

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

**Analýza CSS frameworků pro tvorbu responzivního
designu webových stránek**

Bc. Lukáš Kopecký

© 2018 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Lukáš Kopecký

Informatika

Název práce

Analýza CSS frameworků pro tvorbu responzivního designu webových stránek

Název anglicky

Analysis of CSS frameworks for creating responsive web design

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku tvorby responzivního designu webových stránek. Hlavním cílem práce je analýza CSS frameworků pro tvorbu responzivního designu. Dílčí cíle práce jsou:

- charakteristika trhu CSS frameworků,
- aplikace vybraného CSS frameworku,
- otestování vybraného CSS frameworku.

Metodika

Metodika řešení problematiky diplomové práce je založena na studiu a analýze odborných informačních zdrojů. Praktická část práce je zaměřena na analýzu CSS frameworků, následnou aplikaci a otestování vybraného nástroje. Na základě syntézy teoretických poznatků a analýzy v praktické části budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

Responsivní design, CSS, framework, webdesign, webové stránky, vícekritériální analýza variant

Doporučené zdroje informací

CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN 978-80-2513-733-8

GASTON, Peter. The book of CSS3: a developer's guide to the future of web design. San Francisco: No Starch Press, 2011. ISBN 978-1-59327-286-9

KADLEC, T. *Responsivní design profesionálně*. Brno: Zoner Press, 2014. ISBN 978-80-7413-280-3.

MILLS, Chris. Practical CSS3: develop and design. Develop and design. ISBN 978-0-32182-372-4

SHARKIE, Craig a Andrew FISHER. Responsivní webdesign: okamžitě. Brno: Computer Press, 2015. ISBN 978-80-2514-384-1.

Předběžný termín obhajoby

2017/18 LS – PEF

Vedoucí práce

Ing. Jan Jarolímek, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 30. 10. 2017

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 1. 11. 2017

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 25. 02. 2018

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Analýza CSS frameworků pro tvorbu responzivního designu webových stránek" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou v práci citovány a uvedeny v jejím závěru v seznamu literatury. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 11. března 2018

Poděkování

Rád bych poděkoval panu Ing. Janu Jarolímkovi, PhD. za odborné vedení, konzultace a pomoc při psaní této práce.

Analýza CSS frameworků pro tvorbu responzivního designu webových stránek

Analýza CSS frameworků pro tvorbu responzivního designu webových stránek

Souhrn

Diplomová práce je zaměřena na analýzu CSS frameworků určených především pro tzv. responzivní design a následnou aplikaci nejlépe hodnoceného kompromisního nástroje. Práce je rozdělena do dvou částí. První část popisuje teoretické principy problematik responzivního designu webových stránek a vícekritériální analýzy variant.

Část vlastní rozbor je orientovaná na charakteristiky jednotlivých CSS frameworků, zejména jejich předností a nedostatků. Následně je provedena vícekritériální analýza variant a komparace systémů metodami AHP a váženého součtu z hlediska kritérií: velikost frameworku, implementace, customizace, mřížka, zlomové body, navigace, responzivita obrázků, carousel, podpora prohlížečů a v neposlední řadě glyfy, konkrétně jejich počet. Následně je software, který je vyhodnocen jako nejlepší kompromisní varianta, aplikován a použit pro vytvoření smyšlených webových stránek Clubu C.

Po provedení analýzy jsou navzájem porovnány rezultáty autora dosažených metodami AHP a váženého součtu. Poté je provedena komparace výsledků dosažených řešitelem se závěry z jiných odborných zdrojů.

Summary

The diploma thesis is focused on the analysis of CSS frameworks and application of the best-rated compromise software. The diploma thesis is divided into two parts. The first part is theoretical which describes the theoretical principles of CSS frameworks and multi-criteria analysis variants.

The part is called own analysis and is focused on the characteristics of CSS frameworks, especially their strengths and weaknesses. Then solver performs multi-criteria analysis of

variants and comparison of systems. It is performed with methods AHP and a weighted sum. Variants are compared on the basis of the criteria size of frameworks, implementation, customization, grid system, breakpoints, navigation, responsive images, carousel, browser support and, last but not least, glyphs, specifically their number. The software, which is evaluated as the best compromise variant, is applied and used at the create of sample web site of Club C.

After the analysis are compared the results of methods AHP and weighted sum each other. Than they are compared the solver's results and results from other source.

Klíčová slova:

Responzivní design, CSS, framework, web design, webové stránky, vícekritériální analýza variant

Keywords:

Responsive design, CSS, framework, web design, web site, multi-criteria analysis of variants

Obsah

Obsah	8
1 Úvod	11
2 Cíl práce a metodika	12
3 Přehled řešené problematiky	13
3.1 Responzivní design webových stránek	13
3.1.1 Mobilní a responzivní web	14
3.1.1.1 Mobilní web	14
3.1.1.2 Responzivní web.....	16
3.1.1.3 Standardní web	17
3.1.1.4 Srovnání typu webů	18
3.1.2 Strategie tvorby designu webových stránek	19
3.1.2.1 Desktop first	19
3.1.2.2 Mobile first	20
3.1.3 Jednotky při designování webových stránek	20
3.1.3.1 Absolutní jednotky.....	21
3.1.3.2 Relativní jednotky	22
3.1.4 Layout.....	22
3.1.4.1 Typy layoutů.....	22
3.1.4.2 Mřížka.....	24
3.1.4.3 Fluidní mřížka.....	25
3.1.5 Dotazy na médium	26
3.1.5.1 Syntaxe.....	26
3.1.5.2 Typy médií.....	27
3.1.5.3 Vlastnosti médií.....	28
3.1.5.4 Zlomové body.....	29
3.1.5.5 Shrnutí dotazů na médium.....	31
3.1.6 Flexibilní obrázky	31
3.1.6.1 Relativní šířka	32
3.1.6.2 Značka <i>img</i> s pokročilými atributy	32
3.1.6.3 Značka <i>picture</i>	34
3.1.7 CSS preprocesory.....	35
3.1.7.1 Typy CSS preprocesorů.....	36
3.1.7.2 Použití CSS preprocesorů.....	37
3.1.7.3 Porovnání základní syntaxe CSS preprocesorů.....	39

3.1.8	CSS reset a CSS normalizace.....	40
3.1.9	Framework.....	40
3.1.9.1	Typy frameworků.....	41
3.1.9.2	CSS frameworky.....	41
3.2	Analýza CSS frameworků	42
3.2.1	Vícekriteriální analýza variant (VAV).....	42
3.2.1.1	Saatyho metoda.....	42
3.2.1.2	Metoda AHP	44
3.2.1.3	Metoda váženého součtu.....	44
4	Vlastní práce	46
4.1	Analýza CSS frameworků	46
4.1.1	Stanovení kritérií.....	46
4.1.2	Charakteristika variant dle kritérií.....	50
4.1.2.1	Bootstrap.....	51
4.1.2.2	Foundation	55
4.1.2.3	Skeleton	59
4.1.2.4	Kube	61
4.1.2.5	Materialize.....	63
4.1.3	Ohodnocení variant dle kritérií	66
4.1.4	Stanovení vah kritérií	66
4.1.5	Analýza metodou AHP	67
4.1.6	Analýza metodou váženého součtu	68
4.2	Aplikace vybraného nástroje.....	70
4.2.1	Implementace.....	70
4.2.2	Mřížka a zlomové body	71
4.2.3	Navigace.....	74
4.2.4	Carousel	76
4.2.5	Responzivita obrázků	78
4.3	Otestování vytvořené webové prezentace	78
5	Výsledky a diskuse.....	80
6	Závěr	83
7	Použitá literatura	85
8	Seznam obrázků, tabulek a příloh.....	90
8.1	Obrázky.....	90

8.2	Tabulky	91
8.3	Grafy.....	92
8.4	Přílohy.....	92
9	Přílohy.....	93

1 Úvod

Web je jedno z nejnestabilnějších a nejrychleji se měnících prostředí, kde mohou působit designéři a kreativci. Za posledních několik let se také v této oblasti neustále rozvíjí nové vývojářské techniky. Ty umožňují a vedou k novým stylům, trendům a návrhovým vzorům, které přecházejí v nesčetný počet webových stránek.

