smutnému, že si mě vzal pod svá křídla. Za jeho rady a zábavné konzultace v přátelském duchu. Dále bych ráda poděkovala všem, kteří mě rozptylovali od psaní této práce, a pomohli mi tak zachovat duševní zdraví.

Abstrakt

Tato bakalářská práce se zabývá tvorbou flexibilního designu webových stránek pro rozličná zobrazovací zařízení. Teoretická část práce pojednává o přístupech a strategiích k tvorbě webových stránek vhodných pro různá zařízení. Dále je věnována pozornost zejména responzivnímu přístupu k tvorbě webových stránek. Praktická část práce spočívá ve vytvoření vlastní webové HTML prezentace s responzivním designem. Webové stránky jsou následně otestovány v různých zařízeních, která umožňují prohlížení webových stránek. Doprovodnou součástí teoretické části je dotazníkové šetření. Jeho hlavním účelem bylo prokázat, že má smysl zabývat se tématem flexibilního designu webových stránek. Cílem bakalářské práce je návrh a realizace vlastní responzivně optimalizované webové stránky, za využití popsaných metod a vyhodnocení testování těchto stránek.

Klíčová slova

responzivní webdesign, separátní verze webových stránek, mobilní webdesign, media queries, webový prohlížeč, HTML5, CSS3

Abstract

This thesis deals with the creation of flexible website design for diverse display devices. The theoretical part deals with the approaches and strategies for building web pages suitable for different display devices. The thesis is mainly focused on responsive webdesign. In the practical part there is built a responsive web page. This website has been tested in a variety of display devices and web browsers. An additional part of the thesis is a questionnaire. The main purpose of the questionnaire was to prove that it makes sense to deal with the topic of flexible webdesign. The aim of this thesis is to design and built responsive website using the described methods and evaluation of testing of this website.

Keywords

responsive webdesign, separate versions of websites, mobile webdesign, media queries, web browser, HTML5, CSS3

Obsah

1	Úvod...	1
2	Flexibilní design webových stránek a separátní mobilní verze	3
2.1	Separátní mobilní verze	3
2.2	Webové stránky s flexibilním designem	5
2.2.1	Flexibilní mřížkové rozvržení	5
2.2.2	Flexibilní obrázky a média	6
2.2.3	Dotazy na médium	6
2.3	Separátní verze webových stránek vs. responzivní webové stránky	9
2.3.1	Pozitiva a negativa jednotlivých přístupů	10
2.4	Zásady tvorby webových stránek optimalizovaných pro mobilní zařízení	13
2.4.1	Metoda postupného vylepšování (progressive enhancement)	13
2.4.2	Mobile first v CSS	14
2.4.3	Snížení počtu requestů na server a datová velikost	15
2.5	Dotazníkové šetření	15
2.5.1	Zhodnocení dotazníkového šetření	22
3	Obecný postup tvorby responzivně optimalizované webové stránky	24
3.1	Vytvoření kompletního obsahu webových stránek	24
3.2	Návrh drátového modelu (wireframe)	24
3.3	Návrh grafiky	25
3.4	Tvorba HTML kódu	25
3.4.1	HTML	25
3.4.2	Zobrazovaná oblast	26
3.4.3	Media queries a prvky HTML5 pro starší verze Internet Explorer	27
3.5	Kaskádové styly	28
3.5.1	CSS	28
3.5.2	Základní kaskádové styly	29
3.5.3	Kaskádové styly pro různé velikosti obrazovky a breakpointy	29
3.5.4	CSS Frontend framework	30
3.6	Testování	30
4	Tvorba vlastní responzivně optimalizované stránky	32
4.1	Sestavení obsahu stránek	32
4.2	Drátový model	33
4.3	Vytvoření prototypu a grafického návrhu stránek	34
4.4	Sestavení HTML kódu	38
4.5	Vytvoření CSS šablony	39

4.6 Testování	44
4.6.1 Online nástroje pro a emulátory pro testování	44
4.6.2 Testování v různých zařízeních	47
5 Závěr.....	51
Seznam literatury.....	52
Seznam obrázků a tabulek	56
Seznam obrázků	56
Seznam tabulek	56
Přílohy.....	57

1 Úvod

Trh nabízí širokou škálu různorodých zobrazovacích zařízení, pomocí nichž lze prohlížet webové stránky. Tato zařízení běží na různých operačních systémech, mají odlišná rozlišení obrazovky a umožňují instalaci různých internetových prohlížečů. Prohlížení webových stránek nám dnes umožňují telefony, tablety, netbooky, herní konzole, elektronické čtečky knih, televize, hodinky nebo třeba domácí spotřebiče. Již dávno nestačí vytvářet obyčejné statické webové stránky pro desktopové počítače, které by sloužily pro všechny návštěvníky webu. Nyní je nutné vytvářet stránky s jednoduchým designem. Stránky, které budou přehledné a načtou se během pár vteřin, ať už k nim návštěvník webu přistupuje z jakéhokoliv zobrazovacího zařízení. Cílem teoretické části práce je popis možností optimalizace stránek pro různá zařízení, jejich porovnání a zhodnocení. Větší pozornost je však věnována strategii výstavby stránek s flexibilním designem, které se dokážou přizpůsobit celé řadě nejrůznějších zařízení. Tomuto přístupu se říká responzivní design. Responzivní design a způsob jeho tvorby, je hlavní náplní této práce. Alternativou ke stránkám s flexibilním designem, je vytváření separátních verzí webů. Od separátních verzí webu se v posledních letech upouští. Webdesignéři se stále více zabývají možnostmi výstavby stránek s flexibilním designem. Druhá kapitola popisuje přístup k tvorbě stránek s flexibilním designem a separátní verze webů. Oba přístupy jsou vzájemně porovnány a jsou popsána jejich pozitiva a negativa. Dále se druhá kapitola zabývá některými zásadami, které je dobré dodržovat při stavbě stránek optimalizovaných pro mobilní telefony. Tyto zásady je vhodné dodržovat jak při tvorbě responzivního webu, tak i při vytváření separátních verzí webu.

Téma flexibilního webdesignu nabývá spolu s rozrůstajícím se počtem různých zobrazovacích zařízení čím dál většího významu. Jedním z dílčích cílů bakalářské práce je obhájení tohoto tvrzení. Hlavně z tohoto důvodu bylo provedeno navíc k této práci doplňující dotazníkové šetření. Dotazník byl zaměřen na věkovou skupinu 15 až 30 let. Dotazník zjišťoval, zda lidé používají k prohlížení webových stránek mimo desktopových počítačů i jiná zařízení (a jak velké procento lidí tvoří tuto skupinu uživatelů), a tak prokázat smysl tématu bakalářské práce. Podařilo se nasbírat přes 260 odpovědí v dané věkové skupině. Výsledky dotazníkového šetření svědčí ve prospěch tvrzení, že flexibilní webdesign nabývá stále více na významu.

Praktická část bakalářské práce využívá principů popsaných v teoretické části k návrhu a následné implementaci webové stránky s responzivním designem, což je vlastně hlavním cílem celé bakalářské práce. Dílčím cílem praktické části je následné otestování stránek a popis podpory dotazů na média v moderních internetových prohlížečích. Webové stránky vytvořené v rámci praktické části jsou jednoduchou webovou prezentací o kreslení. Webová prezentace stojí především na technologiích HTML5 a CSS3. Využívá také některé JavaScriptové knihovny jako je Modernizr nebo jQuery. Webová prezentace je navrhována a následně implementována pro tři úrovně responzivnosti. Nesnaží se prostřednictvím dotazů na média detekovat konkrétní zařízení či kategorie různých zařízení (např. chytré telefony, tablety, desktopové počítače), ale o přijatelné a přehledné zobrazení stránek v různých zařízeních. Webové stránky jsou následně otestovány v některých emulátorech a online nástrojích a samozřejmě i v reálných zařízeních. Do testování bylo zahrnuto celkem dvanáct zařízení z kategorií smartphone, tablet, netbook/ mini notebook a desktopové zařízení. (Za desktopový počítač je v práci mnohdy považován i notebook, protože z hlediska responzivního designu mají tato zařízení prakticky stejný význam.)

2 Flexibilní design webových stránek a separátní mobilní verze

Webdesign (návrh webových stránek) je proces tvorby webových stránek od počáteční analýzy cílů a návrhu řešení projektu, přes tvorbu drátěného modelu webu, samotného grafického návrhu webu, kódování a programování až po následnou optimalizaci webu pro vyhledávače a marketing. Někdy je webdesign chápán též jen jako grafické zpracování webových stránek.

Webové stránky můžeme prohlížet ve velmi rozmanité škále zobrazovacích zařízení. Statické webové prezentace přestávají být dostačujícím řešením. Existuje více možností, jak webové stránky přizpůsobit potřebám různých zobrazovacích zařízení. Přirozeně se nejvíce řeší, jak přizpůsobit stránky pro chytré mobilní telefony. Smartphony jsou velmi rozšířené a počet mobilních přístupů neustále roste (Vyleťal, 2013). Webové stránky optimalizované pro mobilní telefony znamenají konkurenční výhodu. Podle studie společnosti Google je 67 % lidí ochotnější koupit si daný produkt či službu, pokud je stránka optimalizovaná pro mobilní telefony (Fisch, 2012). Naopak 61 % lidí stránku raději rychle opustí, pokud není přizpůsobena pro mobilní zařízení. Dříve se tento problém řešil vytvářením separátních verzí webových stránek. Poslední dobou začíná být stále populárnější přístup výstavby webových stránek, které se flexibilně přizpůsobují všem zařízením (tzv. responzivní design). Nicméně separátní verze webových stránek (případně nativní aplikace) svůj význam neztratily. Dnes jsou navíc často kombinovány s responzivním designem (Michálek a Držka, 2013). V této kapitole jsou rozebrány oba tyto přístupy, jejich pozitiva, negativa a porovnání - kdy který z nich zvolit.

2.1 Separátní mobilní verze

Tento způsob spočívá ve vytvoření speciální verze webových stránek, která je přizpůsobena potřebám mobilních telefonů. Webové stránky pro mobilní telefony jsou oddělené od verze určené pro uživatele tabletů, desktopových počítačů a jiných zařízení s velkou zobrazovací plochou. Někdy jsou také vytvářeny samostatné verze webových stránek určené pro uživatele tabletů. Vznikají pak až tři různé samostatné verze webových stránek.

Separátní webové stránky určené pro mobilní telefony se liší od verze pro ostatní zařízení často i svým kontextem. Tento přístup je založen na předpokladu, že uživatel mobilního telefonu hledá na webu jiné informace než uživatel přistupující z desktopového počítače. Uživatelům mobilních telefonů by měly být posílány jen ty nejnutnější informace. Mobilní telefony bývají totiž méně výkonné a mívají pomalejší připojení k Internetu. Je třeba dobře zvážit, které informace uživatel mobilního telefonu nejvíce ocení. Vybrané informace jsou poté na webových stránkách uspořádány do takové podoby, aby uživatel mobilního telefonu nemusel dlouho čekat, stahovat velké množství dat či hodně posouvat obsah (Castro a Hyslop, 2012).

Při vytváření samostatných verzí webových stránek je nutné provést detekci zařízení. Po detekci zařízení na straně serveru jsou ze serveru stahována data určená pro konkrétní zařízení. K detekci zařízení se dají využívat informace z HTTP hlavičky User-Agent. Pomocí této hlavičky můžeme identifikovat prohlížeč, ve kterém se webová stránka zobrazuje. Hlavička User-Agent se však liší v závislosti na verzi internetové prohlížeče i zařízení. Pro usnadnění detekce zařízení je dobré využít některou z veřejně dostupných databází mobilních zařízení. WURFL (Wireless Universal Resource File) je jednou z nejznámější databází tohoto druhu (Pešout, 2011). Usnadnit detekci zařízení na straně serveru nám může pomoci i některá z PHP knihoven, například knihovna Mobile Detect (Michálek, 2013a). Detekce zařízení má zatím svůj význam i v případě vytváření responzivních webů, kdy často potřebujeme zastaralým zařízením posílat jiné CSS šablony.

K vytváření mobilních webových stránek a aplikací pro weby s hlubokou strukturou se často používají nejrůznější PHP frameworky založené na architektuře MVC (Model, View a Controller). Architektura MVC rozděluje aplikaci do třech logických částí (Bernard, 2009a):

- *Model* – Zahrnuje data a business logiku (doménová logika) aplikace.
- *View* – Představuje zobrazení modelu a prvků uživatelského rozhraní. V prostředí webu se někdy za View označuje HTML. V reálu jde o serverový kód, který HTML kód generuje. Může jít i o formáty XML či JSON (Bernard, 2009b).
- *Controller* – Je ústřední výkonná jednotka starající se o provázání funkčnosti aplikace.

PHP frameworky jsou vlastně sadou PHP skriptů. Tyto frameworky usnadňují práci programátorům. Pomáhají eliminovat mnoho řádek kódu, který by jinak museli programátoři psát stále dokola. Na architektuře MVC je postaven i u nás velmi populární PHP framework Nette (Škrášek, 2008). V dnešní době se sice od separátních mobilních verzí upouští, ale u větších projektů mají stále svůj význam. Většina separátních mobilních verzí se kombinuje s principy responzivního designu. Mnohdy vývoj separátních webů nekončí jen u webové aplikace, ale jejím převedením do nativní aplikace určené konkrétním zařízením a prostředím. Takové aplikace jsou mnohem intuitivnější a lze je na rozdíl od webových aplikací snadno zpoplatnit (Castledine et al., 2013). K úpravě vzhledu se většinou používají frontend technologie, které zahrnují hlavně HTML a CSS šablony a JavaScriptové knihovny (Koperski, 2014).

2.2 Webové stránky s flexibilním designem

Trh se v oblasti vývoje nových zařízení, pomocí nichž lze přistupovat k Internetu, neustále rozvíjí. Nikdo neví, jaká nová zařízení budou vynalezena v budoucnu. Právě proto není zcela vhodné vytvářet samostatné verze webových stránek pro všechna tato zařízení. Místo toho je dobré zamyslet se nad strategií, jak vytvořit pouze jedny webové stránky, které se dokážou přizpůsobit všem zařízením.

Postup vytváření stránek, které se dokážou přizpůsobit téměř všem zařízením, se nazývá responzivní design. Zjednodušeně jde o to, že všem zařízením je dodán stejný HTML kód, ale zároveň jsou stránky upraveny pomocí kaskádových stylů pro různé velikosti obrazovek. Responzivní přístup k tvorbě webových stránek znamená nový pohled na tvorbu webu a jeho design (Castro a Hyslop, 2012). Tento postup vymyslel a popsal Ethan Marcotte. Přístup k vytvoření responzivně optimalizovaných stránek dle Marcotta stojí na třech základech (Marcotte, 2011):

- Flexibilní mřížkové rozvržení
- Flexibilní obrázky a média
- Dotazy na médium (*media queries*)

2.2.1 Flexibilní mřížkové rozvržení

Flexibilní mřížkové rozvržení layoutu je vlastně klasické rozvržení pomocí kaskádových stylů.

Ovšem místo pevně stanovených hodnot v pixelech u velikosti písma a vlastností `width`, `margin` a `padding` nastavujeme velikost pomocí procent. Díky tomu jsou všechny komponenty v mřížce relativní. Nastavíme-li pro element `<body>` velikost písma na 100 % (`font-size: 100%`), vykreslí se písmo v prohlížeči v defaultní velikosti 16 pixelů. Místo pixelů se používají relativní jednotky délky *Em*. Jednotky *Em* představují relativní velikost vztahenou k velikosti písma pro daný element HTML. Převést pixely na jednotky *Em* lze jednoduše dle tohoto vzorce (Marcotte, 2011):

$$\text{cíl} \div \text{kontext} = \text{výsledek}$$

Cíl označuje hodnotu zadanou v pixelech, kterou chceme převést do jednotek *Em*. Kontext je v našem případě defaultní velikost písma, což je 16 pixelů. Tento vzorec funguje za předpokladu, že je elementu `<body>` nastavena velikost písma na 100 %. Všechny ostatní elementy, pak zdědí tuto vlastnost elementu `<body>` a webový prohlížeč je bude defaultně vykreslovat ve velikosti 16 pixelů. Chceme-li například převést 24 pixelové písmo do relativních jednotek *Em*, uděláme to následovně: $24px \div 16px = 1,5em$.

2.2.2 Flexibilní obrázky a média

Z obrázků a jiných typů médií můžeme rovněž udělat flexibilní prvky, které se budou přizpůsobovat velikosti obsahu. Abychom toho dosáhli, musíme jim nastavit velikost v procentech. Dělá se to pomocí vlastnosti `max-width`. Vlastnosti `width` a `height` v elementu `<img>` nezažíváme.

2.2.3 Dotazy na médium

Dotazy na médium jsou součástí jazyka CSS3. Pomocí nich můžeme přizpůsobovat design webových stránek vlastnostem média. Jde například o šířku zobrazovací plochy internetového prohlížeče. Kaskádové styly na určitý typ média můžeme aplikovat pomocí vložení direktivy `@media` do pravidla stylu v dokumentu CSS.

Vlastnosti médií:

Do dotazů na médium můžeme zahrnout následující vlastnosti médií (W3C, 2012):

- `width` – Pomocí vlastnosti `width` se můžeme dotazovat na šířku zobrazované plochy výstupního zařízení. Lze používat předpony `min` a `max`.

- `height` – Pomocí vlastnosti `height` se můžeme dotazovat na výšku zobrazované plochy výstupního zařízení. Lze používat předpony `min` a `max`.
- `device-width` – Pomocí vlastnosti `device-width` se můžeme dotazovat na šířku obrazovky výstupního zařízení. Lze používat předpony `min` a `max`.
- `device-height` – Pomocí vlastnosti `device-height` se můžeme dotazovat na výšku obrazovky výstupního zařízení. Lze používat předpony `min` a `max`.
- `orientation` – Tato vlastnost slouží k určení, zda je zařízení natočeno na šířku (`orientation: landscape`) či na výšku (`orientation: portrait`).
- `aspect-ratio` – Vlastnost `aspect-ratio` je definována jako hodnota poměru šířky vůči hodnotě výšky obrazovky výstupního zařízení. Lze použít předpony `max` a `min`.
- `device-aspect-ratio` – Tato vlastnost slouží k detekci poměru stran obrazovky. Například (`device-aspect-ratio:32/18`). Používají se zde hodnoty `device-width` a `device-height`. Lze použít předpony `max` a `min`.
- `color` – Funkce média `color` popisuje počet bitů na barevnou složku výstupního zařízení. Pokud zařízení není barevné, hodnota `color` je nula (`color:0`). U této vlastnosti lze použít předpony `max` a `width`.
- `color-index` – Vlastnost `color-index` popisuje počet položek v barevné paletě výstupního zařízení. Pokud zařízení není barvené, je hodnota vlastnosti `color-index` nula. `Color-index` umožňuje používat předpony `max` a `width`.
- `monochrome` – Vlastnost `monochrome` popisuje počet bitů na pixel v zařízení, které nezobrazuje barvy ale jen různé odstíny šedi. Není-li výstupní zařízení monochromní, hodnota vlastnosti `monochrome` je nula. Lze využít předpony `min` a `max`.
- `resolution` – Vlastnost `resolution` slouží k určení rozlišení výstupního zařízení (hustoty pixelů). Hodnoty vlastnosti se mohou udávat v jednotkách dpi (dots per inch) či dpcm (dots per centimeter). `Resolution` umožňuje používat předpony `min` a `max`.
- `scan` – Vlastnost `scan` popisuje proces vykreslování obrazu na televizních výstupních zařízeních (tv). Může nabývat hodnot `progressive` a `interlace`.

- grid – Grid se využívá k dotazu, zda je médium funguje na bázi bitmapy (běžná obrazovka) nebo využívá například nějaký „tty“ terminál (zařízení s omezeným počtem znaků, např. obrazovky starších mobilních zařízení). Grid může nabývat pouze dvou hodnot. Hodnota nula značí, že se jedná o bitmapový display.

Logické operátory a syntaxe dotazů na médium:

Formulovat dotazy na médium můžeme přímo v CSS souboru nebo vložit do odkazu na externí šablonu stylů. V následujících výpisech kódů (Výpis 1, Výpis 2) je naznačeno, jak dotazy na médium zapsat (Coyier 2013; Castro a Hyslop, 2012).

Výpis 1: Ukázka elementu <link> s atributem media.

```
<link rel="stylesheet" media="logika typ and (vlastnost-media:hodnota)"  
href="sablon.CSS" />
```

Výpis 2: Ukázka dotazu na médium uvnitř šablony CSS.

```
@media logika typ and (vlastnost-média:hodnota) {/*pravidla stylů*/}
```

Logika může nabývat hodnot *only* (jen) nebo *not* (negace). Klíčové slovo *only* zaručí, že starší webové prohlížeče, které nepodporují CSS3 nebudou načítat zbytek dotazu na médium. *Not* přiřazuje dotazu na médium opačnou pravdivostní hodnotu.

Typ označuje typ média, což je například *screen* nebo *print*. *Vlastnost-media* může nabývat některou z vlastností médií. U všech vlastností kromě *color*, *color-index* a *monochrome* nemusíme uvádět jejich hodnotu. Výraz *vlastnost-media:hodnota* se píše do kulatých závorek, před které píšeme logický operátor *and*.

Pomocí operátoru *and* můžeme spojovat také dvojice hodnot. Oddělováním jednotlivých dotazů na médium pomocí čárky z nich můžeme vytvářet seznamy dotazů na médium. Celý seznam se pak stává pravdivým, pokud platí aspoň jeden z jeho dotazů.

V následujících výpisech (Výpis 3, Výpis 4) je ukázka příkladů dotazů na médium.

Výpis 3: Konkrétní příklad dotazu na médium uvnitř elementu <link> s atributem media.

```
<link rel="stylesheet" media="only screen and (orientation:portrait)"  
href="styl.css" />
```

Tento dotaz načte dokument CSS pojmenovaný *styl* a použije jeho pravidla v případě, že médium je typu *screen* a je natočeno na výšku.

Výpis 4: Konkrétní příklad dotazu na médium uvnitř dokumentu CSS, tzv. *breakpoint*.

```
@media only screen and (min-width:480px) and (max-width:860px) {/*pravidla stylů*/}
```

Tento dotaz uplatní naše pravidla stylů v případě, že obrazovka je široká minimálně 480px a maximálně 860px.

2.3 Separátní verze webových stránek vs. responzivní webové stránky

Na začátku vývoje webu či webové aplikace pro mobilní a jiná než desktopová zařízení stojíme před rozhodnutím, jakou konkrétní strategii zvolit pro vyřešení tohoto úkolu. Tato podkapitola je zaměřena na určení vhodné strategie. Obecně lze říci, že záleží hlavně na velikosti projektu, který chceme realizovat. Pokud je naším cílem vytvořit jen jednoduchou webovou prezentaci, je vhodnější zvolit variantu responzivně optimalizovaného webu. Je-li však naším úkolem vytvořit složitější webové stránky s hlubokou strukturou, je výhodnější vytvořit samostatnou verzi webu pro mobilní zařízení, kterou lze kombinovat s responzivním webdesignem (Držka a Michálek, 2013).

Chceme-li, aby webové stránky nabízely uživatelům mobilních zařízení jiný obsah než plnohodnotná verze webu, je lepší zvolit separátní verzi webových stránek. Pokud lidé navštěvující stránky z mobilních telefonů používají jiná klíčová slova než návštěvníci z desktopových počítačů, jde o výhodnější variantu i z pohledu SEO (Search Engine Optimization – optimalizace pro vyhledávače). Obecně je však responzivní web považován pro optimalizaci SEO za přijatelnější. Zejména protože je tento přístup upřednostňován společností Google (Taylor, 2013; Forgáč, 2013). SEO je proces úpravy webových stránek, který usnadní vyhledávacím strojům procházení, indexování a pochopení obsahu stránek. Ve výsledku mají tyto úpravy zásadní vliv na jejich umístění ve výsledcích vyhledávání a tedy i dojem na uživatele (Fishkin, 2013).

Důležité je také vědět, jak velký podíl z celkového počtu návštěv webových stránek tvoří lidé, kteří navštěvují webové stránky prostřednictvím mobilního telefonu. Bohužel tuto informaci nemáme před začátkem tvorby úplně nového webu k dispozici. Ovšem obecně počet návštěv z jiných než desktopových zařízení neustále roste, proto je lepší nevytvářet jen klasické statické webové stránky.

Například studie Walker Sands Communications Quarterly Mobile Traffic Report uvádí, že počet mobilních přístupů v Q2/2013 byl na sledovaném vzorku webů téměř 30 %. Ve srovnání s Q2/2012 narostl počet mobilních přístupů na těchto webech o 70 % (Melich, 2014).

Jsou-li prioritou mobilní návštěvníci webu, poté je vhodné zvážit variantu tvorby nativní mobilní aplikace. Nativní aplikace jsou budovány přesně na míru pro konkrétní zařízení. Mohou to být například aplikace pro iPhone, Android atd. Nativní aplikace přinášejí svým uživatelům lepší uživatelský prožitek. Mohou zachytit události jako tažení (drag) a multitouch, přistupovat k hardwaru jako je GPS či kamera a nabízejí mnohem více možností než webové aplikace (Bílek, 2011).

Projekt by neměly příliš ovlivňovat finance. Špatně fungující nebo nevzhledný web potencionálního zákazníka pravděpodobně odradí. Pokud máme omezený rozpočet, je vhodné zůstat jen u statického webu, případně responzivního.

2.3.1 Pozitiva a negativa jednotlivých přístupů

Jak separátní mobilní verze tak i responzivní design mají svá pozitiva i negativa. Právě také jejich popis může pomoci s rozhodnutím, jakou strategii vývoje webu zvolit:

a) Separátní mobilní verze

Hlavní výhodou tohoto přístupu je, že se můžeme zaměřit jen na potřeby uživatelů mobilních zařízení. Separátní verze webu nám umožňuje perfektní optimalizaci webu pro konkrétní zařízení. Pro webové stránky je vybírán jen určitý obsah z plné verze webu, který je považován za nejdůležitější pro uživatele mobilních zařízení. Uživatel mobilního telefonu tedy nemusí stahovat velké množství dat a stránky se načítají rychleji. Uživatel také není nucen stahovat prvky, které jsou na konkrétních zařízeních nefunkční.

Ořezání domovského webu však skrývá také své nevýhody. Vybereme-li pro mobilní verzi stránek nevhodný obsah, uživatel s velkou pravděpodobností přepne na plnohodnotnou verzi webu pro desktopové počítače. Mobilní verze tak zcela pozbývá význam. Stejně tak se stane, pokud navrhne nevhodné uživatelské prostředí pro mobilní web, ve kterém se uživatel nebude moci zorientovat.

Velkým problémem je detekční mechanismus, pomocí kterého rozpoznáváme přistupující zařízení. Na základě tohoto mechanismu odesíláme konkrétní verzi webu.

Při detekci zařízení jsou využívány informace, které o sobě uživatel poskytuje při zasílání požadavků serveru. V internetových prohlížečích lze řetězec uživatelského agenta snadno změnit a často se to i děje za účelem zmaření detekčního mechanismu. Riziko tedy spočívá hlavně v tom, že přistupující zařízení detekujeme špatně. Dále také můžeme způsobit problémy uživatelům, kteří přistupují z neznámého zařízení.

Náklady na vývoj a provoz více verzí webu bývají poměrně vysoké. Většinou si to mohou dovolit jen větší společnosti. Mezi další nevýhody můžeme zařadit delší a složitější URL adresu webových stránek. Mobilní verze webu bývají součástí subdomény *m.* nebo *mobile.* Nejlepším řešením při separátních verzích webu je ponechat uživateli možnost přepnout si mezi verzí pro mobilní zařízení a desktopové počítače. Jednotlivé verze webových stránek tedy od sebe oddělujeme, ale neizolujeme (Castledine et al., 2013; Forgáč, 2013).

b) Responzivně optimalizované webové stránky

Největší výhodou responzivně optimalizovaných webových stránek je bezpochyby fakt, že máme jen jednu verzi webových stránek. Takto optimalizované stránky se přizpůsobují různým zařízením. Jde tedy o řešení, které je méně nákladné než tvorba a provoz několika různých verzí webu. Pomocí dotazů na média můžeme vzhled webových stránek přizpůsobit i pro zařízení, která v době vzniku webových stránek ani neexistovala. Responzivně optimalizované webové stránky většinou neumožňují uživateli, aby se přepnul na jejich plnohodnotnou verzi, jako je tomu u separátních webových stránek. Pokud tedy navrhujeme uživatelské prostředí pro menší obrazovky nevhodně a uživatelé se v něm nedokážou zorientovat, nejspíš web zkrátka rychle opustí. Problémy mohou nastat i s podporou *media queries* v různých internetových prohlížečích.

Dobře responzivně optimalizované webové stránky přenášejí méně dat než běžné statické webové stránky. Ovšem přenášejí zpravidla více dat než separátní verze webu, které jsou optimalizované přímo pro konkrétní zařízení (Forgáč, 2013). Responzivní stránky mohou přenášet poměrně velké množství dat, hlavně pokud jde o obrázky. S adaptibilními obrázky jsou problémy, ale existuje již několik metod jejich řešení (Držka a Michálek, 2013). Musíme pamatovat na skutečnost, že všem zařízením odesíláme stejný HTML kód. Jen pomocí kaskádových stylů ovlivňujeme, jak budou stránky vypadat na různých zobrazovacích zařízeních.

Proto je důležité budovat webové stránky pomocí metody postupného vylepšování, která je zmíněna v následující podkapitole a dodržovat zásadu *Mobile First*. Složitější prvky, jako jsou různé animace, barevné přechody, obrázky na pozadí či obrázkové slideshow, je lepší nechat zobrazovat až na obrazovkách s větším rozlišením (Castledine et al., 2013).

U tohoto přístupu zcela odpadá problém s detekčním mechanismem. *Media queries* totiž v žádném případě neslouží k detekci zařízení (Michálek, 2013c). Zobrazovacích zařízení je na dnešním trhu nespočetně a nemůžeme počítat s tím, že jednoduše pomocí dotazů na média dokážeme detekovat každé z nich. O tom responzivní design není. Mnoho tvůrců webu se snaží o vytvoření samostatných stylů pro chytré telefony, tablety a desktopové počítače, ale ve skutečnosti to bez nějakého dalšího detekčního mechanismu nefunguje. Porovnání přístupů k tvorbě webových stránek vhodných pro různá zobrazovací zařízení (Tab. 2.1).

Tab. 2.1: Porovnání přístupů k tvorbě webových stránek vhodných pro různá zobrazovací zařízení (zdroj: autor; Forgáč, 2013)

Kritéria	Separátní verze webových stránek	Responzivní design (RWD)	Porovnání přístupů
Rychlost načítání dat na mobilních zařízeních	Rychlá	Rychlejší než u statického webu	Separátní verze
Aktualizace webových stránek a údržba	Náročná a nákladná	Méně náročná – jen jeden HTML kód	RWD
Možnost přepnout na plnohodnotnou verzi	Ano	Ne	Separátní verze
Možnost přizpůsobení obsahu uživatelům mobilních zařízení	Ano	Částečně	Separátní verze
Přizpůsobování různým šířkám obrazovky	Ne	Ano	RWD
Nutnost detekčního mechanismu	Ano	Ne	RWD
Přizpůsobování se všem zařízením (i doposud neexistujícím)	Ne	Ano	RWD
Cena	Vysoké náklady	Nižší náklady než u separátní verze webu	RWD

Z předchozího textu a tabulky vítězně vychází responzivní přístup k tvorbě webových stránek. Pokud tedy povaha projektu umožňuje zvolení této varianty, je vhodnější vydat se touto cestou. Responzivní design je budoucnost. Navíc internetové prohlížeče v chytrých telefonech jsou stále výkonnější, stejně tak samotné přístroje.

Přirozeně se objevují i negativní ohlasy na responzivní design. Především, že je pro uživatele mnohdy matoucí a není možné z něj přepnout na plnou verzi webových stránek (Vyleťal, 2013). Dále se responzivní design často neguje z důvodů datové velikosti a tedy pomalého načítání stránek (zvláště ve starších zařízeních). Pokud se však při návrhu a samotné tvorbě responzivního webu postupuje správně, tyto problémy s výkonností mizí (Castledine et al., 2013).

2.4 Zásady tvorby webových stránek optimalizovaných pro mobilní zařízení

V posledních letech se celkově mění pohled na přístup k tvorbě webových stránek. Většinou jsou primárně zvažovány potřeby uživatelů mobilních telefonů, protože jejich počet neustále roste (Melich, 2014). Separátní verze webových stránek jsou vytvářeny především pro mobilní telefony a responzivně optimalizované stránky by se měly přizpůsobovat všem zařízením – tedy i mobilním telefonům. Obě varianty se tedy snaží o optimalizaci webových stránek pro mobilní zařízení. Při vytváření webových stránek optimalizovaných pro mobilní zařízení je dobré dodržovat několik zásad, které jsou popsány níže v této podkapitole.

2.4.1 Metoda postupného vylepšování (progressive enhancement)

Jak uvádí (Castledine et al., 2013), metoda postupného vylepšování spočívá v tom, že nejprve ustanovíme základní funkce, které fungují ve všech prohlížečích. Poté teprve aplikujeme styly pomocí jazyka CSS a další funkce (například funkce kódu jazyka JavaScript) pro prohlížeče, které je podporují. Metoda postupného vylepšování stojí na základě testování funkce prohlížeče. Testuje se, zda prohlížeč danou funkcionalitu podporuje, pokud testem podpory neprojde, je mu odeslána základní šablona. Webové stránky se nemusí chovat a vypadat ve všech prohlížečích stejně, ale jejich obsah by měl být přístupný všem. Negativum této metody spočívá ve vytvoření všech potřebných testů.

Modernizr

K testování nám může posloužit knihovna JavaScriptu Modernizr. Tato knihovna dokáže detekovat, zda prohlížeč podporuje určité funkce HTML5, CSS3 a některé další funkce. Knihovnu Modernizr je možné stáhnout z oficiálního webu modernizr.com ve dvou verzích. Buď si můžeme stáhnout plnou verzi knihovny, nebo customizovanou verzi. To je verze, ve které si můžeme sami zvolit funkce a prvky CSS3 a HTML5 (a některé další funkce), které potřebujeme testovat. Aktivace této knihovny je velmi jednoduchá. Stačí vložit skript s odkazem na knihovnu Modernizr (Výpis 5) do elementu `<head>` a vložit `class="no-js"` do elementu `<html>` (Výpis 6).

Výpis 5: Vložení knihovny Modernizr do elementu `<head>`.

```
<skript type="text/javascript" src="js/modernizr.custom.js"></skript>
```

Výpis 6: Vložení třídy `no-js` do elementu `<html>`.

```
<html class="no-js">
```

Přidání třídy `no-js` je nezbytné. V případě, že internetový prohlížeč podporuje JavaScript, Modernizr tuto třídu automaticky změní na třídu `js`. Pomocí knihovny Modernizr tak můžeme cílit i na zařízení nepodporující JavaScript. V případě, že prohlížeč JavaScript podporuje, přidá Modernizr do elementu `<html>` třídy pro všechny testy, které podporuje. Element `<html>` s třídami generovanými Modernizrem se liší prohlížeč od prohlížeče v závislosti na jeho možnostech. Pokud například prohlížeč nepodporuje zaoblené hrany (vlastnost CSS `border-radius`), vygeneruje Modernizr třídu `borderradius` s předponou `no-`. Třída `<html>` pak bude mít tento tvar: `no-borderradius`. Tyto třídy můžeme jednoduše používat v CSS šablonách (Castledine et al., 2013).