Po příchodu různých zařízení začaly narůstat požadavky uživatelů na weby z hlediska zobrazování na různých velikostech displeje. Zařízení používaná pro zobrazování webových stránek se samozřejmě měnila vždy, ale v posledních letech dochází k nárůstu rozsahu a především tempa, s jakým změny nastávají. Dříve stačilo pouze přizpůsobit stránky na šířku 1024 nebo 800 pixelů. Ovšem dnešní doba jde kupředu a nyní se při tvorbě webových stránek musí počítat nejen s klasickými desktopovými počítači a monitory v poměru 4:3, ale i s desktopy nebo notebooky s poměrem displeje například 16:9, televizory či telefony a tablety s různou orientací displeje. Po vstupu na trh všech těchto zařízení ovládaných dotykovými obrazovkami, trackpady apod. se stal nezbytný tzv. responzivní design.

Responzivní design je přístup k vývoji webových stránek umožňující přizpůsobení jejich zobrazení na různých zařízeních. Responzivní webová stránka je tedy univerzální, poddajná, flexibilní a schopna se adaptovat na jakoukoli obrazovku.

Pro tvorbu responzivní webové prezentace je možné použít předem připravenou softwarovou strukturu nazývanou CSS framework. Použití vhodného frameworku může velmi urychlit vývoj nového designu, zlepšit celkovou funkčnost a výkon celého webu.

2 Cíl práce a metodika

Primárním cílem diplomové práce je analýza CSS frameworků metodami vícekriteriální analýzy variant. Mezi sekundární cíle patří zmapování nabídky frameworků, jejich charakteristika a aplikace včetně otestování nástroje vybraného jako nejlépe hodnocená kompromisní varianta na základě analýzy.

Diplomová práce je rozdělena do tří hlavních částí. První část je tvořena teoretickými východisky získanými především z odborné literatury zaměřené na responzivní design a vícekriteriální analýzu variant, které jsou doplněny ověřenými internetovými zdroji. Tyto zdroje dat jsou uvedeny v použité literatuře na konci práce.

Analytická část mapuje trh CSS frameworků a jejich charakteristiku. Následuje analýza těchto nástrojů metodami vícekriteriální analýzy variant: AHP a váženého součtu. Pro samotné porovnání jednotlivých systémů jsou stanovena kritéria. Jejich samotná volba a stanovení vah kritérií je klíčovým okamžikem celého rozhodovacího procesu. Kritéria a jejich váhy jsou vyvozeny ze subjektivního postoje autora práce na základě pracovních zkušeností a konzultací s dvěma odborníky z praxe, kteří se zabývají designováním webových stránek. Váhy jsou konkrétně stanoveny Saatyho metodou párového porovnání. Hodnoty kvantitativních kritérií jsou vyhledávány ve volně dostupných zdrojích a kvalitativní jsou klasifikována na základě vlastních zkušeností řešitele s vybranými nástroji a důkladné studii dokumentací k daným nástrojům.

Na základě provedené vícekriteriální analýzy variant je vybrána nejlépe ohodnocená kompromisní alternativa z klasifikovaných CSS frameworků. Vybraný nástroj je následně aplikován pro tvorbu smyšlených ilustrativních webových stránek. Dále je provedeno testování chování jednotlivých možností napříč prohlížeči a zařízeními. Samotné testování je realizováno s použitím většiny obvykle používaných webových prohlížečů.

Na konci práce následuje vzájemné porovnání výsledků dosažených jednotlivými metodami. Následně jsou výsledky řešitele porovnány s recenzemi z jiných odborných zdrojů. Ze zjištěných poznatků jsou formulovány závěry, zda jsou výsledky řešitele a z jiných zdrojů shodné nebo odlišné.

3 Přehled řešené problematiky

V této části diplomové práce jsou rozebrány teoretické principy dvou problematik. Nejprve jsou popsána východiska tématu samotného responzivního designu webových stránek a CSS frameworky. Poté jsou charakterizovány vícekriteriální analýzy variant.

3.1 Responzivní design webových stránek

Zásluhou masivního šíření internetu je disciplína zabývající se designem webových stránek známa prakticky od začátku zmiňované počítačové sítě. Tento obor má už tedy za sebou několik let vývoje a postupem času se stal mezioborovou vědní oblastí s velmi rozsáhlou tematikou. Ta je složená především z interakce mezi člověkem a počítačem, komunikace, designu, copywritingu a jiných.

Za procesem působivého a osobitého designu webových prezentací stojí především několik základních konceptů, jimiž lze dosáhnout maximálního vlivu na uživatele. Tyto koncepty pak ve většině případů využívá každý designer a vylepšuje jimi určitou metodologii pro úpravu layoutu a struktury obsahu [1].

Na začátku tisíciletí se technologie stále vyvíjely a mobilní zařízení byla stále vyspělejší, ale internet byl dostupný jen na několika málo typech mobilních telefonů a to jen velmi primitivním způsobem. Dále se vyskytovalo mnoho dalších problémů, které přístup komplikovaly – velmi slabý hardware, pomalé mobilní připojení, malé obrazovky, těžkopádné metody vstupu a mobilní zařízení byla schopna interpretovat velmi zjednodušenou XML verzi webu nazývanou WirelessMarkupLanguage (WML). Zvrat nastal až v roce 2007 příchodem iPhone první generace s dotykovým displejem. Náhle bylo možné webové prezentace „plnohodnotně“ zobrazovat. Po vzoru Applu začaly i jiné společnosti vyrábět chytré telefony [2]. Ty se rozšiřovaly mezi veřejnost a uživatelé si stále více prohlíželi internet na přenosných zařízeních. Ovšem na nich se webové stránky navržené pro velké monitory zobrazovaly stejně jako na desktopech, což způsobovalo na malém displeji problémy při procházení, jelikož bylo nutné neustále přibližovat a skrolovat všemi směry. V té době se poprvé objevuje termín responzivní design.

Responzivní design se stará o co možná nejlepší uživatelský prožitek pro uživatele webové prezentace. Cílem je, aby pro návštěvníky bylo zapotřebí přibližovat a skrolovat obsah všemi směry co nejméně a to nezávisle na tom, zda používají desktop, notebook nebo menší přenosné zařízení, jako jsou chytré telefony a tablety [3]. Jinými slovy se dá tedy říct, že responzivní design obstarává, aby grafické rozhraní webových stránek bylo pro uživatele vždy uzpůsobeno typu zařízení, které je používáno, bez ohledu na rozlišení displeje. To je zajišťováno pomocí zmenšování nebo přeskupování jednotlivých elementů [4]. Schéma možného uzpůsobení obsahu pro různá zařízení je zobrazeno na obrázku č. 1.

Obrázek 1: Schéma responzivního designu



Zdroj: [5]

3.1.1 Mobilní a responzivní web

Web optimalizovaný pro všechna zařízení lze vytvořit dvěma, potažmo třemi, způsoby:

- a) **dva „různé weby**: mobilní a desktopový web,
- b) **jeden responzivní web**: jeden, který se přizpůsobuje velikosti displeje,
- c) **standardní web**: jedná se o klasický web, u kterého je design zaměřen na jednoduchost [6].

Obsah mobilního i responzivního webu je sice optimalizován pro mobilní zařízení, ale technologie obou těchto možností jsou odlišné. Zásadní rozdíl spočívá v tom, že u mobilního webu je nutnost mít dvě verze jedné webové prezentace, ale u responzivního pouze jednu [7].

3.1.1.1 Mobilní web

U této varianty se jedná o samostatně existující verzi těchto stránek, tedy pro jednu webovou prezentaci je nutné mít dvě odlišné verze – desktopovou a mobilní. Zpravidla

jsou obě formy na odlišných doménách, ale mají stejnou datovou základnu [8]. Druhá zmiňovaná verze webu je většinou na subdoméně domény, kde je umístěna desktopová. Tedy například desktopová verze společnosti Alza a.s. je umístěna na www.alza.cz (obrázek č. 2) a mobilní na m.alza.cz (obrázek č. 2) [6].

Obrázek 2: Desktopová (vlevo) a mobilní (vpravo) verze webu společnosti Alza a.s.



Zdroj: Vlastní zpracování na základě webové prezentace společnosti Alza a.s.

Princip samotného mobilního webu je založen na tom, že sama webová stránka detekuje na straně serveru typ zařízení, na jakém se momentálně uživatel nachází, a podle toho nechá zobrazit jednu nebo druhou formu. Obvykle se pak přidává i možnost přepínání mezi oběma verzemi [6].

Výhody mobilního webu

Nespornou výhodou mobilního webu je jeho rychlost načítání, která je vyšší [7], což je dáno tím, že se přenáší až 10x méně dat než u klasického webu a 5x méně než u responzivního. Dále je možné poskytovat přepnutí do klasického zobrazení [5] nebo odlišný obsah na desktopu a telefonu s možností zobrazení pouze části obsahu [7]. Mezi další výhodou lze zařadit optimalizaci pouze části obsahu, například jednu podstránku, což je sporná výhoda, jelikož toto řešení sice přinese úsporu v pořizovacích nákladech, ale celkový dojem z webových stránek pro uživatele nebude tak pozitivní [5].