2.4.2 Mobile first v CSS

Ve své podstatě můžeme metodu *Mobile First* zařadit do metody postupného vylepšování. Tento přístup je v širším kontextu dobře popsán ve stejnojmenné knize od Luka Wroblewského (Wroblewski, 2011). Podle *Mobile First* nejprve do základního CSS souboru píšeme pravidla pro mobilní zařízení a teprve později definujeme pomocí dotazů na média styly pro desktopové počítače (Michálek, 2013b).

2.4.3 Snížení počtu requestů na server a datová velikost

Při vývoji webových stránek pro mobilní zařízení se počítá každý kilobajt, proto je dobré vkládat minimalizované soubory typu CSS nebo JS. Můžeme například využít i zabalení těchto souborů pomocí programu Gzip (Castledine et al., 2013). Obrázky bychom měli optimalizovat, co nejvíce to jde, a zbytečně neodesílat velké obrázky mobilním zařízením.

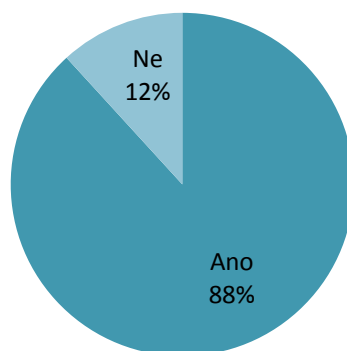
Dále bychom se měli snažit snížit počet requestů na server. Toho můžeme dosáhnout například sloučením CSS souborů do jednoho a to samé udělat i se soubory typu JS. Můžeme také využít tzv. CSS sprity. Jde o sloučení ikon do jednoho jediného obrázku. CSS sprity používá například Facebook (Michálek a Sládek, 2013) Je dobré vyhýbat se složitým selektorům v CSS (Melich, 2014).

2.5 Dotazníkové šetření

K podpoření teoretické části této práce bylo provedeno dotazníkové šetření. Dotazník byl vytvořen pomocí služby Google Docs. Všechny grafy jsou zpracovány v programu Microsoft Excel. Otázky k dotazníku jsou v příloze A. Cílovou skupinu tvořili lidé ve věku od 15 do 30 let. Účelem dotazníku bylo v první řadě zjistit, jaká jiná zařízení kromě desktopových počítačů a notebooků lidé k prohlížení webových stránek používají a kolik lidí taková zařízení využívá – tedy jestli má téma flexibilního designu webových stránek smysl. Dále měl dotazník zjistit, které internetové prohlížeče ve svých zařízeních lidé preferují a zda jsou s nimi spokojeni.

Na dotazník odpovědělo celkem 291 lidí, po vyřazení nekorektních odpovědí zbylo 263 plnohodnotných odpovědí od lidí cílové věkové skupiny. Na dotazník odpověděl téměř stejný počet mužů i žen (132 žen a 131 mužů). Většina odpovědí pochází od studentů vysokých škol – 218 lidí (83 %), druhé nejvyšší procento patří zaměstnancům na plný úvazek – 22 lidí (8 %). Následující graf ukazuje, že drtivá většina lidí vlastní některé z běžných zobrazovacích zařízení, jako je tablet, smartphone či netbook/ mini notebook (Obr. 2.1).

Vlastníte některé z těchto zařízení - Tablet, Smartphone, Netbook/ Mini notebook)?



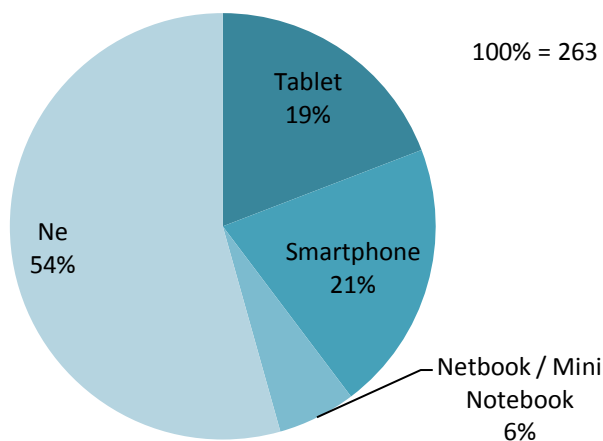
100% = 263 odpovědí

Obr. 2.1:

Grafické znázornění odpovědi na otázku „Vlastníte některé z těchto zařízení – Tablet, Smartphone, Netbook/ Mini notebook?“ (zdroj: autor)

Jak tedy můžeme vyčíst z grafu na Obr. 2.1, mnoho lidí využívá k prohlížení internetu tablety, smartphony a netbooky/ mini notebooky. Tato zařízení mají menší obrazovky, nižší výkon než desktopové počítače. Klasické statické webové stránky se na nich nemusí zobrazit správně nebo dostatečně rychle. Statické stránky prohlížené v telefonu, bývají pro uživatele velmi nepraktické. Hlavně pokud jde o prvky navigace, které mohou být v telefonu příliš malé pro stlačení. Dále dotazník zjišťoval (Obr. 2.2), zda si lidé chtějí zhruba do jednoho roku pořídit tablet, smartphone či netbook/ mini notebook.

Chcete si v budoucnu (zhruba do jednoho roku) pořídit některé z těchto zařízení?



100% = 263 odpovědí

Obr. 2.2

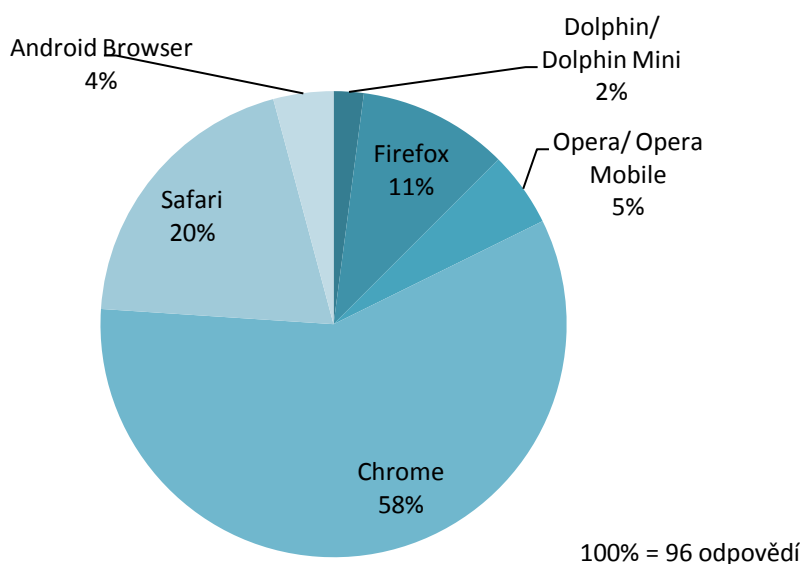
Grafické znázornění odpovědi na otázku „Chcete si v budoucnu (zhruba do jednoho roku) pořídit některé z těchto zařízení?“ (zdroj: autor)

Z grafu (Obr. 2.2) vyplývá, že zhruba polovina z odpovídajících si do jednoho roku chce pořídit buď tablet, smartphone nebo netbook/ mini notebook. Celkem 31 lidí (12 % viz Obr. 2.1) odpovědělo, že nevlastní žádné podobné zařízení a 15 z nich si ani žádné takové zařízení pořídit nechce. Celkově to však znamená, že v budoucnu bude stále méně lidí, kteří nevlastní žádný tablet, smartphone nebo netbook/ mini notebook. Význam vytváření responzivních webů tak stále poroste.

Z průzkumů vyplývá, že většina lidí nemá tušení, co si představit pod pojmem responzivní design. Pouze 27 % lidí (71 lidí), odpovědělo, že ví, co responzivní design znamená. Těmto lidem byla položena další otázka: „*Máte pocit, že jsou responzivní webové stránky pro uživatele matoucí?*“. 11 lidí ze 71 odpovědělo, že ano. Jeden člověk odpověděl: „Jak kdy.“. Při dodržování zásady *Mobile First* by tomu tak být nemělo.

96 lidí z celkového počtu 263 odpovědělo, že vlastní tablet. Někteří i více než jeden. Kromě dotázání se na konkrétní typ tabletu byla vlastníkům tabletů položena otázka, jaký internetový prohlížeč ve svém zařízení preferují, viz graf na Obr. 2.3.

Jaký internetový prohlížeč preferujete ve Vašem tabletu?

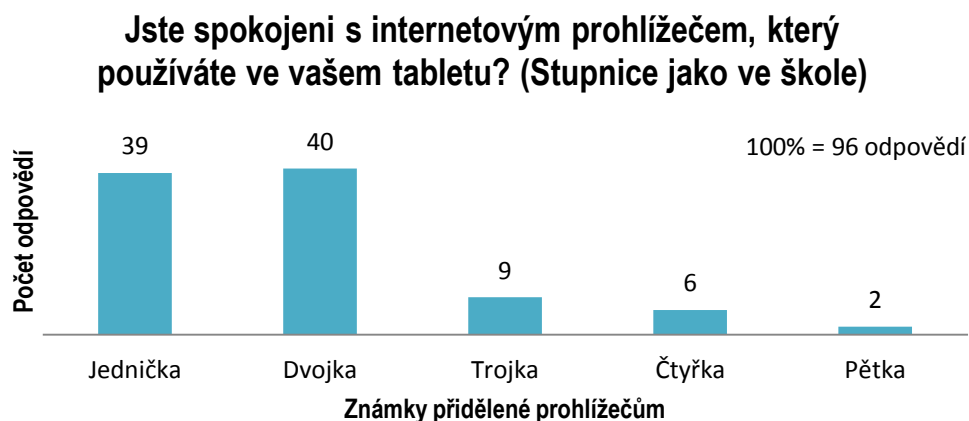


Obr. 2.3:

Graf znázorňující odpověď na otázku: „*Jaký internetový prohlížeč preferujete ve Vašem tabletu?*“ (zdroj: autor)

Z grafu na Obr. 2.3 můžeme zjistit, že většina uživatelů tabletů využívá k prohlížení webových stránek internetový prohlížeč Google Chrome.

Jde povětšinou o vlastníky zařízení s Androidem, ale našlo se i pár vlastníků iPadů (řada tabletů od společnosti Apple), kteří místo Safari (prohlížeč vyvinutý společností Apple) označili za preferovaný prohlížeč Google Chrome. Další otázka byla mířená na preferovaný prohlížeč a týkala se spokojenosti uživatelů s daným prohlížečem. Jde tedy spíše o vlastní pocity uživatelů ohledně jejich oblíbených prohlížečů. Prohlížečům byly udělovány známky jako ve škole na stupnici od 1 do 5 (Obr. 2.4).



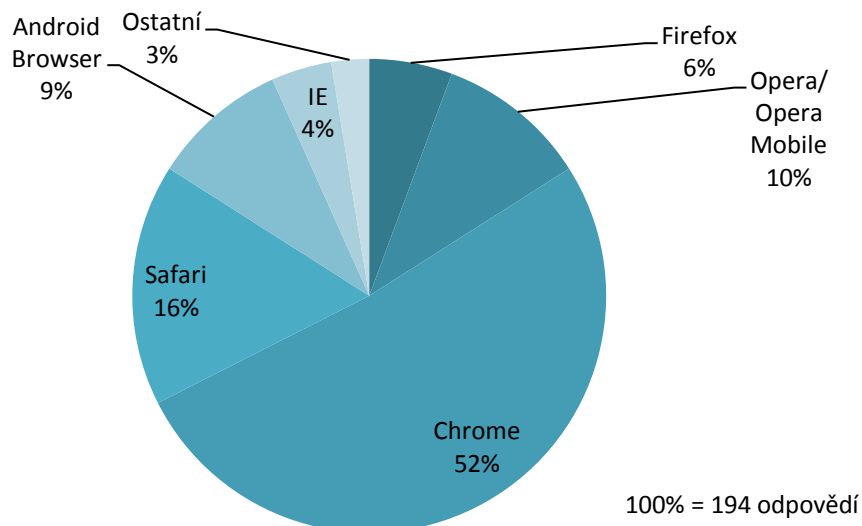
Obr. 2.4:

Graf znázorňující odpovědi na otázku: „Jste spokojeni s internetovým prohlížečem, který používáte ve vašem tabletu? (Stupnice jako ve škole)“ (zdroj: autor)

Jak si můžeme povšimnout, jsou uživatelé se svými oblíbenými prohlížeči povětšinou spokojeni. Nejčastěji udělili prohlížeči známku 2. Nejlepší průměrnou známku získal prohlížeč Safari – 1,7. Na druhém místě se umístil prohlížeč Google Chrome se známkou 1,8. A na třetí pozici skončil prohlížeč Firefox s průměrnou známkou 1,9. U ostatních prohlížečů není dostatek odpovědí pro srovnání s předchozími třemi prohlížeči. Mezi nejčastěji uváděná zařízení v kategorii tabletů byly Samsung Galaxy Tab (různé verze z této řady tabletů); Samsung Galaxy Note (různé verze z této řady tabletů); a iPady Mini, Air, 2, 3 a 4.

Celkem 194 lidí, což představuje 74 % z 263, vlastní smartphone, někteří i více než jeden. Jaké internetové prohlížeče vlastníci chytrých telefonů preferují, je znázorněno na následujícím obrázku (Obr. 2.5).

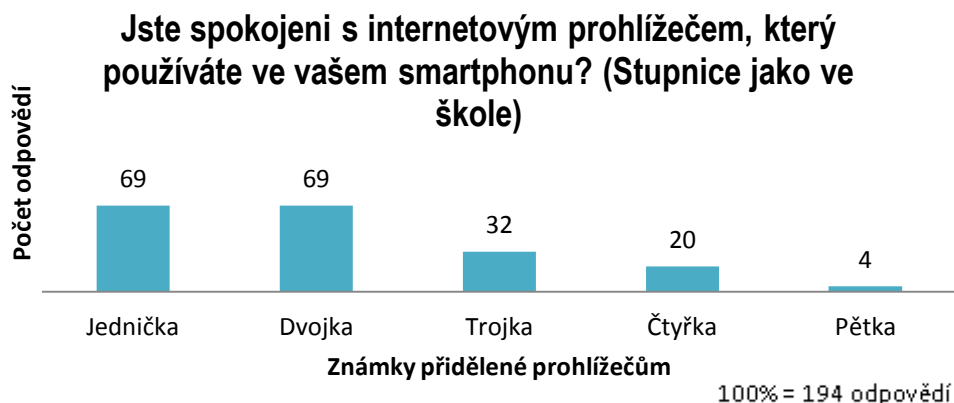
Jaký internetový prohlížeč preferujete ve Vašem smartphonu?



Obr. 2.5:

Grafické znázornění odpovědí na otázku: „*Jaký internetový prohlížeč preferujete ve Vašem smartphonu?*“ (zdroj: autor)

Jak můžeme vidět na grafu, také uživatelé chytrých telefonů preferují nejvíce prohlížeč Google Chrome. Oproti tabletům je zde vidět poměrně velký nárůst oblíbenosti Opery a Android Browseru. Android Browser je defaultním prohlížečem v zařízeních s operačním systémem Android. Naopak oblíbenost Firefoxu u uživatelů mobilních telefonů klesá. V následujícím grafu je znázorněno uživatelské známkování internetových prohlížečů (Obr. 2.6).



Obr. 2.6:

Grafické znázornění odpovědí na otázku: „*Jste spokojeni s internetovým prohlížečem, který používáte ve vašem smartphonu? (Stupnice jako ve škole)*“ (zdroj: autor)

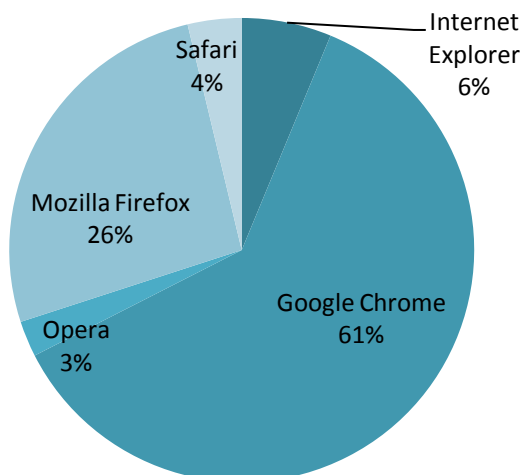
Graf ukazuje mírný nárůst nespokojenosti s prohlížeči u uživatelů mobilních telefonů oproti uživatelům tabletů. Bereme-li v úvahu menší obrazovky telefonů a slabší výkon, nejde o příliš překvapivý výsledek. Zvláště ve starších modelech chytrých telefonů se mohou statické weby zobrazovat velmi špatně a pomalu. Nejlepší průměrnou známku 1,9 obdržel prohlížeč Google Chrome. Druhou nejlepší průměrnou známku 2,0 získal prohlížeč Safari. Na třetím místě se umístili prohlížeče Firefox a Android Browser s průměrnou známkou 2,2. Internet Explorer obdržel průměrnou známku 2,3 a na posledním místě se umístila Opera s průměrnou známkou 2,6.

Vlastníkům chytrých telefonů byla udělena otázka navíc: „*Kde nejčastěji používáte svůj smartphone k prohlížení webových stránek?*“. Na výběr bylo ze 4 možností – doma; v práci/ ve škole; na cestách a jinde. Nejčastější odpovědí bylo, že doma (101 odpovědí - 52 %). Z toho se dá usuzovat, že mobilní verze webů nejsou pro mnohé z návštěvníků příliš užitečné. Doma ocení spíše plnou verzi webu se všemi informacemi než osekanou verzi webových stránek.