Nevýhody mobilního webu

Hlavní nevýhodou je vyšší pořizovací cena, jelikož je nutné vyrobit dva weby s dvěma rozdílnými uživatelskými rozhraními. Poté je potřeba rozeznat, o jaké zařízení se jedná

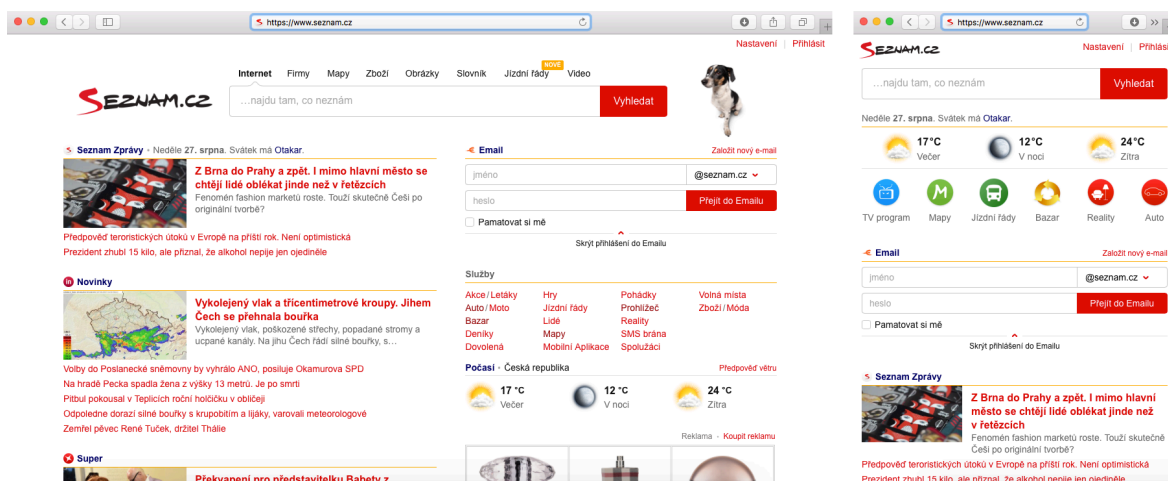
a nechat zobrazit pouze správnou formou webu [5]. To se ovšem ne vždy podaří správně, protože například i velký tablet je většinou rozpoznáván jako mobilní zařízení, ale na jeho velkém displeji nevypadá redukováný obsah pro telefony příliš dobře [7]. Nevýhodou se také může stát problematika ohledně SEO. Pokud jsou pro mobilní a desktopovou verzi použity různé domény nebo subdomény, mohou vznikat duplicity, což jsou dvě různé URL adresy se stejným obsahem [5], a takovéto weby nejsou příznivě hodnoceny vyhledávači [7].

3.1.1.2 Responzivní web

Na mobilních zařízeních se zobrazuje totožná verze webu jako desktopu s odlišným uživatelským rozhraním. Web na straně serveru tedy nerozlišuje, na jakém zařízení je právě uživatel, a jsou zasílána vždy stejná data. Ovšem tato stejná data, potažmo totožný web, se na mobilu, tabletu nebo notebooku a jiných zařízeních zobrazuje jinak v závislosti na úhlopříčce displeje, tedy velikosti zobrazované plochy [8].

Z výše uvedeného důvodu není nutné vytvářet žádnou druhou verzi webu a tím pádem ani vlastnit další doménu nebo subdoménu mimo základní pro desktopovou formu. Pro zobrazení verze webu optimalizovanou pro mobilní se tedy nezadává jiná adresa, ale je dostačující změnit šířku okna, a to i na běžném stolním počítači [5] (viz. obrázek č. 3).

Obrázek 3: Verze klasická (vlevo) a optimalizovaná pro telefony (vpravo) webu Seznam.cz



Zdroj: Vlastní zpracování na základě webu Seznam.cz

Výhody responzivního webu

Největší výhodou je, že tento typ webu umožňuje více úrovní responzivnosti. To znamená, že lze stránky přizpůsobit každému zařízení zvlášť v závislosti na velikosti displeje. Je tedy možné vytvořit rozdílný layout pro počítač, notebook, tablet na šířku, tablet na výšku, malé a velké telefony apod. [8]. Vzhledem k této skutečnosti se web automaticky adaptuje i na zařízení, která při jeho spuštění ještě neexistovala [5].

Jednou z dalších výhod je, že stačí vytvořit pouze jednu verzi webu pro všechna zařízení a tím pádem dostačuje jedna URL, což s sebou přináší pozitiva se spojením problematikou SEO, jelikož není nutné nijak hlídat duplicity [7].

Nevýhody responzivního webu

Přenáší se více dat po internetu než u čistě mobilního. Z toho vyplývá, že se stránky nenačítají tak rychle, ale i přesto je rychlost stále vyšší než u klasických webových prezentací nijak optimalizovaných pro mobilní zařízení [4].

Jelikož je pouze jeden web, prakticky není možnost přepínání mezi klasickou a mobilní verzí. Proto je nutné uživatelské rozhraní mobilních verzí navrhnout opravdu dobře, protože pokud se v něm uživatel dobře neorientuje a nemá možnost přepnutí na klasické zobrazení, s největší pravděpodobností stránky opustí [5]. Toto je poměrně velká nevýhoda oproti mobilním webům. U nich má uživatel možnost přepnout na klasické zobrazení, pokud grafické rozhraní pro telefony návštěvníkovi nevyhovuje.

3.1.1.3 Standardní web

Jedná se o klasický web, který není nijak optimalizovaný ve smyslu mobility nebo responzivity. Zobrazuje se tedy naprosto stejně na všech zařízeních bez ohledu na velikost či rozlišení displeje. Optimalizace tohoto typu webu tedy spočívá v tom, že je navrhnout tak, aby byl plně funkční nejen na desktopu, ale i tabletech nebo mobilech. U těchto stránek je kladen důraz na jednoduchost, funkčnost a použitelnost [5].

Výhody standardního webu

Hlavními výhodami je nižší cena než u předchozích dvou řešení a jednotné uživatelské rozhraní na všech zařízeních. Návštěvníci si tedy nemusejí zvykat na jiné uspořádání

a například hledat tlačítka na různých verzích. Mají jistotu, že se dané tlačítko nebo funkce vyskytuje vždy na stejné pozici [5].

Nevýhody standardního webu

I přes jednoduchost je poměrně vysoká pravděpodobnost nutnosti zvětšování zobrazení a skrolování všemi směry. Další nevýhodou je objem dat, který je přenášen. To je naprosto stejné na desktopu i telefonu. Tato skutečnost má za následek nižší rychlost načítání a poměrně vysoký úbytek dat u mobilního připojení poskytovaného jedním z telefonních operátorů. Toto je patrně největší nevýhoda oproti předchozím řešením, kde se u mobilní verze webových stránek přenáší až desetkrát méně dat [5].

3.1.1.4 Srovnání typu webů

V tabulce č. 1 je uvedeno srovnání několika základních parametrů výše uvedených jednotlivých řešení vytváření webových prezentací optimalizovaných pro přenosná zařízení.

Tabulka 1: Srovnání základních parametrů jednotlivých řešení tvorby webových stránek

X	Mobilní web	Responzivní web	Standardní web
pořizovací cena	nejvyšší	střed	nejnižší
rychlost načítání	nejrychlejší	střed	nejpomalejší
optimalizované uživatelské rozhraní pro mobilní zařízení	ano	ano	ne
řeší různé šířky displeje	ne	ano	ne
řeší i zařízení, která v čase výroby webu ještě neexistovala	ne	ano	ne
možnost přepnutí na klasickou verzi webu	ano	ne	—
všude stejné rozhraní (návštěvník si nemusí zvykat na jiné uspořádání)	ne (ale je možnost přepnutí na klasické zobrazení)	ne	ano

Zdroj: Vlastní zpracování dle [5]

Dle výsledků tabulky č. 1 je patrné, že každé z jednotlivých řešení má svá pozitiva i negativa, ale dle názoru autora by se měl standardní web používat jen pro jednoduché projekty. Moderní náročnější webové prezentace je vhodnější řešit mobilním nebo responzivním webem. Avšak druhý jmenovaný způsob tvorby je dle autora vhodnější, jelikož zohledňuje i šířku displeje a tím dokáže odlišit i velké tablety od malých telefonů

včetně orientace obrazovky. Proto se řešitel bude v následujících pasážích této práce věnovat preferovanému responzivnímu webu.