Nejčastěji uváděným chytrým telefonem se v průzkumu stal Samsung Galaxy ve verzi SIII, velké zastoupení měla také verze SIII Mini a SII. Velký podíl tvoří opět produkty značky Apple a jejich iPhony – hlavně verze iPhone 4 a iPhone 5. Početné zastoupení tvoří také telefony značky HTC převážně ze série One a Desire. Pověštinou se v průzkumu objevovala modernější zařízení, našlo se ale také několik zastaralých modelů jako je například Nokia E52.

80 lidí z 263 odpovědělo, že vlastní netbook či mini notebook. Tak jako u tabletů a chytrých telefonů byli lidé tázáni, jaký typ netbooku/ mini notebooku vlastní. Dle průzkumu vlastní netbook či mini notebook většinou lidé, kteří nevlastní tablet. Jaké internetové prohlížeče ve svých zařízeních lidé preferují, můžeme vidět na následujícím grafu (Obr. 2.7).

Jaký internetový prohlížeč preferujete ve Vašem netbooku/ mini notebooku?



100% = 80 odpovědí

Obr. 2.7:

Grafické znázornění odpovědí na otázku: „*Jaký internetový prohlížeč preferujete ve Vašem netbooku/ mini notebooku?*“ (zdroj: autor)

Také u uživatelů netbooků a mini notebooků vítězí v oblíbenosti internetový prohlížeč Google Chrome. Na druhém místě se umístil prohlížeč Mozilla Firefox, který má v této kategorii nejvyšší počet uživatelů. Je to přirozený úbytek na úkor prohlížeče Safari, které používají většinou uživatelé produktů společnosti Apple. Důvodem je nízký počet zařízení od Applu v této kategorii. Operu na zařízeních typu netbook/ mini notebook preferuje oproti chytrým telefonům a tabletům nejméně lidí. Známkování prohlížečů uživateli v kategorii netbook/ mini notebook je znázorněno na grafu (Obr. 2.8).

Jste spokojeni s internetovým prohlížečem, který používáte ve vašem netbooku/ mini notebooku? (Stupnice jako ve škole)



100% = 80 odpovědí

Obr. 2.8:

Grafické znázornění odpovědí na otázku: „*Jste spokojeni s internetovým prohlížečem, který používáte ve vašem netbooku/ mini notebooku? (Stupnice jako ve škole)*“ (zdroj: autor)

Z grafu můžeme vyčíst, že spokojenost uživatelů s internetovými prohlížeči v zařízeních typu netbook/ mini notebook roste. Většina lidí je se svým prohlížečem velmi spokojena. To je přirozeně způsobeno tím, že netbooky a mini notebooky mívají vyšší výkon než tablety nebo chytré telefony a rozlišení jejich obrazovek bývá podobné jako u desktopových počítačů. Nejlepší průměrnou známku 1,5 získal prohlížeč Mozilla Firefox. Hned za ním se s průměrnou známkou 1,6 umístil prohlížeč Google Chrome. Internet Explorer získal průměrnou známku 2,0. Ostatní internetové prohlížeče nezískaly dostatečné množství odpovědí pro jejich ohodnocení.

Vysoké zastoupení v uvedených typech zařízení z kategorie netbook/ mini notebook mají zařízení z řady netbooků ASUS Eee PC. Dále hodně lidí uvedlo, že vlastní netbooky od společnosti Acer – převážně z řady produktů Acer Aspire One. Poměrně hodně lidí uvedlo, že vlastní netbook od společnosti HP – řada produktů HP Mini.

Dotazník měl za úkol také zjistit, jestli lidé používají i jiná než běžná zobrazovací zařízení k prohlížení webových stránek. Dle průzkumu největší zastoupení mezi těmito neobvyklými zařízeními tvoří elektronické čtečky knih. Většina lidí vlastní některou z řady čteček Kindle od společnosti Amazon.com. Mezi další neobvyklá zařízení lidé uváděli různé smart TV a herní konzole. Jeden z dotázaných odpověděl, že vlastní smart hodinky značky Samsung.

2.5.1 Zhodnocení dotazníkového šetření

Nejdůležitějším zjištěním je, že naprostá většina lidí ve věkové kategorii 15 až 30 let (povětšinou studentů vysokých škol) vlastní zařízení typu tablet, smartphone, netbook/ mini notebook a více než polovina z nich si chce přibližně do jednoho roku koupit další zařízení tohoto druhu. Průzkum tedy říká, že uživatelů podobných zařízení bude neustále přibývat, tudíž i mobilních přístupů k internetu. Dotazníkové šetření splnilo svůj hlavní účel. Dokázalo, že má téma flexibilního designu webových stránek čím dál tím větší význam.

Škála zařízení, která se v průzkumu objevila, je velmi rozmanitá. Vytvoření separátních verzí webových stránek, které by se přizpůsobovaly všem těmto zařízením, by bylo poměrně komplikované. Ovšem ani tvorba responzivního webu by nebyla snadná, pokud bychom se snažili pomocí *media queries* přizpůsobit naše CSS šablony pro jednotlivé kategorie těchto zařízení (např. CSS šablona pro tablety, smartphony, desktopy).

Poměrně užitečné je znát odpověď na otázku, kde lidé svůj smartphone nejčastěji používají. Nejčastější odpovědí bylo *doma*. V prostředí domova lidé asi zrovna moc neocení osekanou mobilní verzi webových stránek a raději přepnou na plnohodnotnou verzi.

Dle průzkumu v oblíbenosti zvítězil internetový prohlížeč Google Chrome a to ve všech kategoriích - smartphone; tablet; netbook/ mini notebook. Nezanedbatelné procento v oblíbenosti v kategoriích smartphone a tablet získalo Safari. Poměrně hodně lidí používá na mobilních telefonech s Androidem defaultní prohlížeč Android Browser a najde se i pár uživatelů Opery (V ostatních kategoriích bylo uživatelů Opery zanedbatelné množství). Na netboocích a mini noteboocích je situace jiná a velké popularity v nich dosahuje prohlížeč Mozilla Firefox. Internet Explorer nepoužívá téměř nikdo. Jen pár uživatelů telefonů Nokia s Windows Pohone. Z těchto zjištění vyplývá, že má velký smysl vyladit stránky hlavně pro Google Chrome. Zdá se, že jeho oblíbenost stále roste. U mobilních telefonů je určitě dobré testovat stránky i v prohlížečích Opera a v prostředí iOS - podíl uživatelů iPhonů je poměrně velký.

3 Obecný postup tvorby responzivně optimalizované webové stránky

V zásadě lze postup tvorby responzivních webových stránek shrnout jen do několika málo jednoduchých kroků. Nebudeme zde uvažovat použití některého z frontend frameworků a CSS preprocesorů. O třech základech, na kterých stojí základy klasického responzivního designu, se můžete dočíst v předchozí kapitole.

3.1 Vytvoření kompletního obsahu webových stránek

Prvním krokem je sestavení kompletního obsahu webu - veškerých textů i obrázků. Při návrhu obyčejné statické webové prezentace to není nutné. Někdy stačí vytvořit jen stránky se zkušebním textem. Ale u responzivních stránek bychom později mohli zjistit, že námi navržený design není možné použít kvůli obsahu. Již na počátku musíme promyslet, které prvky se budou flexibilně měnit a jakým způsobem (Castro a Hyslop, 2012).

3.2 Návrh drátového modelu (wireframe)

Poté co máme připravený obsah webových stránek, můžeme navrhnout drátový model webu. Nejprve navrhne model pro nejmenší obrazovky – povětšinou tedy chytré mobilní telefony. Drátový model, neboli wireframe, je vizuální reprezentace rozvržení webové stránky či webové aplikace. Jde vlastně o náčrt projektu, který vytváříme předtím, než se pustíme do jeho vlastní realizace. Drátový model nám umožňuje udělat si lepší představu o vzhledu a celkové struktuře webu či webové aplikace. Wireframe neobsahuje obrázky ani barevný koncept projektu. Můžeme ho využít k reprezentaci výchozího rozložení obrazovek a navigačního modelu webu. Drátové modely se dají vytvářet snadněji a mnohem rychleji než plnohodnotné grafické návrhy a makety projektu. Slouží tedy hlavně k tomu, abychom si mohli vyzkoušet třeba hned několik možných přístupů k řešení daného projektu. Drátové modely můžeme tvořit například i jen za pomoci tužky a papíru. Existuje však i celá řada nástrojů, pomocí nichž můžeme drátový model vytvořit (Castledine et al., 2013).

3.3 Návrh grafiky

Po vytvoření drátových modelů teprve navrhujeme grafiku webových stránek. V této fázi vybíráme celkový barevný koncept, řešíme typografii, vybíráme obrázky na pozadí, ikony, řešíme vzhled tlačítek, formulářů atd. U responzivního designu je vhodné vybírat ikony, které se chovají jako fonty a do CSS se vkládají pomocí direktivy `@font-face`. Složitější grafické prvky navrhujeme až pro větší obrazovky.

3.4 Tvorba HTML kódu

Vytvářet responzivní stránky můžeme i bez používání jazyka HTML5. Existuje však několik důvodů, proč používat právě tuto verzi HTML. HTML5 myslí i na potřeby jiných než desktopových počítačů. Nabízí několik rozšíření, která ztenčují kód a zrychlují tak načítání stránek. Například při používání některých nových formulářových polí již není nutné používat JavaScript (Juneja, 2012).

3.4.1 HTML

Jde o jednoduchý značkovací jazyk, který podporují téměř všechna zařízení. Vznikl na počátku 90. let 20. století. Zpočátku šlo o stručný dokument, který obsahoval popis několika elementů k tvorbě webových stránek. Spolu s rostoucím číslem verze jazyka HTML se tento jazyk neustále rozrůstal o nové elementy a přizpůsoboval se novým požadavkům. Současná verze je HTML5, které je nástupcem HTML 4.01. Jazyk HTML se používá k definování významu obsahu stránek. Lze pomocí něj obsah i stylovat, ale mnohem vhodnější je k tomuto účelu využívat kaskádové styly.

Vývoj specifikace jazyka HTML5 začal v roce 2004 pod hlavičkou WHATWG (Web Hypertext Application Technology Working Group) mimo W3C (World Wide Web Consortium). W3C tuto specifikaci přijalo až v roce 2007. HTML5 zdědilo spoustu vlastností od svých předcházejících verzí, díky čemuž je zaručena i zpětná kompatibilita ve starších webových prohlížečích. Specifikace jazyka HTML5 se neustále mění, jelikož ještě nejsou u HTML5 prohlášena za finální. Zápis DOCTYPE pro dokumenty typu HTML5 je opravdu jednoduchý: `<!DOCTYPE html>`. HTML5 přináší celou řadu nových funkcí. Například některé nové doplňkové elementy pro popis obsahu:

- `article` (článek)

- section (část obsahu, jejíž odstranění by narušilo logiku dokumentu)
- figure (obrázek)
- aside (část obsahu, jejíž odstranění by nenarušilo logiku dokumentu – např. pro postranní panely atp.)
- nav (navigace)
- video a audio
- a další...

Každá část HTML dokumentu, která je specifikována pomocí elementu `article`, `section`, `aside` či `nav`, může mít svou vlastní osnovu od nadpisu `h1` až po nadpis `h6`. Jazyk HTML5 nyní nově podporuje také přehrávání videí a zvukových souborů přímo z webových prohlížečů bez nutnosti instalace dalších doplňků. Mezi další novinky v HTML5 patří řada funkcí, které usnadňují tvorbu formulářů. HTML5 obsahuje nové formulářové elementy, atributy, typy vstupních polí, validační funkce a možnosti změny vzhledu formuláře. Mezi nové formulářové elementy patří `<output>`. Tento element umožňuje zobrazení výpočtů, které jsou založeny na hodnotách z jiných formulářových polí (Castro a Hyslop, 2012; Sládek, 2010).

3.4.2 Zobrazovaná oblast

Při tvorbě responzivního webu nesmíme zapomenout doplnit do HTML kódu element `<meta>` se jménem `viewport`, který se vkládá do elementu `<head>`. Tento element řeší šířkovou nekompatibilitu mezi hodnotami `width` a `device-width` u mobilních přístrojů. Pro pochopení nutnosti vložit tento element do HTML kódu, je nezbytné vysvětlit si pojem zobrazovaná oblast.

Zobrazovaná oblast je viditelný prostor uvnitř okna prohlížeče. V dotazech na médium odpovídá vlastnost `width` šířce zobrazované plochy. Tato šířka je odlišná od vlastnosti `device-width`, která označuje šířku obrazovky. U desktopových počítačů se tyto hodnoty rovnají, ale u mobilních telefonů bývají hodnoty těchto šířek rozdílné. Okno prohlížeče může být větší než obrazovka mobilního telefonu. Například webový prohlížeč mobile Safari má výchozí šířku 980 pixelů, Opera Mini 850 pixelů, Android Web Kit 800 pixelů a Internet Explorer 974 pixelů. Zatímco obrazovky chytrých telefonů mají něco okolo 320 pixelů na šířku.

Výchozí šířky mobilních prohlížečů jsou takové proto, aby se na nich zobrazily celé i webové stránky, které jsou určeny pro desktopové počítače (Castledine et al., 2013; Castro a Hyslop, 2012).

V následujícím výpisu kódu je příklad použití značky `<meta>` s atributy `name` a `content` (Výpis 7).

Výpis 7: Ukázka elementu `<meta>` se jménem `viewport` (Castro a Hyslop, 2012).

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

Důležitým výrazem je `width="device-width"`, který umožní, aby prohlížeč přizpůsobil šířku zobrazované plochy šířce obrazovky. Výraz `initial-scale` nastavuje výchozí úroveň přiblížení, hodnota 1.0 znamená 100% přiblížení.

Meta klíč `viewport` nabízí také další vlastnosti (Castledine et al., 2013):

- **Maximum-scale** – Atribut, který určuje největší možné měřítko zobrazované oblasti, které může být uživatelem nastaveno.
- **Minimum-scale** – Atribut, který určuje nejmenší možné měřítko zobrazované oblasti, které může být uživatelem nastaveno.
- **User-scalable** – Hodnotou tohoto atributu lze nastavit, jestli uživatel bude moci sám měnit měřítko zobrazované oblasti.

3.4.3 Media queries a prvky HTML5 pro straší verze Internet Explorer

8. dubna 2014 skončila podpora systému Windows XP (Microsoft, 2014). Vzhledem k této skutečnosti se dá předpokládat, že uživatelů Internet Explorer 8 a nižších bude neustále klesat. Prozatím se však ještě s uživateli Internet Explorer 8 (případně nižších verzí) setkat můžeme.

Aby rozuměly dotazům ne média i starší prohlížeče Internet Explorer, vložíme do kódu skript *respond.js*. Skript *respond.js* vytvořený Scottem Jehlem, je dostupný na adrese: <https://github.com/scottjehl/Respond> pod licencí MIT. Další možností je použít skript *media-queries.js*. Tento skript je dostupný na adrese: <https://code.google.com/p/css3-mediaqueries-js/> pod licencí MIT. Výpis 8 názorně ukazuje, jak můžeme přidat skript *meda-queries.js* do HTML souboru (La, 2011).

Výpis 8: Vložení skriptu *media-queries.js* pro podporu *media queries* ve starších verzích prohlížeče Internet Explorer.

```
<!--[if lt IE 9]>  
<script src="http://css3-mediaqueries-js.googlecode.com/svn/trunk/css3-  
mediaqueries.js"></script>  
<![endif]-->
```

Abychom zajistili funkčnost elementů HTML5 i ve starších verzích Internet Exploreru můžeme použít skript *html5shiv.js*. Skript *html5 Shiv* je dostupný na adrese: <http://html5shiv.googlecode.com/svn/trunk/html5.js> pod licencí MIT/GPL2. Výpis 9 ukazuje, jak vložit tento skript do HTML dokumentu (Coyier, 2009).

Výpis 9: Vložení skriptu *html5shiv.js* pro podporu elementů HTML5 ve starších verzích prohlížeče Internet Explorer.

```
<!--[if lt IE 9]>  
<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>  
<![endif]-->
```

3.5 Kaskádové styly

CSS3 je esenciální součástí responzivního designu. Bez nových dotazů na médium, které přináší CSS3, bychom se při tvorbě responzivních stránek neobešli.

3.5.1 CSS

Jazyk CSS (cascading style sheets) je vhodný k úpravě vzhledu obsahu dokumentu, který je tvořen jazykem HTML. Oficiálně byl jazyk CSS přijat až v roce 1996. Současná verze je CSS3, která stejně jako HTML5, zdělila mnoho vlastností po svých předchůdcích. CSS3 zavádí mnoho nových vizuálních efektů – nejrůznější animace, přechody, zaoblené rohy, nové možnosti v zápisu barev (formáty RGBA, HSLA, HSL), průhlednost prvků a mnoho dalších nových vizuálních prvků. CSS3 také zavádí nové selektory, které umožňují aplikovat styly více způsoby, než tomu bylo možné v dřívějších verzích. Důležitou novinkou pro tvorbu responzivního webdesignu jsou dotazy na médium (*media queries*).

3.5.2 Základní kaskádové styly

Když máme připravený grafický návrh webových stránek, můžeme se pustit do kódování CSS. Na začátek CSS dokumentu je dobré vložit reset stylů pro starší prohlížeče. Jde o pravidla stylů, která by měla snížit nesrovnalosti mezi prohlížeči, jako jsou hodnoty margin, line-height, velikosti písem atd. Meyerův reset stylů je dostupný na internetové adrese: <http://meyerweb.com/eric/tools/css/reset/> - licence: žádná.

Nejprve si vytvoříme základní kaskádové styly, které umožní zobrazení stránky ve všech prohlížečích v jednom sloupci. V základních CSS stylech můžeme formátovat písmo, barvy a například nastavovat hodnoty padding a margin některým obalujícím prvkům. Neměli bychom zarovnávat plovoucí elementy nebo umisťovat elementy na stránku například pomocí absolutního nebo relativního pozicování. Tím bude zajištěno, že obsah stránek bude dostupný ve všech webových prohlížečích. V jednotlivých zařízeních se webové stránky mohou zobrazovat s různými drobnými odchylkami, ale to je v pořádku. V případě, že potřebujeme předělat již hotové stránky na responzivní, postupujeme opačně. Máme již navrhnuté a vytvořené CSS styly pro desktopové počítače a postupně doplňujeme kaskádové styly pro menší zařízení. Tento postup se však neřídí technikou postupného vylepšování (Castro a Hyslop, 2012).

3.5.3 Kaskádové styly pro různé velikosti obrazovky a breakpointy

Po nakódování základních kaskádových stylů postupně přidáváme další pravidla CSS pomocí dotazů na média. Postupujeme od nejmenších obrazovek k těm největším. Teď se dostáváme k problému *breakpointů*. *Breakpoint* je označení pro místo v kódu CSS, kde se tážeme prostřednictvím dotazů na médium na velikost zobrazované plochy nebo obrazovky zařízení. Ve chvíli, kdy velikost okna prohlížeče překročí námi vytyčený *breakpoint*, uplatní se pravidla CSS přiřazené k tomuto *breakpointu* (Jackson, 2012). Někteří vývojáři se snaží používat *media queries* k detekci zařízení. To však není nejlepší postup. Zaprvé je to díky široké škále zařízení téměř nemožné a zadruhé to jde proti samotné myšlence responzivního designu, tj. že se mají stránky přizpůsobovat všem zařízením. Stránky bychom jednoduše měli navrhnout co nejflexibilnější, aby se přijatelně zobrazily v každém zařízení. To znamená, že musíme zajistit, aby se některé prvky na stránce nepřekrývaly, byly dostatečně velké a čitelné, nebo aby uživatel nemusel nevhodně posouvat obsah.

Nejlepší je zkrátka zkoušet měnit velikost okna prohlížeče a ve chvíli, kdy se některé prvky překrývají, vytvořit *breakpoint* (Ralston, 2013). Také není vhodné uživatelům tabletů vnucovat jiný vzhled stránek, než na desktopových počítačích. Na běžně velkých tabletech mohou být webové stránky bez problémů zobrazovány stejně na desktopech. Jiný vzhled by stál za úvahu možná při držení tabletu na výšku (Michálek, 2013c).

3.5.4 CSS Frontend framework

K vytváření responzivních layoutů nám může pomoci některý z CSS frontend frameworků. CSS frameworky jsou ve své podstatě spíše knihovny, které slouží hlavně k resetu výchozích stylů prohlížečů a nastavení typografie. Za frameworky se dají považovat CSS frameworky obsahující prvky, které usnadňují tvorbu layoutu pomocí předdefinovaných tříd. CSS frameworky většinou zahrnují reset stylů, typografii, obecná pravidla pro vzhled tiskových sestav, systém pro rozložení layoutu stránek, systém pro definici vzhledu formulářů, různé sady ikon, vzhled tlačítek atp. (Michálek, 2009).

Existuje již mnoho responzivních CSS frontend frameworků. Mezi nejpopulárnější patří Twitter Bootstrap a Foundation. Zvláště pro ty, kteří ještě nevytvářeli responzivní web, může být podobný framework poměrně užitečný.

Práci na větších projektech může usnadnit používání CSS preprocesoru. CSS preprocesory v podstatě zjednodušují kód stylů pomocí používání proměnných, mixinů, matematických operací, zanořování selektorů atd. Pro začátečníky je poměrně dobrý program LESS, který má blízko k CSS kódu (Michálek, 2013d).

3.6 Testování

Posledním krokem je testování webových stránek v různých zařízeních a prohlížečích. Existuje mnoho nástrojů, pomocí kterých můžeme webové stránky testovat online. Většina těchto nástrojů slouží pouze k testování *breakpointů*. Některé z nich však umožňují simulaci podmínek na konkrétních zařízeních. Jedním z nástrojů, který slouží k testování *breakpointů*, je web screenqueri.es. Tento web nabízí výběr z několika mobilních zařízení a tabletů. Webová stránka se pak zobrazí v mřížce, která se přizpůsobuje rozlišení zvoleného zařízení. Na webu browserstack.com je možné vyzkoušet si, jak se naše stránky zobrazí například na zařízení s Windows XP v prohlížeči Internet Explorer 8 atd.

Za zmínku jistě stojí emulátory pro různá prostředí – například takový emulátor Opery mobile je velmi užitečný. Můžeme si v něm zkusit, jak se naše stránky zobrazují v různých zařízeních v prostředí prohlížeče Opery mobile. Podobné testování však nedokáže simulovat podmínky na konkrétních zařízeních. Může nám poskytnout jen jakousi představu, jak se náš web na těchto zařízeních zobrazuje. Ve skutečnosti se tento způsob k testování stránek použít nedá. Nic se nevyrovná testování webu v konkrétních zařízeních. Při testování je třeba brát ohled nejen na vzhled stránek, ale také na použitelnost a rychlost načítání dat. Je dobré vyzkoušet si, zda je na zařízeních čitelný text, jestli jsou tlačítka dostatečně velká atp. (Ralston, 2013).

4 Tvorba vlastní responzivně optimalizované stránky

Projekt řešený v rámci praktické části bakalářské práce je vytvoření jednoduché webové prezentace o kreslení. Úkolem je navrhnout a vytvořit webovou prezentaci tak, aby se přizpůsobovala různým zobrazovacím zařízením. Protože jde jen o jednoduchou webovou stránku, není vhodné vytvářet separátní mobilní verzi. Nejlepším řešením je výstavba stránek na základech responzivního designu. Všechny soubory k webové prezentaci jsou umístěny v příloze B.

Cíle a předpoklady

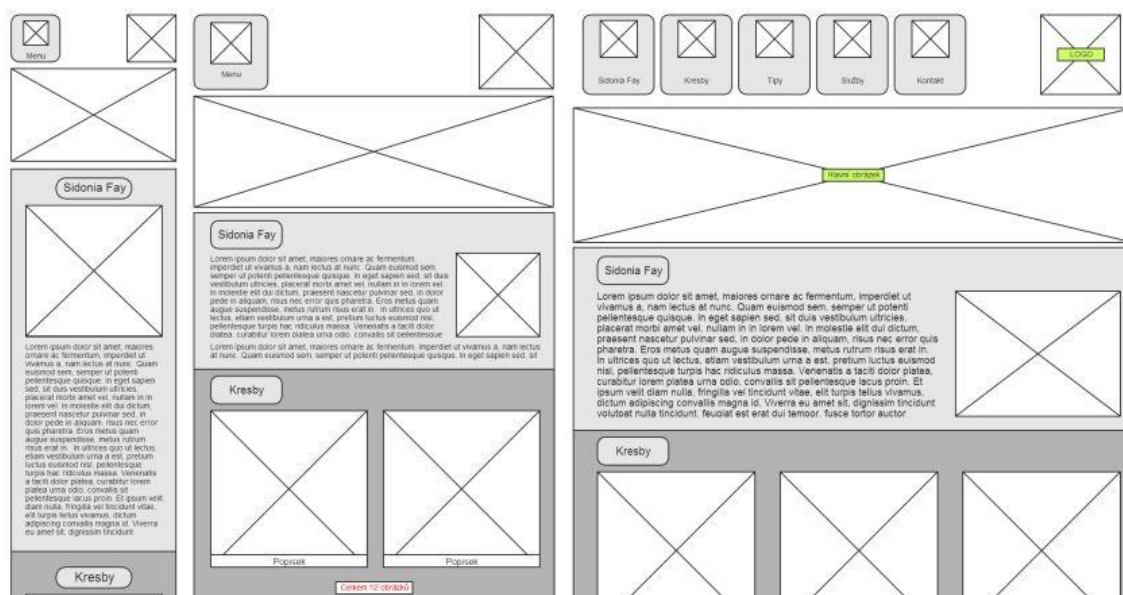
Při vytváření stránek bude brán ohled na zásadu postupného vylepšování a *Mobile First* v CSS. Obsah stránek bude dostupný i v prohlížečích, které nezachytí dotazy na média. Na jednotlivých zařízeních se může vzhled stránek drobně lišit. Stránky budou navrženy aspoň ve 3 úrovních responzivnosti. Webové stránky budou využívat principů responzivního designu, jak je popsal Marcotte. V začátcích výstavby webu byly brány v potaz starší verze Internet Exploreru. Vzhledem k nedávnému ukončení podpory systémů Windows XP bude postupně uživatelů nižších verzí Internet Exploreru ubývat. Z tohoto důvodu nebudou webové stránky testovány v zařízeních s prohlížečem Internet Explorer 8 a nižších verzích. Stránky budou testovány v některých zástupcích nejpoužívanějších typů zobrazovacích zařízení (smartphony, tablety, netbooky, desktopové počítače) v různých internetových prohlížečích. Testování bude probíhat i pomocí některých online nástrojů a emulátorů. Stránky budou dostupné online na této adrese: <http://sidoniafay.himinninn.cz/>. Webové stránky slouží jen pro účely této bakalářské práce. Neberou v potaz optimalizaci pro vyhledávače, počáteční analýzu konkurence či cílové skupiny uživatelů.

4.1 Sestavení obsahu stránek

Nejprve byl sestaven veškerý obsah, který byl použit při výstavbě stránek. Obsah byl následně roztríděn do tematických sekcí (Sidonia Fay, Kresby, Tipy, Služby a Kontakt). Kromě textů byly vybrány také všechny obrázky, které jsou na stránkách použity. Veškerý text i obrázky jsou vlastní tvorbou autora této bakalářské práce.

4.2 Drátový model

Drátový model byl vytvořen pro tři úrovně responzivnosti. Obsah stránek není rozsáhlý, a proto nebylo nutné vytvářet několik samostatných stránek. Drátový model byl vytvořen pomocí programu mockingbird. Jde o online nástroj pro vytváření drátových modelů. Je velmi jednoduchý a podobá se vlastně práci s tužkou a papírem. (Dostupný na adrese: <https://gomockingbird.com/mockingbird/>.) Následující obrázek zobrazuje náhled na drátové modely vytvořené pomocí nástroje mockingbird (Obr. 4.1).



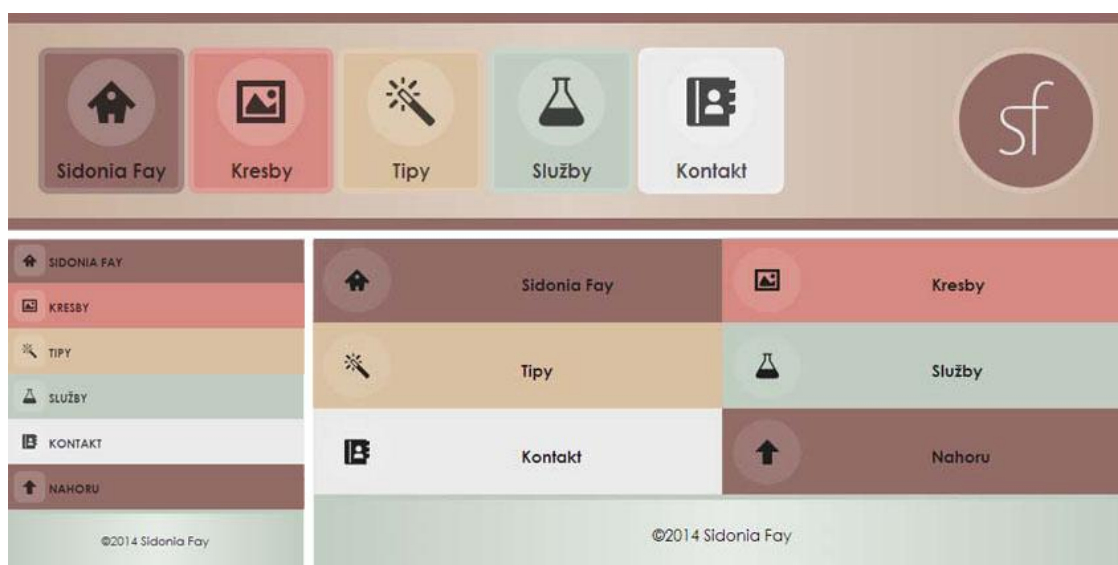
Obr. 4.1:

Náhled na drátové modely pro tři úrovně responzivnosti – od nejmenší obrazovky po největší (zdroj: autor)

Na obrázku můžeme vidět, jak se jednotlivé prvky při různých velikostech obrazovky mění a přesouvají. Největší transformací prochází hlavní navigace stránek. Na desktopových počítačích a tabletech držených na šířku se menu zobrazuje jako pět samostatných položek. Na většině tabletů držených na výšku a mobilních telefonech se místo navigace zobrazí jen jedna zastupující položka, která se aktivuje kliknutím a zobrazí všechny položky navigace.

4.3 Vytvoření prototypu a grafického návrhu stránek

Před vytvořením grafického návrhu webových stránek byl dle drátových modelů vytvořen jednoduchý funkční prototyp stránek, na kterém byla otestována funkčnost a použitelnost stránek. Byla vytvořena jednoduchá HTML a CSS šablona. CSS soubor obsahoval pouze tzv. reset stylů pro starší prohlížeče, několik málo stylů pro základní obalující elementy a jeden dotaz na média. Navigace je vytvořená dle zásady *Mobile First*. Jde o velmi jednoduchou metodu, která funguje čistě na bázi kaskádových stylů. Samotná navigace je v HTML kódu umístěna těsně před patičkou stránek a nahoře v hlavičce je pouze ikona s odkazem na navigaci. Po kliknutí se jednoduše přesuneme přímo na navigaci. Pro větší zařízení se navigace pomocí absolutní pozice posouvá nahoru do hlavičky stránky a mění svůj vzhled. Různý vzhled navigace je znázorněn na následujícím obrázku (Obr. 4.2).



Obr. 4.2:

Nahoře menu pro desktop, vlevo dole pro mobilní telefony, vpravo dole pro tablety a menší desktopy (zdroj: autor)

Menu se transformuje ve chvíli, kdy by se jinak začaly překrývat položky navigace s logem. Inspirací pro grafické zpracování navigace byl článek od Stéphanie Walter (Walter, 2013). Navigace vytvořená Stéphanie Walter funguje však na principu JavaScriptu. Většina stránek používá k řešení problému navigace u responzivního webu více než jen čisté CSS řešení. V tomto konkrétním projektu bylo zvoleno čisté CSS řešení ze dvou důvodů. Zaprvé jde o jednoduché jednoúrovňové menu. Zadruhé je zde jistota, že bude fungovat i na zařízeních nepodporujících JavaScript.

Jde sice o jednouché řešení, ovšem rozhodně ne ideální. Zvláště z uživatelského hlediska může takto řešená navigace působit rušivě a kazit uživatelský prožitek.

Vzhled stránek je v dnešní době velmi důležitý. Pokud návštěvníka webu nezaujme jeho grafické zpracování, s velkou pravděpodobností ho záhy opustí, aniž by se zajímal o jeho obsah. V této fázi byl nejprve vybrán barevný koncept. Každá záložka je jinak barevná spolu s odpovídající barvou položky navigace. Poté bylo vytvořeno logo a hlavní obrázek. V logu jsou spojena písmena S a F. Tato písmena jsou iniciály pseudonymu autorky kreseb, Sidonie Fay. Logo bylo vytvořeno ve třech různých velikostech a hlavní obrázek také. Různým velikostem obrazovky jsou odesílány jiné velikosti obrázků.

Obrázkové ikony použité při implementaci jsou jedním ze zdarma dostupných icon fonts. Ikony jsou vygenerovány pomocí aplikace IcoMoon. (Zjistit více o aplikaci IcoMoon je možné na této adrese: <http://icomoon.io/#hi>.) V aplikaci IcoMoon si jednoduše vybereme ikony, které chceme na stránkách použít. Poté si můžeme vybrané ikony stáhnout a vložit do svého projektu.

Velkou předností icon fontů je, že se s nimi dá pracovat stejně jako s běžným písmem. Dá se jim nastavovat velikost pomocí vlastnosti `font-size` a jednoduše měnit jejich barva. Vkládají se do CSS souboru pomocí direktivy `@font-face`. Font-face je v prohlížečích poměrně dobře podporován. Podle testů, které byly provedeny pomocí knihovny Modernizr, nepodporuje `@font-face` jen prohlížeč Opera Mini a Internet Explorer pro Windows Phone ho podporuje až od 10. verze. Tudíž použití icon fontu se zdá být poměrně bezpečné. U všech prvků s ikon fontem je vložen do HTML kódu element `<span>` s popiskem. V případě, že prohlížeč nepodporuje `@font-face`, zobrazí se aspoň textový popis. Ve výpisu 10 je část kódu, který sloužil k otestování podpory `@font-face` v různých prohlížečích.

Výpis 10: JavaScript k testování podpory font-face v prohlížečích.

```
<script type="text/javascript">
  if (Modernizr.fontface) {
    document.write('Prohlížeč @font-face podporuje');
  } else {
    document.write('Prohlížeč @font-face nepodporuje');
  }
</script>
```

Pokud je `@font-face` podporován, potom tento skript (Výpis 10) vypíše první větu („Prohlížeč `@font-face` podporuje“). Pokud prohlížeč `@font-face` nepodporuje, JavaScript vypíše druhou větu („Prohlížeč `@font-face` podporuje“). Tento skript byl použit jen v prototypu stránek.

Výpis 11 ukazuje, jakým způsobem je vložen icon font do CSS souboru.