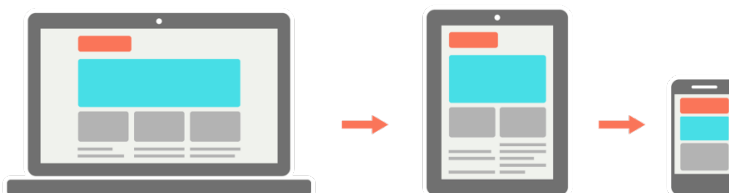
3.1.2 Strategie tvorby designu webových stránek

Existují dvě dominantní strategie pro tvorbu responzivních webových prezentací. Nazývají se desktop first a mobile first. Obě mají svá negativa i pozitiva, jež je potřebné zvážit při rozhodování, kterou z nich použít [9].

3.1.2.1 Desktop first

Strategií desktop first jsou nejdříve designovány webové stránky pro stolní počítače. Poté jsou přizpůsobeny pro zobrazení na menších obrazovkách tabletů, mobilů a jiných přenosných zařízeních [10] (viz. obrázek č. 4).

Obrázek 4: Schéma strategie desktop first



Zdroj: [11]

Základem této strategie je nejdříve vytvoření stylu pro velké obrazovky [9]. Poté je kód CSS souboru doplněn o dotazy na médium (viz. podkapitola 3.1.5 Dotazy na médium), které v případě, že je okno obrazovky menší než zvolená šířka, upraví stránku na požadované zobrazení pomocí „přepsáním“ CSS příkazu na nový [11]. V tomto případě je tedy podmínka šířky řešena pomocí maxima (níže viz. ukázka kódu).

```
.carousel {  
margin-bottom: 50px;  
}  
  
@media screen and (max-width: 768px) {  
    .carousel {  
margin-bottom: 30px;  
    }  
}
```

3.1.2.2 Mobile first

Strategie mobile first je pravým opakem výše popisovaného přístupu. Nejdříve jsou tedy designovány webové stránky pro nejmenší zařízení, nejčastěji pro mobilní telefony. Až poté jsou vytvářeny vzestupně pro zařízení větších rozměrů jako tablety, notebooky, desktopy apod. [10] (obrázek č. 5).

Obrázek 5: Schéma strategie mobile first



Zdroj: [11]

Na níže uvedeném příkladu je vyobrazen základní princip CSS této strategie. Základní myšlenka je stejná jako u předchozího případu, ale v opačném postupu. Nejdříve je stylován malý displej [9] a poté pomocí mediálních dotazů jsou designována zařízení s většími displeji. Zde je používána podmínka minima, protože je stránka upravena pouze v případech větší hodnoty obrazovky než zvolená šířka [11].

```
.carousel {  
margin-bottom: 30px;  
}  
  
@media screen and (min-width: 768px) {  
    .carousel {  
margin-bottom: 50px;  
    }  
}
```

3.1.3 Jednotky při designování webových stránek

U designování webových stránek lze použít poměrně velké spektrum různých jednotek, které lze rozdělit do dvou základních kategorií – absolutní a relativní jednotky. Při každé tvorbě webů je velice důležité používat vhodně jednotky pro určité prvky a u responzivního designu to platí dvojnásobně.

3.1.3.1 Absolutní jednotky

Pixel je základní absolutní jednotkou, kterou lze používat ve webovém designu. Zapisována je jako px. V oblasti responzivního designování je nutné rozlišit tzv. CSS pixel a HW (skutečný, hardwarový) pixel [12].

V minulých letech, kdy sloužily k zobrazování webových prezentací pouze desktopy na obrazovkách s téměř totožnými vlastnostmi, nebyl žádný problém tuto jednotku používat, jelikož dva druhy pixelů zmiňované výše se rovnaly. S příchodem telefonů s ultra jemnými displeji, například Retina nebo Super AMOLED, se ale situace zkomplikovala, protože jeden CSS pixel se může vykreslovat až třemi nebo čtyřmi skutečnými. U kaskádových stylů se až na výjimky pracuje s CSS pixely [13], zejména v případě dotazů na médium (viz. podkapitola 3.1.5 Dotazy na médium), což je jeden z pilířů responzivního designu. V praxi to znamená, že například pro Retina displej (poměr HW pixelů : CSS pixelů = 2 : 1) o šířce 1000 pixelů se u dotazů na médium pracuje s velikostí 500 pixelů. Pokud je třeba pracovat se skutečnými pixely, je potřeba navíc použít atribut *device-pixel-ratio* s hodnotou 2. Poté je CSS rozlišení 500 pixelů vynásobeno dvěma, a je tedy pro další používání vrácena hodnota 1000 pixelů. Tabulka příkladů poměrů mezi HW a CSS pixely je uvedena na obrázku č. 6 [14].

Obrázek 6: Poměr mezi HW a CSS pixely

Poměr mezi HW a CSS rozlišením u vybraných zařízeních			
Zařízení	HW rozlišení	CSS rozlišení	device-pixel-ratio
iPad Mini	768 × 1024	768 × 1024	1
iPhone 4	640 × 960	320 × 480	2
Google Nexus 7	800 × 1280	604 × 966	1,325
HTC One	1080 × 1920	360 × 640	3
Xiaomi Mi3	1080 × 1920	270 × 480	4

Zdroj: [14]

Ostatní absolutní jednotky jsou spíše doplňkové. Jedná se o klasické jednotky z reálného světa, například tedy milimetr, centimetr nebo palec. Ovšem u responzivního designu jsou prakticky nepoužitelné. Vhodné použití mají pro tiskový výstup z webových stránek [12].

3.1.3.2 Relativní jednotky

Relativních jednotek je používáno několik, ale nejčastěji se aplikují jednotky procenta a eM, která se zapisuje jako em. Mění svou velikost v závislosti na aktuální velikosti písma. Praktické využití může být například u obtékání obrázků – v případě nastavení výšky rastru na 5em bude vždy vysoký jako 5 řádků aktuálního textu [2].

Procenta fungují obdobě jako jednotka eM s tím rozdílem, že se nemění podle aktuální velikosti textu, ale v závislosti na rozměrech elementu, ve kterém je zpracováván objekt umístěn [15].

3.1.4 Layout

Ve většině případů je tělo webu rozděleno do několika sekcí. Ty rozvrhují stránku a tvoří tzv. layout. V minulosti se k jeho tvorbě využívaly tabulky, ale postupem času se začaly komponovat pomocí tagů neboli značek s názvem *div*. Avšak s příchodem HTML5 jsou nahrazovány novými značkami, které při kompozici dávají každé části význam. Například značka *nav* je označení pro sekci navigace neboli menu [16].

3.1.4.1 Typy layoutů

Ve své podstatě se můžeme setkat se čtyřmi základními kategoriemi layoutů. Jedná se o fixní, plovoucí, elastické a hybridní layouty. Každý z těchto přístupů nabízí silné stránky, ale také různá omezení [17].

Fixní layout

U fixního layoutu je web omezen na konkrétní šířku v pixelech, a je tedy neresponzivní. Z toho vyplývá, že pokud jsou například webové stránky široké 960 pixelů, ale uživatel má nastavenou menší velikost okna prohlížeče, zobrazí se pouze jejich část a posuvník k vodorovnému skrolování [17].

Naopak při prohlížení webové prezentace na mnohem větším monitoru, například širokém 1920 pixelů, přinese tato stránka pro uživatele mnoho prázdného nepraktického místa, které nepůsobí příliš dobře.

Tento způsob se používal především v minulých letech, kdy byly webové stránky zobrazovány zejména na desktopech. Když designér přesně zná šířku webu, může tvořit

graficky propracované návrhy. V posledních letech je ovšem tato strategie na ústupu vzhledem k nástupu mobilních zařízení, jelikož zde prakticky neexistuje možnost tvorby responzivního designu [3].

Plovoucí layout

Plovoucí layout je mnohem flexibilnější než výše charakterizovaný způsob, jelikož rozměry se neurčují absolutně v pixelech, ale relativně v procentech. Není tedy podstatné, zda uživatel působí na desktopu, notebooku s širokým displejem nebo tabletu či telefonu s mnohem menší velikostí obrazovky, jelikož web se dokáže adaptovat. Vzhledem k této skutečnosti je zde dostačující ochrana proti horizontálnímu posuvníku nebo nevyužitému prázdnému prostoru [2].