Výpis 11: Vložení icon fontu do CSS souboru.

```
/* @font-face */
@font-face {
    font-family: 'icomoon';
    src: url('../fonts/icomoon.eot');
    src: url('../fonts/icomoon.eot?#iefix') format('embedded-opentype'),
        url('../fonts/icomoon.woff') format('woff'),
        url('../fonts/icomoon.ttf') format('truetype'),
        url('../fonts/icomoon.svg#icomoon') format('svg');
    font-weight: normal;
    font-style: normal;
}

/* nastavení icon fontu pro konkrétní prvek */

.icon-da {
    font-family: 'icomoon';
    font-style: normal;
    font-weight: normal;
    font-variant: normal;
    text-transform: none;
    line-height: 1;
}

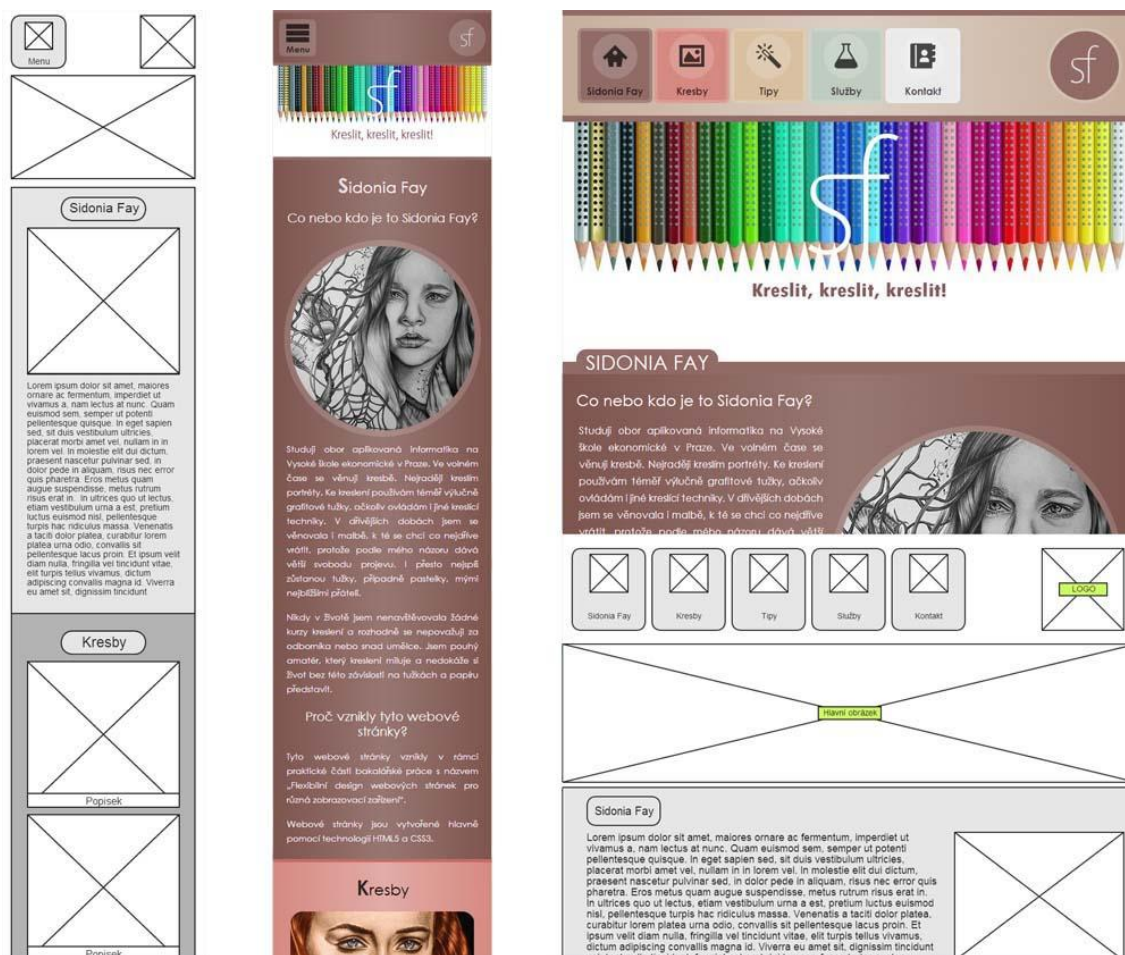
.icon-da:before {
    content: "\e609";
}
```

V prohlížeči Google Chrom je problém s „nehezským“ vykreslením ikon. Existuje trik, který tento problém řeší. Výpis 12 ukazuje řešení „nehezkého“ vykreslení ikon v tomto prohlížeči.

Výpis 12: Náprava zobrazení fontů v prohlížeči Chrome.

```
/* Windows Chrome ugly fix http://stackoverflow.com/questions/13674808/chrome-
svg-font-rendering-breaks-layout/14345363#14345363 */
@media screen and (-webkit-min-device-pixel-ratio:0) {
  @font-face {
    font-family: 'icomoon';
    src: url('../fonts/icomoon.svg#icomoon') format('svg');
  };
}
```

Na obrázku můžeme vidět, jak velký je rozdíl mezi drátovým modelem a samotným grafickým návrhem (Obr. 4.3).



Obr. 4.3:

Rozdíly mezi drátovými modely a samotným grafickým provedením (zdroj: autor)

4.4 Sestavení HTML kódu

Na prvním řádku HTML kódu je specifikace DOCTYPE pro HTML5: `<!DOCTYPE html>`. Jako další následují podmíněné komentáře (Janovský, 2014) kvůli starším prohlížečům Internet Exploreru (Výpis 13). Tyto body class pak mohou být použity v CSS dokumentu.

Výpis 13: Podmíněné komentáře pro starší verze Internet Explorer.

```
<!--[if lt IE 9 ]><html class="oldie no-js" lang="cs"><![endif]-->
<!--[if gt IE 8]><!--><html class="no-js" lang="cs"><!--<![endif]-->
```

První podmíněný zápis můžeme použít pro prohlížeče Internet Explorer 8 a nižší verze. Ostatní prohlížeče a jiné verze Internet Exploreru tyto podmíněné zápisy ignorují, protože je považují za poznámku. Druhý zápis je zaměřen na Internet Explorer 9 a vyšší verze.

Důležitý je element `<meta>` se jménem `viewport`, který je vložen do sekce `<head>` (Výpis 14) Tento element je podrobně popsán v předchozí kapitole.

Výpis 14: Použitý element `<meta>` s meta klíčem `viewport`.

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

Výraz `width="device-width"` zajistí přizpůsobení šířky zobrazované plochy šířce obrazovky. Výraz `initial-scale` nastavený na hodnotu 1 určuje výchozí úroveň přiblížení na 100%. `Maximum-scale` je nastaveno tak, že měřítko zobrazované oblasti je maximálně 100%.

Sekce `<head>` obsahuje také skript s odkazem na knihovnu Modernizr (Výpis 15).

Výpis 15: Skript s odkazem na knihovnu Modernizr.

```
<!-- Scripts -->
<script type="text/javascript" src="js/modernizr.custom.sf.js"></script>
```

`Modernizr.custom.sf.js` obsahuje jen testy pro `@font-face` a zahrnuje skript `html5 Shiv`, který umožňuje fungování prvků HTML5 ve starších prohlížečích.

Před koncovým elementem `</body>` je umístěn skript `respond.js` (Výpis 16). Tento skript je popsán v předchozí kapitole.

Výpis 16: Skript s odkazem na skript *respond.js*.

```
<!--[if lte IE 8]>  
  <script src="./js/respond.js"></script>  
<![endif]-->
```

Skript *respond.js* bude použit pro prohlížeč Internet Explorer 8 a nižší verze. Zajistí funkčnost *media queries* ve starších prohlížečích Internet Explorer.

Na stránkách jsou využity nové elementy HTML5 jako jsou `<article>`, `<nav>`, `<header>` a `<footer>`. Celý HTML kód je umístěn v příloze C. HTML kód je validní.

4.5 Vytvoření CSS šablony

Na začátek dokumentu CSS je vložen Meyerův reset stylů (popsán v předchozí kapitole). Dále jsou vytvořeny základní CSS styly pro všechna zařízení a prohlížeče. Při vytváření těchto základních stylů bylo postupováno dle metody *Mobile First*. Tím by měla být zajištěna dostupnost stránek (respektive jejich obsahu) v základní podobě ve všech zařízeních. Velikost fontu byla nastavena v elementu `<body>` na 100 %. Velikosti nadpisů jsou určeny v jednotkách *Em*. Velikosti písma jsou upravovány pod dotazy na média jednoduchým způsobem. Elementu `<body>` je vždy změněna velikost fontu pomocí procent. Například u *breakpointu* 480px a méně je font-size nastaveno na 87.5 %, což znamená 14px. (Písmo nastavené na 100 % = 16px.) Po základních CSS stylech následují styly pro různé šířky obrazovky. Pro zařízení s menší obrazovkou byly zvoleny *breakpointy* dle některých reálných zařízení. U větších obrazovek se *breakpointy* řídí spíš designem. Jsou nastavené podle toho, kde nastal problém s překrýváním prvků nebo nevhodným obtékáním prvků. Celá šablona stylů CSS je umístěna v příloze D. Výpis 17 obsahuje zvolené *breakpointy*.

Výpis 17: Zvolené *breakpointy*.

```
/* 0px a více - styly pro všechny zařízení podporující media queries
----- */
@media all and (min-width: 0px) {/* styly CSS */}

/* 480px a méně
----- */
@media only screen and (max-width: 480px) {/* styly CSS */}

/* 481px a více
----- */
@media only screen and (min-width: 481px) {/* styly CSS */}

/* rozpětí mezi 481px - 859px
----- */
@media only screen and (min-width: 481px) and (max-width: 859px) {/* styly CSS
*/}
/* 860px a více
----- */
@media only screen and (min-width: 860px) {/* styly CSS */}

/* 1200px a více
----- */
@media only screen and (min-width: 1200px) {/* styly CSS */}
```

Výpis 17 obsahuje všechny hlavní *breakpointy*. V dokumentu je použito ještě pár doplňujících *breakpointů*, které musely být použity, kvůli některým prvkům, které nešlo zařadit pod tyto *breakpointy*. *Breakpoint* 480px byl vybrán, protože dokáže zachytit mnoho smartphonů držených na výšku. Ostatní *breakpointy* se řídí designem. Nesnaží se detekovat žádná konkrétní zařízení.

V CSS stylovisu jsou použity nové CSS3 selektory. Například položky v menu mají nastavené barvy pomocí pseudo tříd. Menu obsahuje pět barev a pět položek. Ovšem ve verzi pro menší obrazovky má menu šest položek. Šestá položka obsahuje odkaz, který navrací na začátek stránky. Je to z důvodů zachování symetrie menu, které je v rozpětí mezi 481px až 859px dvousloupcové (viz obr. 4.2). (K návratu na začátek stránky slouží také fixní prvek (odkaz) s nápisem „nahoru“. Fixní pozice však může být v některých prohlížečích problematická.) Šestá položka v menu bude pak mít stejnou barvu jako první položka v menu a to díky tomuto kódu ve výpisu 18.

Výpis 18: Použití pseudo tříd u položek menu.

```
#nav li:nth-child(5n+1) a {  
    background: rgb(146, 107, 102); /* kód se uplatní u 1. a 6. položky menu */  
}  
#nav li:nth-child(5n+2) a {  
    background: rgb(214,137,129);  
}  
#nav li:nth-child(5n+3) a {  
    background: rgb(217,192,161);  
}  
#nav li:nth-child(5n+4) a {  
    background: rgb(192,204,192);  
}  
#nav li:nth-child(5n+5) a {  
    background: rgb(234, 234, 234);}
```

Hlavní obrázek stránky je vložen jen jako obrázek na pozadí, takže není vložen do HTML kódu. Díky tomu můžeme odesílat zařízením s různými velikostmi obrazovky různě velké obrázky. Jakým způsobem je obrázek na stránky vložen, je patrné z Výpisu 19 a Výpisu 20.

Výpis 19: Vložení hlavního obrázku na webové stránky – HTML kód.

```
<!-- ==== HLAVNÍ OBRÁZEK ==== -->  
<div id="main-img">  
    <span id="pencils" role="img" aria-label="Main picture with logo and  
        color pencils.">  
        <span class="in"></span>  
    </span>  
</div>  
<!-- konec hlavní obrázek -->
```

Výpis 20: Vložení hlavního obrázku na webové stránky – CSS kód.

```
div#main-img {
    max-width: 1200px;
    margin: 0 auto;
    text-align: center;
}

div#main-img #pencils {
    width: 100%;
    display: inline-block;
    vertical-align: middle;
    font: 0/0 serif; /*zabraňuje objevení bílého prostoru pod plovoucím
                    elementem*/
    color: transparent;
    background-size: 100%;
    background-position: top center;
    background-repeat: no-repeat;
}

span.in {
    padding-top: 41.6666%; /* 500px/1200px */
    display: block;
    height: 0;
}

#pencils {
    background-image: url("../img/main-pastelky-basic.jpg");
}

@media only screen and (max-width: 480px)
{
    #pencils {
        background-image: url("../img/main-pastelky-mobile.jpg");
    }
}
```

Výpis 20 neobsahuje všechny dotazy na médium, v nichž jsou vloženy další verze (různé velikosti) stejného úvodního obrázku.

Obrázky v jednotlivých člancích jsou flexibilní. Jsou umístěny přímo do elementu `<img>` v HTML kódu. V CSS jsou jim nastaveny následující vlastnosti (Výpis 21).

Výpis 21: CSS vlastnosti u flexibilních obrázků.

```
#img-sidonia,  
#img-sansa,  
#img-oci,  
#img-obrysy,  
#img-linky,  
#img-predloha,  
#img-potreby,  
#img-svoboda {  
    max-width:100% !important;  
    height:auto;  
    display:inline-block;  
    border:none;  
}
```

Obrázek v prvním článku má nastavené obtékání (`float:right`) až od 635px, kdy se vedle něj vejde rozumné množství textu. Jinak je vycentrován na střed bez obtékání textem. Obrázková galerie obsahuje dvanáct obrázků. Jsou vloženy přímo do HTML kódu. Podle základních CSS stylů se obrázky zobrazí v jednom sloupci pod sebou. Pomocí dotazů na média je obalujícím prvkům obrázkům nastaveno obtékání zleva a různé šířky v %. Při šířce 860px a více, se vedle sebe zobrazí tři obrázky. V rozpětí mezi 481px až 859px se vedle sebe zobrazí dva obrázky. Pod 480px jsou uplatněny styly ze základní části šablony stylů a obrázky jsou zobrazeny v jednom sloupci. Popisky obrázků se zobrazují při najetí kurzorem na danou kresbu. V dotekových zařízeních se popisky zobrazí po klepnutí na obrázek pomocí jQuery. Elementu `<body>` je nastavena třída `no-touch`. Na dotykových zařízeních se tato třída mění na `touch`. S těmito třídami pak je možné pracovat v CSS dokumentu. Výpis 22 obsahuje kód jQuery použitý pro zobrazování popisků u obrázků.

Výpis 22: jQuery kód použitý pro zobrazování popisků u obrázků v galerii.

```
// vložení knihovny jQuery
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js">
</script>
<script type="text/javascript">
    $(function(){
        if ('ontouchstart' in window) // Detekce dotekového zařízení
        {
            // Nastavení třídy touch pro dotyková zařízení
            $('body').removeClass('no-touch').addClass('touch');

            //Vybere nejbližšího předka obrázku v galerii a přidá mu třídu tapped
            $('div.inside-box img').click(function(){
                $(this).closest('.inside-box').toggleClass('tapped');
            });
        }
    });
</script>
```

Layout stránek se přizpůsobuje maximálně do šířky 1200px (max-width:1200px, pro starší verze Internet Exploreru jen width:1200px). Důvodem je hlavně obsah stránek. Při větší šířce by články nevypadaly dobře, není na nich dostatečné množství obsahu.

4.6 Testování

Webové stránky byly testovány pomocí některých emulátorů a online nástrojů již v průběhu tvorby. Finální testování proběhlo v reálných zařízeních. Tato podkapitola zahrnuje záznamy a výsledky z testování.

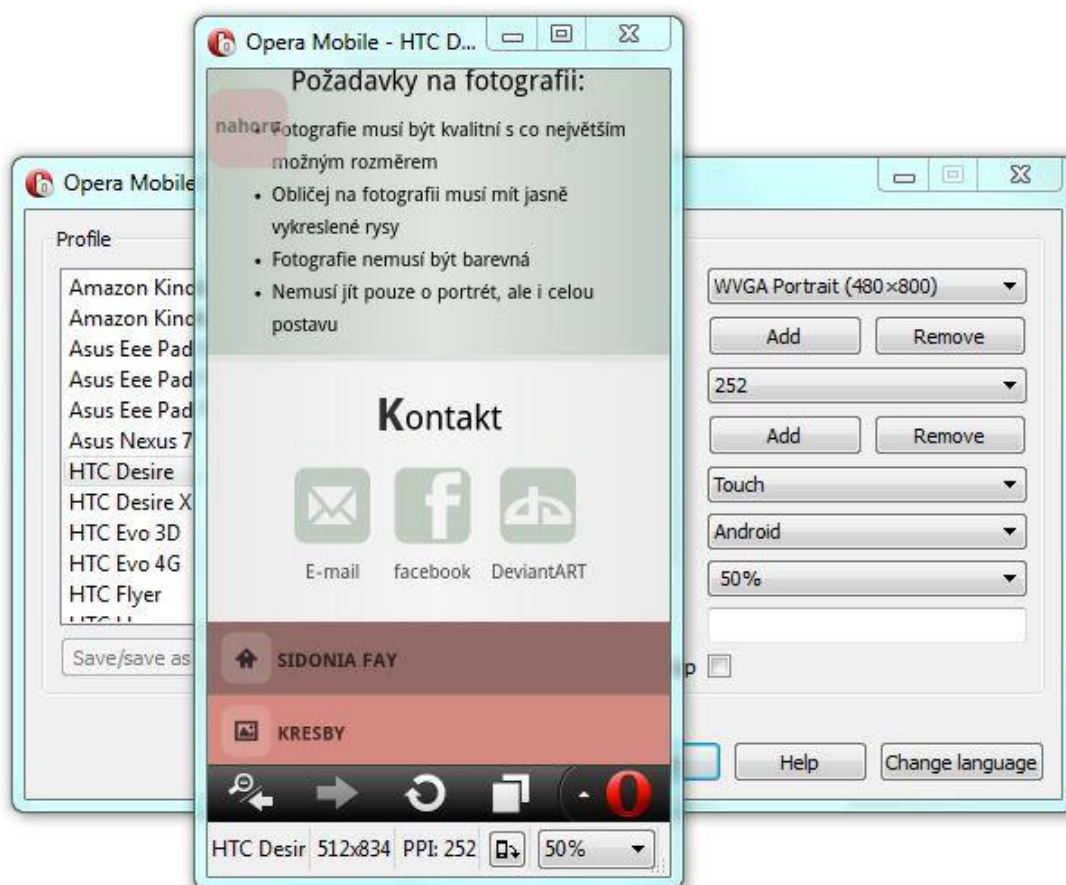
4.6.1 Online nástroje pro a emulátory pro testování

Do této kategorie nástrojů pro testování byl vybrán Emulátor Opera Mobile, server browserstack.com a server screenqueri.es. Emulátor pro Operu Mobile jako jediný z nich nepracuje online.

Emulátor pro Operu Mobile

Tento emulátor umožňuje výběr z několika různých zařízení. Zobrazení stránek je pak nasimulováno v daném zařízení v prostředí Opery Mobile. Emulátor simuluje i dotykový display.

Pohybujete se v něm sice pomocí myši, ale ukazatel funguje spíše jako prst. Emulátor Opery Mobile byl používán v průběhu vytváření stránek k případnému zachycení některých problémů. Dotazníkové šetření ukázalo, že mnoho lidí Operu nepoužívá, ale u mobilních telefonů se jich přece jen několik našlo. Testování stránek v Opeře Mobile tedy určitě má svůj význam. V simulátoru pro Operu Mobile nebyl se stránkami zaznamenán žádný problém. Na obrázku je ukázka testu v mobilním telefonu HTC Desire (Obr. 4.4).

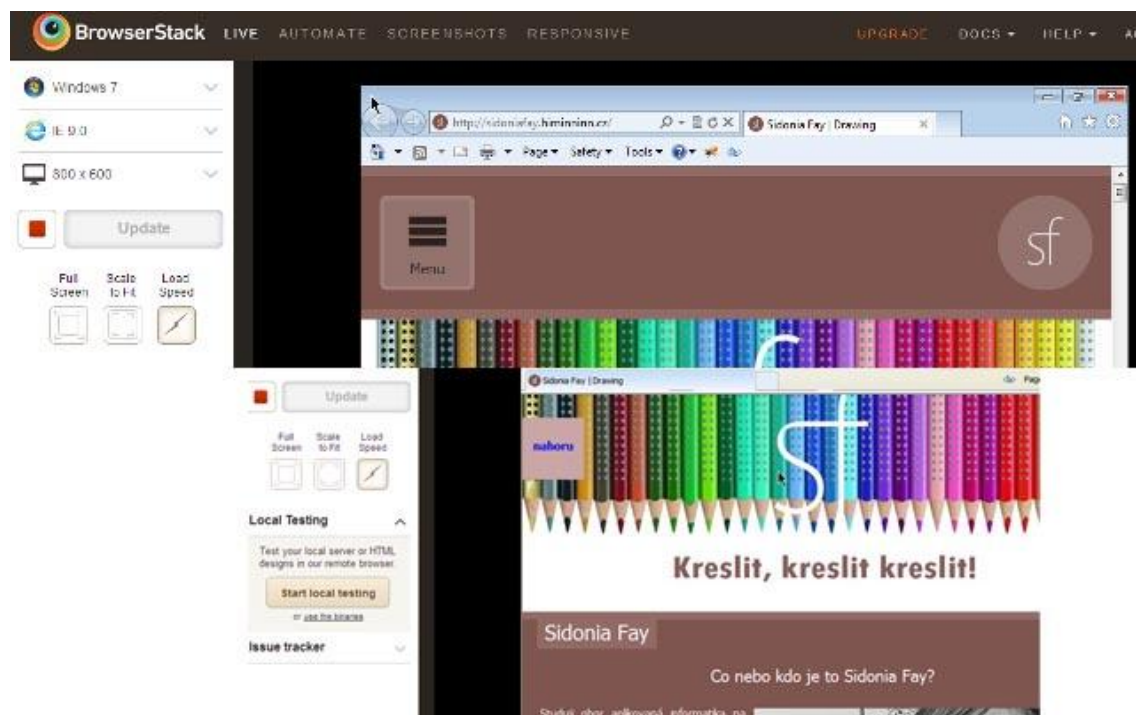


Obr. 4.4:

Ukázka testování stránek v prostředí emulátoru Opera Mobile (zdroj: autor)

Browserstack.com

Server browserstack.com nabízí poměrně rozsáhlé možnosti testování. Stránky v něm můžeme testovat například ve starších prohlížečích Internet Explorer nebo v různých mobilních platformách. Ukázka z testování stránek ve Windows XP v prohlížeči Internet Explorer 8 a ve Windows 7 v prohlížeči Internet Explorer 9 je na následujícím obrázku (Obr. 4.5).



Obr. 4.5:

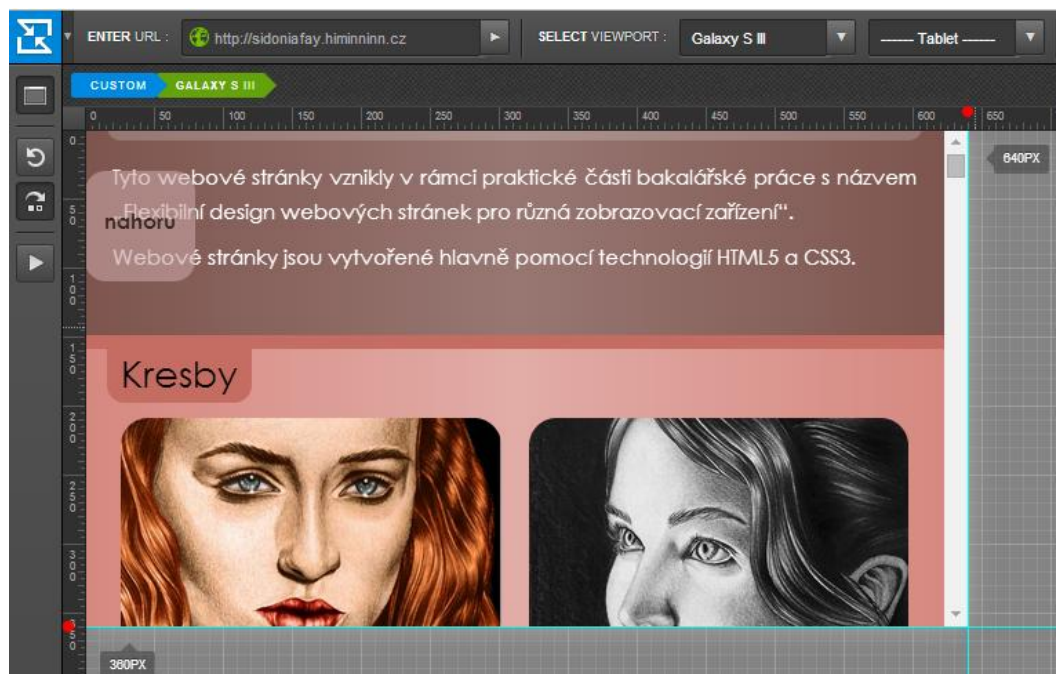
Testování stránek na serveru browserstack.com. Nahoře v IE 9, dole v IE 8 (zdroj: autor)

V Internet Exploreru 9 není se zobrazením stránek žádný problém. V Internet Exploreru 8 se stránky zobrazují poměrně dobře. *Media queries* jsou zaznamenány. Nefunguje průhlednost prvků nebo zaoblené okraje, ale například icon fonty se v něm zobrazují (ovšem v modré barvě, jako ostatní odkazy). Některé menší problémy s grafickými prvky byly zaznamenány na starších zařízeních od Applu. Například v iPhone 3 se nezobrazovaly správně zaoblené hrany. Na browserstack.com byly stránky otestovány také v několika zařízeních s Androidem. Problém byl opět se zaoblenými hranami v některých straších přístrojích. Na webu browserstack.com byly stránky otestovány i v některých zařízeních, která byla zařazena do reálného testování. Simulátor ve všech případech odpovídal realitě. Šlo o zařízení iPhone 5, iPad 4, Samsung Galaxy SIII a Samsung Galaxy Note II.

Screenqueri.es

Na serveru screenqueri.es je možné testovat použité *breakpointy* podle reálných zařízení. Slouží jen pro představu, jak stránky mohou v daných zařízeních vypadat. Na výběr je zde z několika tabletů a smartphonů. Navíc je možno s mřížkou uvnitř volně pohybovat a nastavit si tak unikátní rozlišení.

V tomto testeru *breakpointů* se žádná chyba webových stránek neprokázala. Tester použité dotazy na média zachytil. Ukázku z testování *breakpointů* na serveru screenqueri.es můžeme vidět na dalším obrázku (Obr. 4.6).



Obr. 4.6:

Testování na serveru mediaqueri.es v Samsungu Galaxy SIII otočeným na šířku (zdroj: autor)

4.6.2 Testování v různých zařízeních

Závěrečné testování stránek probíhalo v reálných zařízeních. Do testu se dostala široká škála zařízení. Všechna zařízení jsou poměrně nová s novějšími verzemi operačních systémů. Pokud to bylo možné, byly stránky otestovány na jednom zařízení ve více prohlížečích. Ne každý vlastník daného zařízení byl ochotný kvůli testování nainstalovat do svého přístroje další prohlížeč. Nicméně na chytrých telefonech byly stránky testovány v prohlížečích Google Chrome, Android Browser, Opera Mobile, Opera Mini, Safari, Mozilla Firefox, Internet Explorer 9 a Internet Explorer 11. V tabletech byly stránky testovány v prohlížečích Google Chrome, Android Browser, Opera, Safari, Dolphin a Mozilla Firefox. V ostatních zařízeních byly webové stránky otestovány v prohlížečích Google Chrome, Opera, Internet Explorer a Mozilla Firefox. Až na Internet Explorer 9 u mobilního telefonu Nokia Lumia 800, byly stránky testovány v nejnovějších verzích všech zmíněných prohlížečů.

Nejvíce zařízení zařazených do testování je z kategorie chytrých telefonů. Dále byly stránky otestovány na třech tabletech od různých výrobců. V kategorii netbooků a mini notebooků je zařazeno do testu jen jedno zařízení. Netbooky/ Mini notebooky jsou podobné notebookům, jen mají menší výkon a obrazovky. V kategorii notebooků je také jen jedno zařízení. Notebook s rozlišením 1280px na 800px. Vzhledem k tomu, že se stránky přizpůsobují jen do šířky 1200px, je tento notebook k otestování stránek dostatečný. Následující tabulka obsahuje přehled zařízení zařazených do testování (Tab. 4.1):

Tab. 4.1: Zařízení zařazená do testování webových stránek (zdroj: autor)

Zařízení	Operační systém
Tablety:	
ASUS Transformer Prime TF201	Android 4.1.1
Samsung Galaxy Tab 3 – 7“	Google Android 4.1
iPad 4. Generace	iOS 6
Mobilní telefony:	
iPhone 5	iOS 6
HTC Desire 500	Android 4.2
Samsung Galaxy SIII mini	Android 4.1
Samsung galaxy Note II	Android 4.1
Nokia Lumia 720	Windows Phone 8.1 Developer Preview
Nokia Lumia 800	Windows 7.5
GIGABYTE GSmart Roma R2	Android 4.2
Netbooky/ mini notebooky:	
ASUS Eee PC 1005PXD	Windows 7 Starter
Notebooky:	
Acer Aspire 5530	Windows 7

Tabulka obsahuje záznam z testování v různých prohlížečích a zařízeních (Tab. 4.2). Sloupeček „*rozeznání media queries*“ říká, zda prohlížeč rozeznal *breakpointy*. V sloupečku „*jiné problémy se zobrazením*“ jsou zaznamenány další chyby v zobrazení stránek. Písmeno „X“ značí, že v daném prohlížeči stránky nebyly testovány.

Tab. 4.2: Testování stránek v různých zařízeních a prohlížečích (zdroj: autor)

Tablety	Google Chrome		Mozilla Firefox		Opera Mobile		Dolphin		Android Browser		Safari	
	Rozpoznání Media Queries	Jiné problémy se zobrazením	Rozpoznání Media Queries	Jiné problémy se zobrazením	Rozpoznání Media Queries	Jiné problémy se zobrazením	Rozpoznání Media Queries	Jiné problémy se zobrazením	Rozpoznání Media Queries	Jiné problémy se zobrazením	Rozpoznání Media Queries	Jiné problémy se zobrazením
ASUS Transformer Prime TF201	Ano	Ne	Ano	Ne	Ano	Ne	Ano	Ne	Ano	Ne	X	X
iPad 4. generace	Ano	Ne	X	X	X	X	X	X	X	X	Ano	Ne
Samsung Galaxy Tab 3	Ano	Ne	Ano	Ne	Ano	Ne	Ano	Ne	Ano	Ne	X	X
Smartphony	Google Chrome		Mozilla Firefox		Opera Mobile / Opera Mini		Internet Explorer		Android Browser		Safari	
iPhone 5	X	X	X	X	X	X	X	X	X	X	Ano	Ne
HTC Desire 500	Ano	Ne	Ano	Ne	Ano/Ne	Ne/Ano	X	X	Ano	Ne	X	X
Samsung Galaxy SIII mini	Ano	Ne	X	X	X	X	X	X	Ano	Ne	X	X
Samsung galaxy Note 1	Ano	Ne	X	X	X	X	X	X	Ano	Ne	X	X
GIGABYTE GSmart Roma R2	Ano	Ne	X	X	X	X	X	X	X	X	X	X
Nokia Lumia 720	X	X	X	X	X	X	Ano	Ne	X	X	X	X
Nokia Lumia 800	X	X	X	X	X	X	Ano	Ano	X	X	X	X
Netbooky/Mini notebooky	Google Chrome		Mozilla Firefox		Opera		Internet Explorer 11					
ASUS Eee	Ano	Ne	Ano	Ne	Ano	Ne	Ano	Ne	X	X	X	X
Notebooky	Google Chrome		Mozilla Firefox		Opera		Internet Explorer 11					
Acer Aspire 5530	Ano	Ne	Ano	Ne	Ano	Ne	Ano	Ne	X	X	X	X

Z tabulky je patrné, že při testování stránek nebylo zaznamenáno příliš mnoho problémů. Problémy se objevily pouze v prohlížečích Opera Mini a Internet Explorer 9 u přístroje Nokia Lumia 800. Druhý telefon s Windows Phone zařazený do testu (Nokia Lumia 720) používala Internet Explorer 11 a v něm nebyl zaznamenán žádný problém. Rozhodně nejhůř vyšla z testu Opera Mini, která nerozeznala dotazy na média. Po zkoušce odstranit z dotazu na médium klíčové slovo *only* a nahrazením typu média místo *screen* na *all*, na tom byla Opera Mini s rozpoznáváním *media queries* o něco lépe. Jen díky metodě *Mobile First* je v Opeře Mini dostupný aspoň jejich obsah. Dále je v Opeře Mini problém s podporou *@font-face*, ale díky elementům `<span>` s popisky odkazů je aspoň jasné, k čemu dané odkazy slouží. Problematický je také fixní odkaz k návratu na začátek stránek, který zůstává skrytý za hlavičkou stránek a vůbec se tak neobjeví. Ani JavaScript, který po kliknutí zobrazuje popisky obrázků, v Opeře Mini nefunguje. Problémů v prohlížeči Internet Explorer 9 v mobilním telefonu Nokia Lumia 800 bylo podstatně méně. *Media queries* Internet Explorer 9 rozeznal a uplatnil daná pravidla stylů. Nefunguje zde také fixní odkaz k návratu na začátek stránek a prohlížeč nepodporuje *@font-face*, tudíž se zde opět zobrazují jen popisky odkazů.

V ostatních prohlížečích nebyl s rozeznáváním *media queries* žádný problém a stránky se v nich zobrazovaly a fungovaly správně. Na dalším obrázku jsou screenshoty z telefonu Nokia Lumia 720 v Internet Exploreru 11 (Obr. 4.7).



Obr. 4.7:

Screenshoty z Internet Exploreru 11 v zařízení Nokia Lumia 720 (zdroj: autor)

5 Závěr

Práce s tématem „*Flexibilní design webových stránek pro různá zobrazovací zařízení*“ se zabývá možnostmi optimalizace webových stránek pro různá zobrazovací zařízení. V začátcích práce jsou zmíněny přístupy, pomocí kterých lze takové stránky vytvořit – stránky s flexibilním designem (responzivní design) a separátní verze webových stránek. Teoretická i praktická část se více soustřeďuje na responzivní design. Responzivní přístup k výstavbě webů je dnes velmi populární a rozšířený. Separátní verze webů však stále mají svůj význam a nejspíš ho nikdy neztratí. Hlavně proto jsou samostatné verze webů v začátcích práce zmiňovány a porovnány s responzivním designem. Doplňující součástí teoretické části je dotazníkové šetření. Jeho cílem bylo v podstatě prokázat, že má téma flexibilního designu webových stránek stále větší význam. Podle jeho výsledků lze usoudit, že ano. Více než 80 % lidí ve věku 15 až 30 let totiž odpovědělo, že vlastní kromě desktopových počítačů i jiná zobrazovací zařízení vhodná k prohlížení webových stránek. A více než 50 % lidí se chystá pořídit si zhruba do jednoho roku další podobné zařízení.