Vzhledem k dovednosti adaptace je tato strategie používána jako základ pro tvorbu responzivního designu webových prezentací. Další velice přínosná pozitivní vlastnost je především jednodušší implementace mediálních dotazů a stylů pro optimalizaci na různá zařízení [2].

Elastický layout

Elastický layout je svým způsobem nástavbou plovoucího, který posunuje o stupeň výše. Zásadní rozdíl je, že omezení se určují pomocí relativní jednotky em. Ta je rovna momentálně definované velikosti písma v pixelech. Pokud je tedy například v těle webu velikost písma 16 pixelů, pak je 1em roven 16 pixelům a 2em rovno $2 \times 16 = 32$ pixelů [2].

Tento způsob s sebou přináší poměrně dobré řízení typografie u webových stránek, například v oblasti čitelnosti textu [18]. Ten je nejlépe čitelný, pokud je délka řádku mezi 45 a 70 znaky. Po definování šířky kontejneru na hodnotu, která leží v intervalu 45 až 70, je zabezpečeno, že řádek má vždy patřičnou délku [3].

Hybridní layout

Hybridní layout kombinuje dva nebo více výše jmenovaných způsobů. Nejčastěji je využíván, pokud je nutné umístit na webové stránky s responzivním designem prvky s pevnou šířkou. Ovšem vzhledem k pevnému definování šířky jednotlivých sekcí se mohou vyskytovat negativa, která se objevují u fixního layoutu, a samotný responzivní design pak ztrácí smysl [2].

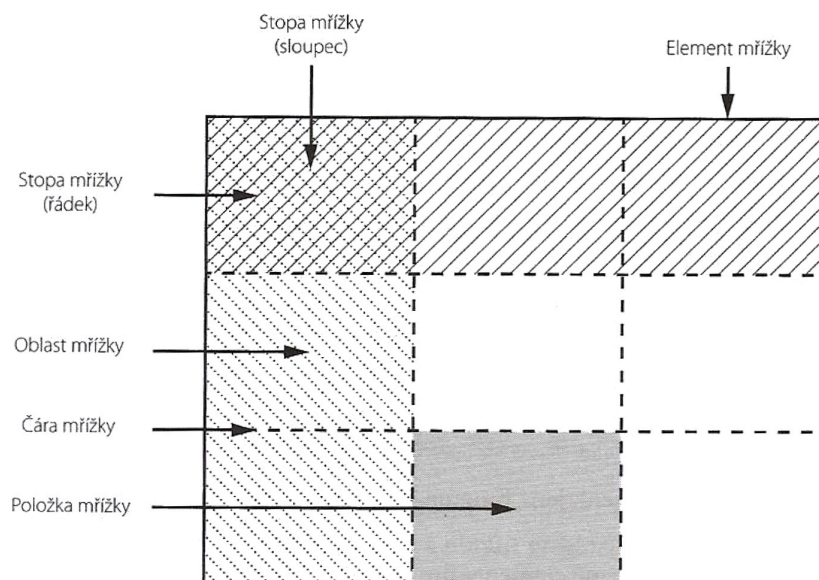
3.1.4.2 Mřížka

Mřížka je základní stavební kámen při návrhu a samotném designování bez ohledu na to, jaký layout je při tvorbě využíván. Mřížka je využitelná především pro vytvoření přehledného designu. Opakem je náhodnost. Náhodné uspořádání elementů může být zajímavé, ale je složité navrhnout kvalitní design. Strategie náhodnosti tedy není vhodná pro logicky organizované webové stránky zaměřené na informace [19].

Mřížka se používá prakticky pro návrhy v každém grafickém odvětví. Jednoduché mřížky písaři používali už ve středověku, ale moderní typografická forma se vyskytuje až od druhé poloviny 20. Století [18]. Jedná se tedy o velmi efektivní nástroj, který vychází z několika tisíc let známých obecných zákonitostí, jakými jsou například zlatý řez nebo pravidlo třetin [19].

Tento nástroj je v grafice používán zejména k systematickému zarovnání objektů, jelikož myšlení člověka je postaveno především na tom, že se lidský mozek snaží zjednodušovat vše, co vnímá pro lepší čitelnost. A právě z toho důvodu je mřížka používána, jelikož je natolik přehledná, že mozek velice rychle rozpozná systém zarovnání a identifikuje daný vzor [19].

Obrázek 7: Mřížka (3x3) se základními pojmy



Zdroj: [18]

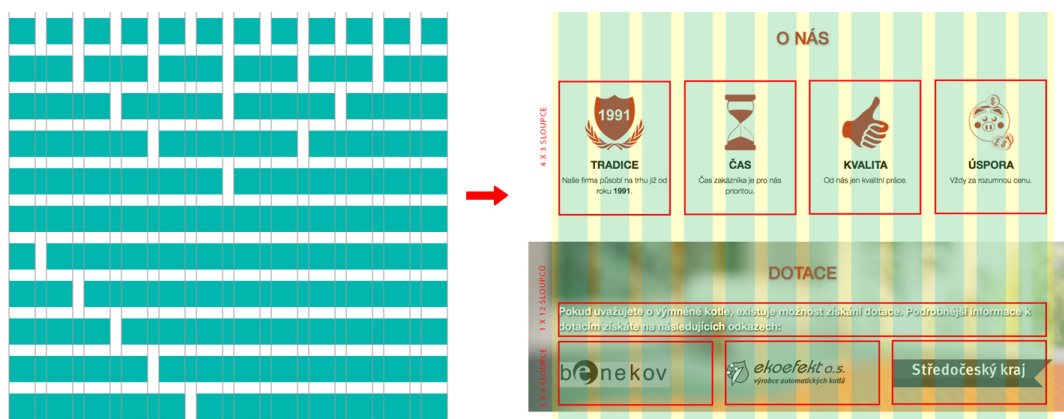
Na obrázku č. 7 je zobrazena jednoduchá názorná mřížka o rozměru 3x3 s vyobrazenými základními klíčovými pojmy:

- d) **Element mřížky** definuje hranice a rozměry mřížky,
- e) **položky mřížky** jsou jednotlivé dceřiné elementy,
- f) **čára mřížky** označují hranice mezi řádky a sloupci,
- g) **stopy mřížky** je souhrnné označení pro řádky i sloupce a
- h) **oblast mřížky** je každé protnutí sloupce s řádkem [18].

3.1.4.3 Fluidní mřížka

Fluidní mřížka je jedním z typů mřížek, tvoří součást plovoucího layoutu a je prvním pilířem responzivního designu webových stránek [8]. Nejjednodušší mřížky mohou být tvořeny pouze jedním, nebo třemi sloupci, ale pro responzivní design jsou spíše používány složitější mřížky. Ty obsahují více sloupců (9, 12, 15 atd.). U těchto mřížek mají sloupce funkci tzv. registračních bodů, které slouží pro rozvrhování obsahu řádku. U dvanáctisloupcové je tedy možné do jednoho řádku zahrnout všech 12 sloupců, vytvořit tři oblasti v sloupcích 1, 4 a 8 či vybudovat čtyři oblasti nebo zhotovit zcela jinou odlišnou kombinaci [3] (obrázek č. 8).

Obrázek 8: Schéma mřížky (vlevo) a použití mřížky (vpravo)



Zdroj: Vlastní zpracování

Pro responzivní design se používá především dvanáct sloupců. Tato struktura se využívá i ve většině CSS frameworků, jelikož toto číslo je dobře dělitelné, a tak je možné vytvářet dvě, tři, čtyři nebo šest rovnoměrně rozvržených oblastí, což je ideální pro kombinaci desktopu, tabletu a telefon. Například pro desktop se vytvoří čtyři třísloupcové, pro tablet tři čtyřsloupcové a pro telefon dvě šestisloupcové oblasti. (obrázek č. 9)

Obrázek 9: Plovoucí mřížka pro responzivní design webu – vpravo desktop, uprostřed tablet, vlevo telefon



Zdroj: Vlastní zpracování

Oproti klasické mřížce, která obsahuje pevné šířky, má fluidní mřížka výhodu flexibilních rozměrů. Fluidní mřížky se skládají ze sloupců a mezer mezi nimi. Rozměry obou jmenovaných veličin jsou určovány relativně vzhledem k obalujícímu elementu. Například u dvanáctisloupcové mřížky je šířka jednoho sloupce 8,33%. Z toho vyplývá, že na desktopu, jehož obalující element má šířku například 1400 pixelů, je absolutní hodnota jednoho sloupce 116,7 pixelů, ale na telefonu, jehož obalující element má rozměr kupříkladu 600 pixelů, je sloupec široký 50 pixelů. Tato flexibilita je patrná na obrázku č. 9. Ve své podstatě tedy finální rozměr webu určí sama mřížka tak, aby se webová stránka zobrazila ideálně na zařízení uživatele [3].

3.1.5 Dotazy na médium

Dotazy na médium jsou druhým pilířem responzivního designu. Používá se především ke změně webové prezentace na základě vlastností výstupních zařízení. Dotaz na médium je ve své podstatě logicky výraz, jenž je pravdivý či nikoli [8]. Pomocí zmiňovaného principu je možné detekovat, mimo jiných vlastností, velikost obrazovky, která byla dříve zjišťována pomocí JavaScriptu, a webové stránky optimalizovat pro dané zařízení [20].

3.1.5.1 Syntaxe

Níže pod odstavcem je umístěna ukázka kódu dotazu na médium. Tento příkaz zajistí změnu šířky na 50 pixelů u prvku s přiřazenou třídou *adv*, pokud je splněna podmínka, že uživatel používá výstupní zařízení s obrazovkou, jež má minimální šířku 320 pixelů a maximální šířku 480 pixelů.

```

@mediascreen and (min-width: 320px) and (max-width: 480px) {
    .adv {
        width: 50px;
    }
}

```

Na obrázku č. 10 je patrné schéma syntaxe dotazu na médium. K vyhodnocení, zda je logický výraz pravdivý či nikoli, je standardně používáno klíčové slovo *and* pro klasifikaci více podmínek současně nebo klíčový znak, (čárka) pro samostatné vyhodnocování. Například dotaz na médium *screen and (min-width: 800px)* je pravdivý pouze v případě použití počítače s barevnou obrazovkou o minimální šířce 800 pixelů, ale dotaz *screen, (min-width: 800px)* je pravdivý při splnění alespoň jedné z podmínek. To znamená, že stačí používat počítač s barevnou obrazovkou nebo jakékoli zařízení s minimální šířkou 800 pixelů [15].

Obrázek 10: Schéma syntaxe dotazů na médium

```

          typ média      vlastnost média      bod zlomu
          |              |              |
@media screen and (min-width: 320px) {
    ....
}

```

Zdroj: Vlastní zpracování

3.1.5.2 Typy médií

Typ média určuje, pro jaký druh zařízení je dotaz na médium určen. Nejpoužívanější jsou *screen* a *print*. První jmenovaný vymezuje barevné obrazovky počítačů a druhý je tiskový výstup [21]. Ostatní typy médií i s popisem jsou uvedeny v tabulce č. 2.

Tabulka 2: Typy médií

Typy médií	Popis
screen	pro barevné obrazovky počítačů
print	pro tiskový výstup
aural/speech	pro výstup na hlasových syntézách
braille	pro doteková zařízení s podporou braillova písma
embossed	pro tiskárny s podporou braillova písma
projection	pro projektory
tty	pro terminály a jiná zařízení s pevně definovou mřížkou a omezenou kapacitou zobrazování
tv	pro televizory
all	pro všechny typy zařízení
handheld	pro přenosná zařízení (ve zařízení s malou obrazovkou)

Zdroj: Vlastní zpracování dle [21]

3.1.5.3 Vlastnosti médií

Pomocí vlastností medií je možné zjišťovat různé parametry výstupních zařízení, například velikost obrazovky, rozlišení displeje apod. [22]. Všechny vlastnosti i s popisem a možností minimalizačních/maximalizačních prefixů jsou uvedeny v tabulce č. 3.

Tabulka 3: Vlastnosti médií

Vlastnosti médií	Popis	Možnost prefixů MIN/MAX
width	šířka okna prohlížeče	ANO
height	výška okna prohlížeče	ANO
device-width	šířka obrazovky výstupního zařízení	ANO
device-height	výška obrazovky výstupního zařízení	ANO
orientation	orientace displeje na výšku nebo šířku	ANO
aspect-ratio	poměr stran výšky a šířky okna prohlížeče	ANO
device-aspect-ratio	poměr stran výšky a šířky výstupního zařízení	ANO
color	počet bitů na barvu jednoho pixelu	ANO
color-index	počet prvků ve vyhledávací tabulce barev výstupního zařízení	ANO
monochrome	počet bitů pro jeden pixel uvnitř vyrovnávací paměti monochromatických snímků	ANO
resolution	rozlišení displeje	ANO
scan	skenovací proces pro televizory	NE
grid	ke zjištění, jestli zařízení používá mřížky nebo bitmapy	NE

Zdroj: Vlastní zpracování dle [22]

Jak je patrné v tabulce č. 3, u několika vlastností se může objevit prefix *device*, například *device-width* a *width*. Rozdíl mezi těmito parametry je, že prvně jmenovaný zkoumá velikost obrazovky výstupního zařízení a druhý jmenovaný velikost okna prohlížeče [22] (obrázek č. 11).

Obrázek 11: Rozíl mezi vlastnostmi *device-width* a *width*



Zdroj: Vlastní zpracování

Na mobilních zařízeních jsou ve finálním zobrazení webové stránky téměř vždy identické v obou případech, jelikož mobily a tablety většinou postrádají možnost manipulace s velikostí okna aplikace. Velikost displeje a okna prohlížeče je proto stejná (obrázek č.

11). Ovšem změna nastává na desktopech, kde je možné manipulovat s okny. Když je použita v dotazu na médiu vlastnost *widht*, webové stránky se chovají responzivně i na desktopech při zmenšení okna prohlížeče a tím pádem je možnost zobrazit mobilní verzi webové prezentace i na velkém monitoru osobního počítače (obrázek č. 11). Tato možnost je vyloučena při použití parametru *device-widht*. V tomto případě se bere v potaz velikost displeje, a tudíž se při zmenšení okna prohlížeče stránky nepřizpůsobují, což má za následek nutnost horizontálního skrolování. Tento jev je nežádoucí, a proto má ve své podstatě praktické využití pouze vlastnost *widht*. Parametry s prefixem *device* začínají být odebírány z webových standardů a ukončuje se jejich podpora moderními webovými prohlížeči [22].

3.1.5.4 Zlomové body

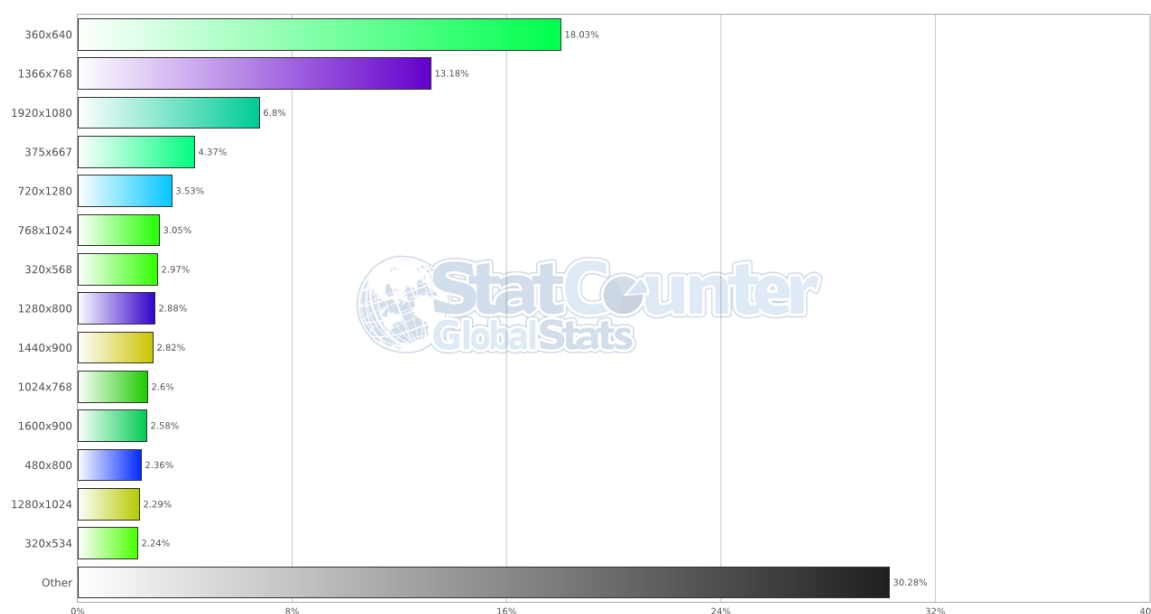
Zlomové body neboli breakpointy jsou zvolené hodnoty vlastností medií. Nejčastěji jsou používány pro detekci velikosti displeje na šířku parametrem *widht* pomocí CSS rozlišení. Důležitá informace je, že pro dotazy na médium jsou používány CSS pixely a nikoli HW pixely zařízení. Zvolenými body je možné omezit hraniční rozměry používané pro vymezení prostoru, kde má být přiřazen příslušný styl a webová prezentace má vypadat identicky na zařízeních spadávajících do definované oblasti [23].