V rámci praktické části je vytvořena jednoduchá webová prezentace o kreslení. Tyto stránky jsou vybudovány na základech responzivního designu. Při výstavbě stránek byl brán ohled na metodu *Mobile First* v CSS. V závěru praktické části jsou stránky otestovány v různých zařízeních a prohlížečích. Hlavním cílem testování bylo zjistit, jaká je podpora *media queries* v moderních prohlížečích. Bylo zjištěno, že podpora *media queries* je v moderních prohlížečích velmi dobrá. Problémy s rozeznáním dotazů na média byly zaznamenány pouze u prohlížeče Opera Mini. Obsah stránek měl být přístupný i v prohlížečích, které *media queries* nezachytí. Že tomu tak je, bylo zjištěno při testování stránek v prohlížeči Opera Mini, který *media queries* ignoroval.

Responzivní design je stále poměrně novou oblastí webdesignu. Jde však o oblast, která se velmi rychle rozvíjí. Myšlenka flexibilního přizpůsobení webových stránek všem zařízením je totiž velmi lákavá. Počet mobilních přístupů k internetu stále roste a nikdo neví, jaká nová zařízení s možností surfovat po internetu se objeví v budoucnu. Brzy se možná dočkáme doby, kdy statické webové stránky zcela zmizí.

Seznam literatury

BERNARD, Borek. Úvod do architektury MVC. In: *Zdroják.cz* [online]. 2009a [cit. 2014-04-25]. Dostupné z: <http://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>

BERNARD, Borek. Prezentační vzory z rodiny MVC. In: *Zdroják.cz* [online]. 2009b [cit. 2014-04-25]. Dostupné z: <http://www.zdrojak.cz/clanky/prezentacni-vzory-zrodiny-mvc/>

BÍLEK, Jan. HTML5 versus nativní: debata o mobilních aplikacích. In: *Zdroják.cz* [online]. 2011 [cit. 2014-04-25]. Dostupné z: <http://www.zdrojak.cz/clanky/html5-versus-nativni-debata-o-mobilnich-aplikacich/>

CASTLEDINE, Earle, Myles EFTOS a Max WHEELER. *Vytváříme mobilní web a aplikace pro chytré telefony a tablety*. 1. vyd. Brno: Computer Press, 2013. ISBN 978-80-251-3763-5.

CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.

COYIER, Chris. Logic in Media Queries. In: *CSS-tricks* [online]. 2013 [cit. 2014-05-09]. Dostupné z: <http://css-tricks.com/logic-in-media-queries/>

COYIER, Chris. Make HTML5 Elements Work in Old IE. In: *CSS - Tricks* [online]. 2009 [cit. 2014-05-10]. Dostupné z: <http://css-tricks.com/snippets/javascript/make-html5-elements-work-in-old-ie/>

DRŽKA, Martin a Karel MICHÁLEK. Obama nebo Romney? Rozhovor o mobilních verzích webů s Martinem Držkou. In: *eBRÁNA* [online]. 2013 [cit. 2014-04-25]. Dostupné z: <http://ebrana.cz/magazin/obama-nebo-romney-rozhovor-o-mobilnich-verzich-webu-s-martinem-drzkou>

FISHKIN, Rand. SEO: The Free Beginner's Guide From Moz. In: *Moz: Inbound Marketing and SEO Software, Made Easy* [online]. 2013 [cit. 2014-05-09]. Dostupné z: <http://moz.com/beginners-guide-to-seo>

FISCH, Masha. Mobile-friendly sites turn visitors into customers. In: *Google Mobile Ads Blog* [online]. 2012 [cit. 2014-05-09]. Dostupné

z: <http://googlemobileads.blogspot.cz/2012/09/mobile-friendly-sites-turn-visitors.html>

FORGÁČ, Ján. Mobilní nebo responzivní web?. In: *ArtFocus: Blog* [online]. 2013 [cit. 2014-04-25]. Dostupné z: <http://www.artfocus.cz/blog/mobilni-stranky-nebo-responzivni-web>

JACKSON, Jamal. Overview of Breakpoints in Responsive Web Design.

In: *1stwebdesigner* [online]. 2012 [cit. 2014-04-25]. Dostupné

z: <http://www.1stwebdesigner.com/design/overview-of-breakpoints-in-responsive-web-design/>

JANOVSKÝ, Dušan. Podmíněné komentáře. In: *Jak psát web* [online]. 2014 [cit. 2014-05-09]. Dostupné z: <http://www.jakpsatweb.cz/html/podminene-komentare.html>

JUNEJA, Gopal. What role is played by HTML5 in responsive web design?. In: *Reality On Web* [online]. 2012 [cit. 2014-05-10]. Dostupné

z: <http://realityonweb.com/html5/what-role-is-played-by-html5-in-responsive-web-design/>

KOPERSKI, Adrian. Difference between front-end and back-end development.

In: *Manning Digital* [online]. 2014 [cit. 2014-05-09]. Dostupné

z: <http://manningdigital.com/blog/2014/01/23/difference-between-front-end-and-back-end-development>

LA, Nick. Responsive Design in 3 Steps. In: *Web Designer Wall: Design Trends and Tutorials* [online]. 2011 [cit. 2014-04-25]. Dostupné

z: <http://webdesignerwall.com/tutorials/responsive-design-in-3-steps>

MARCOTTE, Ethan. *RESPONSIVE WEB DESIGN*. New York, New York: A Book Apart, 2011. ISBN 978-0-9844425-7-7.

MELICH, David. Responzivní design webu: Principy, přínosy a rady. In: *ICT manažer* [online]. 2014 [cit. 2014-04-25]. Dostupné

z: <http://www.ictmanazer.cz/2014/03/responzivni-design-webu-principy-prinosy-a-rady/>

MICHÁLEK, Martin. Dočasná nutnost serverové detekce mobilních zařízení.

In: *Vzhůru dolů* [online]. 2013a [cit. 2014-04-25]. Dostupné

z: <http://kratce.vzhurudolu.cz/post/44227784297/docasna-server-detekce>

MICHÁLEK, Martin. Mobile First v CSS. In: *Vzhůru dolů* [online]. 2013b [cit. 2014-

04-25]. Dostupné z: <http://kratce.vzhurudolu.cz/post/42187934506/mobile-first-css>

MICHÁLEK, Martin. Jaké breakpointy zvolit v responzivním webdesignu?. In: *Vzhůru dolů* [online]. 2013c [cit. 2014-04-25]. Dostupné

z: <http://kratce.vzhurudolu.cz/post/46416507703/jake-breakpointy-zvolit-v-responzivnim-webdesignu>

MICHÁLEK, Martin. CSS preprocesory: Nejspíš nějaký potřebujete a nejspíš je to LESS. In: *Vzhůru dolů* [online]. 2013d [cit. 2014-04-25]. Dostupné

z: <http://kratce.vzhurudolu.cz/post/56084086629/css-preprocesory>

MICHÁLEK, Martin. CSS frameworky pro masy, díl první. In: *Zdroják.cz* [online].

2009 [cit. 2014-04-25]. Dostupné z: <http://www.zdrojak.cz/clanky/css-frameworky-pro-masy-dil-prvni/>

MICHÁLEK, Martin a Honza SLÁDEK. 6 chyb cestou do responzivního pekla.

In: *Zdroják.cz* [online]. 2013 [cit. 2014-04-25]. Dostupné

z: <http://www.zdrojak.cz/clanky/6-chyb-cestou-responzivniho-pekla/>

MICROSOFT, Konec podpory pro Windows XP. In: *Microsoft Windows* [online]. 2014

[cit. 2014-05-09]. Dostupné z: <http://windows.microsoft.com/cs-cz/windows/end-support-help>

RALSTON, Steve. Responzivní design: jak otestovat jeho funkčnost.

In: *Interval.cz* [online]. 2013 [cit. 2014-04-25]. Dostupné

z: <http://interval.cz/clanky/responzivni-design-jak-otestovat-jeho-funkcnost/>

SLÁDEK, Honza. Webdesignérův průvodce po HTML5 – díl nultý.

In: *Zdroják.cz* [online]. 2010 [cit. 2014-05-09]. Dostupné

z: <http://www.zdrojak.cz/clanky/webdesigneruv-pruvodce-po-html5-dil-nuly/>

ŠKRÁŠEK, Jan. PHP frameworky. In: *Programujte.com* [online]. 2008 [cit. 2014-04-

25]. Dostupné z: <http://programujte.com/clanek/2008022000-php-frameworky/>

TAYLOR, Jay. 3 Reasons Why Responsive Web Design is the Best Option For Your Mobile SEO Strategy. In: *Search Engine Watch* [online]. 2013 [cit. 2014-04-25]. Dostupné z: <http://searchenginewatch.com/article/2253965/3-Reasons-Why-Responsive-Web-Design-is-the-Best-Option-For-Your-Mobile-SEO-Strategy>

PEŠOUT, Martin. Detekce mobilních zařízení. In: *Igloonet blog* [online]. 2011 [cit. 2014-04-25]. Dostupné z: <http://blog.igloonet.cz/detekce-mobilnich-zarizeni/>

VYLEŤAL, Martin. Mobile Internet FORUM 2013: uživatelé utěšeně rostou, následovat by měly peníze. In: *Lupa.cz* [online]. 2013 [cit. 2014-05-09]. Dostupné z: <http://www.lupa.cz/clanky/mobile-internet-forum-2013-uzivatele-utesene-rostou-nasledovat-by-mely-i-penize/>

W3C. Media Queries. In: *World Wide Web Consortium (W3C)* [online]. 2012 [cit. 2014-04-25]. Dostupné z: <http://www.w3.org/TR/css3-mediaqueries/>

WALTER, Stéphanie. Responsive Retina-Ready Menu. In: *Codrops* [online]. 2013 [cit. 2014-05-09]. Dostupné z: <http://tympanus.net/codrops/2013/05/08/responsive-retina-ready-menu/>

WROBLEWSKI, Luke. Mobile First. In: *LukeW* [online]. 2011 [cit. 2014-05-10]. Dostupné z: http://www.lukew.com/resources/mobile_first.asp

Seznam obrázků a tabulek

Seznam obrázků

Obr. 2.1: Grafické znázornění odpovědi na otázku „Vlastníte některé z těchto zařízení – Tablet, Smartphone, Netbook/ Mini notebook?“ (zdroj: autor)	16
Obr. 2.2.: Grafické znázornění odpovědi na otázku „Chcete si v budoucnu (zhruba do jednoho roku) pořídit některé z těchto zařízení?“ (zdroj: autor)	16
Obr. 2.3: Graf znázorňující odpověď na otázku: „Jaký internetový prohlížeč preferujete ve Vašem tabletu?“ (zdroj: autor)	17
Obr. 2.4: Graf znázorňující odpovědi na otázku: „Jste spokojeni s internetovým prohlížečem, který používáte ve vašem tabletu? (Stupnice jako ve škole)“ (zdroj: autor)...	18
Obr. 2.5: Grafické znázornění odpovědi na otázku: „Jaký internetový prohlížeč preferujete ve Vašem smartphonu?“ (zdroj: autor)	19
Obr. 2.6: Grafické znázornění odpovědi na otázku: „Jste spokojeni s internetovým prohlížečem, který používáte ve vašem smartphonu? (Stupnice jako ve škole)“ (zdroj: autor).....	19
Obr. 2.7: Grafické znázornění odpovědi na otázku: „Jaký internetový prohlížeč preferujete ve Vašem netbooku/ mini notebooku?“ (zdroj: autor)	21
Obr. 2.8: Grafické znázornění odpovědi na otázku: „Jste spokojeni s internetovým prohlížečem, který používáte ve vašem netbooku/ mini notebooku? (Stupnice jako ve škole)“ (zdroj: autor)	21
Obr. 4.1: Náhled na drátové modely pro tři úrovně responzivnosti – od nejmenší obrazovky po největší (zdroj: autor)	33
Obr. 4.2: Nahoře menu pro desktop, vlevo dole pro mobilní telefony, vpravo dole pro tablety a menší desktopy (zdroj: autor)	34
Obr. 4.3: Rozdíly mezi drátovými modely a samotným grafickým provedením (zdroj: autor).....	37
Obr. 4.4: Ukázka testování stránek v prostředí emulátoru Opera Mobile. (zdroj: autor).....	45
Obr. 4.5: Testování stránek na serveru browserstack.com. Nahoře v IE 9, dole v IE 8 (zdroj: autor).....	46
Obr. 4.6: Testování na serveru mediaqueri.es v Samsungu Galaxy SIII otočeným na šířku (zdroj: autor)	47
Obr. 4.7: Screenshoty z Internet Exploreru 11 v zařízení Nokia Lumia 720 (zdroj: autor).....	50

Seznam tabulek

Tab. 2.1: Porovnání přístupů k tvorbě webových stránek vhodných pro různá zobrazovací zařízení (zdroj: autor; Forgáč, 2013).....	12
Tab. 4.1: Zařízení zařazená do testování webových stránek (zdroj: autor).....	48
Tab. 4.2: Testování stránek v různých zařízeních a prohlížečích (zdroj: autor).....	49

Přílohy

Příloha A: Dotazník

1. Vlastníte některé z těchto zařízení - Tablet/ Smartphone/ Netbook (mini notebook)?

- a) Ano
- b) Ne

2. Používáte některé jiné zobrazovací zařízení, ve kterém lze prohlížet webové stránky? (Mimo tabletu, smartphonu a netbooku)

**Napište prosím celý název Vašeho zařízení. Pokud vlastníte více druhů takovýchto zařízení, můžete vypsát všechny. (Nemusíte odpovídat, pokud vlastníte pouze desktopový počítač nebo notebook.)*

3. Chcete si v budoucnu (zhruba do jednoho roku) pořídit některé z těchto zařízení?

- a) Smartphone
- b) Tablet
- c) Netbook/ Mini notebook
- d) Ne, nechci.

4. Víte, co se skrývá pod pojmem "responzivní design"?

- a) Ano
- b) Ne
- c) Nejsem si zcela jistý(á), ale něco málo už jsem o responzivním designu slyšel (a).

5. Máte pocit, že jsou responzivní webové stránky pro uživatele matoucí?

**Odpovídejte, jen pokud jste na předchozí otázku odpověděli „Ano“*

- a) Ano
- b) Ne
- c) Jiný.....

TABLET - Odpovídejte pouze v případě, že vlastníte tablet.

6. Jaký model tabletu vlastníte? Pokud jich vlastníte více, můžete vypsát všechny. Napište celý název Vašeho zařízení. Například: Google Nexus 10

.....

7. Jaký internetový prohlížeč preferujete ve Vašem tabletu?

- a) Dolphin
- b) Firefox
- c) Opera
- d) Chrome
- e) Safari
- f) Maxhton
- g) Boat Browser
- h) Dolphin
- i) Jiný

8. Jaký internetový prohlížeč preferujete ve Vašem tabletu? (Stupnice jako ve škole)

- a) Jednička
- b) Dvojka
- c) Trojka
- d) Čtyřka
- e) Pětka

SMARTPHONE - Odpovídejte pouze v případě, že vlastníte smartphone.

9. Jaký model smartphonu vlastníte? Pokud jich vlastníte více, můžete vypsát všechny. Napište celý název Vašeho zařízení. Například: HTC Desire 500

.....

10. Jaký internetový prohlížeč preferujete ve Vašem smartphonu?

- a) Dolphin/ Dolphin Mini
- b) Firefox
- c) Opera/ Opera Mini/ Opera Mobile
- d) Chrome
- e) Safari
- f) Maxhton
- g) Boat Browser
- h) Dolphin/ Dolphin Mini
- i) Jiný

11. Jaký internetový prohlížeč preferujete ve Vašem smartphonu? (Stupnice jako ve škole)
- a) Jednička
 - b) Dvojka
 - c) Trojka
 - d) Čtyřka
 - e) Pětka
12. Kde používáte nejčastěji svůj smartphone k prohlížení webových stránek?
- a) Doma
 - b) Ve škole/ v práci
 - c) Na cestách
 - d) Jinde

NETBOOK/ MINI NOTEBOOK - Odpovídejte pouze v případě, že vlastníte netbook/ mini notebook.

13. Jaký model tabletu vlastníte? Pokud jich vlastníte více, můžete vypsát všechny. Napište celý název Vašeho zařízení. Například: Google Nexus 10
-
14. Jaký internetový prohlížeč preferujete ve Vašem netbooku?
- a) Mozilla Firefox
 - b) Opera
 - c) Chrome
 - d) Internet Explorer
- Jiný
15. Jaký internetový prohlížeč preferujete ve Vašem netbooku/ mini notebooku? (Stupnice jako ve škole)
- a) Jednička
 - b) Dvojka
 - c) Trojka
 - d) Čtyřka
 - e) Pětka

16. Status

- a) Student VŠ
- b) Student SŠ
- c) Student ZŠ
- d) Zaměstnanec – plný úvazek
- e) Nezaměstnaný
- f) Jiné

17. Věk

- a) Méně než 15 let
- b) 15 – 20 let
- c) 21 – 25 let
- d) 26 – 30 let
- e) Více než 30 let

18. Pohlaví

- a) Žena
- b) Muž

Příloha B: Soubory k webové prezentaci

webova_prezentace.zip

Příloha C: HTML

HTML_xhatsz00_BP.pdf

Příloha D: CSS

CSS_xhatsz00_BP.pdf