Často vedou designéra, aby byly používány k detekci zařízení – nejčastěji telefony, tablety a desktopy. Ovšem tento postup není příliš spolehlivý, jelikož pomocí breakpointů s definovanými hodnotami nelze rozpoznat zařízení s velmi odlišnými způsoby používání. Například šířku displeje 700 pixelů může mít chytrý telefon větších rozměrů orientovaný na šířku (landscape), ale i menší tablet při orientaci na výšku (portrait). Tento případ by nepůsobil příliš velké komplikace, jelikož se jedná o dotyková zařízení, ale překážky nastávají především u velkých tabletů. Ty mohou mít šířku v režimu landscape stejnou jako menší notebooky, což může působit problémy, zejména pokud je definované u oblasti webové aplikace odlišné ovládání pro dotykové obrazovky než pro klávesnice a myš [24].

Důležitou otázkou při návrhu responzivního webu je: „Kolik a jaké zvolit zlomové body?“. Tento dotaz nemá jednoznačnou odpověď. Většina webových stránek potřebuje individuální přístup a zvolení breakpointů, ale na základě počtu nejčastěji používaných jednotlivých zařízení k zobrazení webových stránek lze alespoň vyčíst, v jakém rozmezí by se měly zlomové body ideálně vyskytovat [24].

V grafu č. 1 je vyobrazeno v procentuální míře 14 nejčastěji používaných rozlišení displejů různých zařízení, na kterých byly prohlíženy webové prezentace. Z grafu je patrné, že nejvíce jsou využívány zařízení s rozlišením 360x640 pixelů, pravděpodobně chytré telefony s 18% [25].

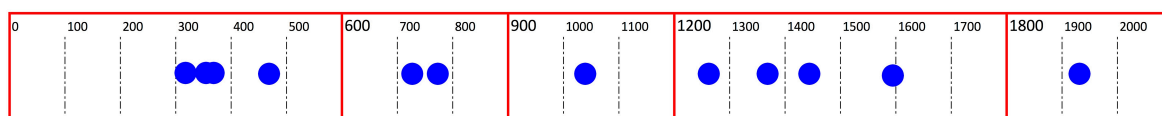
Graf 1: Procentuální poměr zařízení, na kterých jsou zobrazovány webové stránky



Zdroj: [26]

Pokud se zakreslí šířky displejů z grafu č. 1 do osy, jako na obrázku č. 12, je možné dle jejich rozvržení zjistit orientačně ideální umístění zlomových bodů. Na obrázku je patrné, že na základě nejpoužívanějších zařízení by měly být breakpointy zvoleny přibližně v intervalech 500px až 700px, 800px až 1000px, 1100px až 1250px a 1600 až 1900px. V grafu jsou zlomové body zakresleny v přibližném středu jednotlivých intervalů, tedy 600px, 900, 1200px a 1800px. Poslední jmenovaný zlomový bod je používán pouze, pokud je v plánu odlišné rozvržení stránek na extra velkých monitorech. Ovšem v praxi se příliš nepoužívá, jelikož od velikosti 1200 pixelů je layout většinou stejný [25].

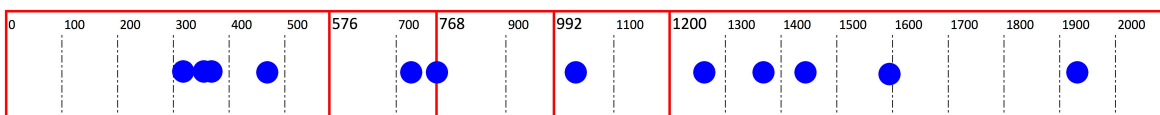
Obrázek 12: Ideální rozložení zlomových bodů



Zdroj: Vlastní zpracování

Skutečnost z předchozího odstavce je potvrzena i na obrázku č. 13, kde jsou vyobrazeny výchozí zlomové body CSS frameworku Bootstrap verze 4. Jak je patrné, zvýrazněné breakpointy 576, 768, 992 a 1200 jsou přibližně stejné jako na obrázku č. 12. Pouze je zde o jeden zlomový bod více na hodnotě 768 pixelů [27].

Obrázek 13: Zlomové body CSS frameworku Bootstrap



Zdroj: Vlastní zpracování dle [27]

3.1.5.5 Shrnutí dotazů na médium

Pomocí dotazů na médium je možné téměř jakkoli změnit layout webové prezentace na základě vlastností výstupního zařízení. Kupříkladu je možné u webu, který používá dvousloupcový layout, dát sloupce pod sebe nebo jeden z nich odstranit pro opravdu malá zařízení, a tak nechat pouze důležitý obsah. Ovšem nepracuje se pouze se samotným rozložením stránky. Pomocí dotazů na médium je možné také měnit i velikost obrázku a jiných elementů, aby byly v souladu s velikostí displejem výstupního zařízení uživatele. Tato flexibilita umožňuje optimalizovat web prakticky pro jakoukoli velikost obrazovky, které mohou uživatelé na svých zařízeních mít, a současně neomezovat funkčnost, použitelnost a přístupnost [28].

3.1.6 Flexibilní obrázky

Adaptivní obrázky jsou po fluidní mřížce a dotazech na médium třetím a posledním pilířem responzivního designu. Ty reagují na různé rozměry obrazovek. Tento pilíř tedy poskytuje obrázky v nejlepším možném zobrazení bez ohledu na velikost displeje zařízení [8].

Základním problémem, který tento druhý pilíř řeší, je, že pokud má webová prezentace flexibilní rozvržení (používá fluidní mřížku), musí být i její obsah adaptivní. U textu toho lze dosáhnout velice snadno, ale u grafických elementů je situace složitější. Pro řešení tohoto problému existuje několik postupů: relativní šířka, značka *img* s pokročilými atributy, značka *picture* [3].

3.1.6.1 Relativní šířka

Jedná se o základní a nejjednodušší řešení ohledně flexibility obrázků. V kaskádových stylech jsou nastaveny vlastnosti: šířka na 100% a výška jako automatická. Tento postup sice zajistí adaptivitu, ale prohlížeč vždy stahuje celý obrázek a až poté ho zmenšuje, což je nevýhoda z pohledu čerpání dat, jelikož i na telefon nebo tablet je obrázek stahován v příliš velké nepotřebné kvalitě [18].

První možností pro řešení problému uvedeného výše je uložení obrázku na server ve více velikostech a následné odeslání uživateli v určité kvalitě v závislosti na velikosti obrazovky zařízení. Druhou možností je použití vektorové grafiky [18].

Vektorová grafika ne webu

Obrazy křivkové neboli vektorové grafiky jsou tvořeny křivkami a geometrickými útvary pomocí vektorů, což jsou matematicky definované křivky. Výsledný obrázek je tedy ukládán pouze jako matematický zápis. Z toho plyne malá náročnost na paměťový prostor a nezávislost na rozměrech grafického objektu, z čehož vyplývá, že je lze zmenšovat i zvětšovat bez ztráty kvality a změny velikosti souboru [29].

Nejvíce používaným vektorovým formátem pro webové stránky je SVG. Tento formát je v reálu XML dokument, ve kterém jsou matematickým zápisem definovány vektory. Zásadou této skutečnosti je obrázek možné editovat i v poznámkovém bloku a může být indexován vyhledávači, jelikož je strojově čitelný. Další výhodou formátu SVG je možnost editace pomocí kaskádových stylů nebo JavaScriptu. Nevýhodou SVG a všech jiných vektorových formátů je použitelnost pouze pro ilustrované obrázky, což jsou loga a jiné „kreslené“ prvky [18].

3.1.6.2 Značka *img* s pokročilými atributy

Značka *img* je součástí kódování webových stránek téměř od samého začátku, ale příchodem HTML5 poskytuje nové atributy vhodné především pro adaptaci obrázku různým šířkám displeje.

Atribut srcset

Srcset je nový atribut v tagu *img*. Umožňuje prohlížeči stažení a zobrazení jiného obrázku, než je uvedeno v atributu *src*. V responzivním designu lze tuto funkci využít k nahrání

rastru ve více verzí na server a poté ho stahovat v určité formě pro danou velikost obrazovky. Zobrazování různých verzí probíhá na základě porovnání šířky okna prohlížeče s šířkovými deskriptory, což jsou zvolená čísla následovaná znakem *w*, uvedeny za URL obrázku. Z praktického hlediska by hodnota deskriptoru měla prezentovat a být rovna skutečné šířce rastru. Do vlastnosti *srcset* je zpravidla uváděno více URL, které se oddělují čárkou. Pokud omezení pro rozměr okna splňuje více variant rastru, je vždy dle podmínek tohoto atributu stažen a zobrazen soubor v nejvyšší přípustné kvalitě. To přináší pozitivita v datové úspoře, jelikož je pokaždé stažen pouze potřebný rastr [30].

```

```

V ukázce zdrojového kódu (nad odstavcem) je uveden příklad zápisu značky *img* s atributem *srcset*. První obrázek s názvem *vychozi.png* je stažen a zobrazen pouze v případě, pokud webový prohlížeč nepodporuje zmiňovaný atribut *srcset* (například Internet Explorer 11 a nižší verze), jelikož základní vlastnost *src* podporuje každý běžně používaný prohlížeč. Obrázek s názvem *maly.png* je použit pro šířku okna prohlížeče pod 600 pixelů, *stredni.png* od 600 do 900 pixelů a *velky.png* nad 900 pixelů [31].

Mimo jednotky deskriptoru *w* je používána jednotka *x*. Ta je aplikována pro určení počtu HW pixelů na virtuální pixel (CSS pixel, referenční pixel apod.), což je CSS vlastnost *device-pixel-ratio* [30] a ta je důležitá pro mobilní telefony s Retina displejem od společnosti Apple a jim příbuzným AMOLED u telefonu s operačním systémem Android, jelikož mají několikanásobný počet HW pixelů oproti referenčním pixelům, a je proto nutné používat rastry ve vyšší kvalitě z důvodu jemnější mřížky displejů s vysokým rozlišením [31].

```

```

V ukázce výše je uveden zdrojový kód značky *img* s atributem *srcset* a deskriptorem používajícím jednotku *x*. Obrázek *img.png* je použit na klasických displejích a *img2x.png* je zobrazen na obrazovkách s poměrem počtu HW pixelů : CSS pixelům = 2 : 1, například Retina displeje. Ovšem tento deskriptor slouží pouze pro bližší specifikaci a individualizaci, jelikož prohlížeč zmiňovaný poměr počítá automaticky při detekci zařízení i v případě použití deskriptoru s jednotkou *w*. Tudíž v kódu uvedeném níže

prohlížeč použije obrázek *velky.png* i v případě malých zařízení s Retina displejem s CSS rozlišením například 540 pixelů, jelikož přes poměr 2 : 1 je vypočítáno HW rozlišení 1080 pixelů, což převyšuje omezení 900 pixelů pro použití rastru *velky.png* [32].

```

```

Atribut sizes

Tento atribut vyvolá určitou událost na webových stránkách pouze v případě, pokud je použit společně s atributem *srcset* (výchozí vlastnost *src* nestačí). Proto je často považován pouze na jeho rozšíření [31].

Atribut *sizes* má dvě základní funkce:

- může nastavit šířku rastru v závislosti na šířce okna prohlížeče a
- definovat deskriptoru *w*, že má jeho hodnotu porovnávat s hodnotou zadanou v *sizes* a nikoli se šířkou prohlížeče [30].

Praktické využití je prezentováno v ukázce zdrojového kódu pod odstavcem. Ten příkazuje pro zařízení s rozlišením nad 768 pixelů šířku rastru 300 pixelů. Na zařízeních s menším rozlišením než 768 pixelů má obrázek velikost 100 procent šířky viewportu a je zobrazen *maly.png* nebo *stredni.png*, což závisí na porovnání aktuálně používané velikosti obrazovky s deskriptorem.

```

```

3.1.6.3 Značka *picture*

Tento HTML element je podporován pouze nejnovějšími prohlížeči a stejně jako tag *img* s pokročilými atributy *src* a *sizes* slouží především k adaptivitě obrázku. Základní princip je téměř totožný, ale v detailech se tyto dvě technologie liší [33].

HTML prvek *picture* je kontejner, který neobsahuje žádné informace o obrázku, ale pouze zapouzdřuje elementy *img* a *source*, v jejichž attributech se nacházejí různé soubory. V prvně jmenovaném je uvedena výchozí varianta obrázku a ve druhém alternativa [34].

V jednom kontejneru *picture* musí být právě jedna značka *img*, jelikož je právě jeden výchozí rastr, ale elementů *source* může být několik v závislosti na tom, kolik je třeba alternativ. Prohlížeč pak následně zvolí nejvhodnější zdroj, stáhne obrázek a poté ho zobrazí [35]. Příklad zápisu zmiňovaného kontejneru je uveden pod odstavcem. V něm je jako výchozí soubor uveden *maly.png* v *img* a jeho alternativy jsou *stredni.png* a *velky.png* v *source*. Prvně jmenovaná možnost je zobrazena na displejích od šířky displeje 600 pixelů a druhá od 900 pixelů.

```
<picture>
  <source media="(min-width: 900px)" srcset="velky.png">
  <source media="(min-width: 600px)" srcset="stredni.png">
  
</picture>
```

Jak je patrné na kódu uvedeném výše, pro určení varianty je nutné použít dotazy na médium, a proto se pro detekci šířky, resp. zařízení, používají CSS pixely a nikoli HW pixely, jak je tomu při použití elementu *img* s atributem *srcset*, což je první odlišnost těchto dvou technologií. Druhá diference je, že HTML prvek *source* obsahuje i vlastnost *type* a v důsledku toho je možné používat nejen kontejner *picture*, ale i různé alternativy obrázků v odlišných formátech. Například při podpoře prohlížečem může být zobrazen rastr v novém formátu WebP, který je úspornější než klasické JPG. Pokud tomu tak není, je stáhnut a zobrazena výchozí varianta souboru [35]. Tento příklad je zobrazen kódem pod odstavcem.

```
<picture>
  <source type="image/webp" srcset="img.webp">
  
</picture>
```

3.1.7 CSS preprocesory

CSS preprocesor je programovací jazyk, pomocí něhož lze rozšířit, resp. vytvořit novou syntaxi pro zápis klasického jazyka CSS sloužícího k designování webu. Po napsání kódu novou syntaxí je CSS kód vygenerován [3].

Tuto technologii lze použít na základě dvou principů. První možností je využití technologií na straně klienta a druhou na straně serveru. Ovšem také je možné obě možnosti navzájem kombinovat [36].

Při použití technologie na straně klienta je nutné vlastní syntaxi interpretovat až na samotném webovém prohlížeči, což může značně snížit rychlost načítání webových stránek, jelikož se jedná o velice náročnou operaci na výkon. Proto existují tzv. hlídači, kteří mají za úkol kontrolovat, jestli se daný soubor změnil, a pouze v případě změny umožní znovu jeho překlad [3].

Při používání technologií na straně serveru je nutné na server instalovat knihovnu s kompatibilním kompilátorem. Při změně souboru dochází k automatickému překladu a vygenerování klasického kódu kaskádových stylů. Tento způsob tolik nezatěžuje webový prohlížeč, a proto neomezuje rychlost načítání, ale nevýhodou může být instalace kompilátoru [3].

3.1.7.1 Typy CSS preprocesorů

CSS preprocesorů je ve světě webového designu mnoho. Ve většině případů se liší pouze ve své syntaxi. Nejčastěji se ovšem v praxi i v odborné literatuře vyskytují především pouze 2 preprocesory – LESS a SASS [37].

LESS

Jak je již psáno výše, CSS preprocesor LESS je jeden z nejpoužívanějších. Umožňuje překlad do klasického kódu kaskádových stylů pomocí technologií na straně serveru i klienta [38].

LESS používá deklarativní zápis kódu, což znamená, že je podobný deklarativní povaze zápisu klasického jazyka CSS. Je tedy poměrně jednoduchý i pro začátečníky, a není proto složité přejít z kaskádových stylů na preprocesor LESS [38].

SASS

SASS se od LESS liší především v imperativním zápisu [36]. Z tohoto důvodu se více podobá spíše jazykům Ruby nebo PHP a nikoli běžnému CSS. K používání tohoto preprocesoru je nutné instalovat skriptovací jazyk Ruby [39].

SASS je možné zapisovat pomocí dvou syntaxí – tou první je původní SASS a druhou je SCSS (Sassy CSS), která přišla později. Ta je přehlednější, a proto i vhodnější pro začátečníky, kteří s tímto preprocesorem začínají [39].

Ostatní CSS preprocesory

Ačkoli jsou nejčastěji používanými CSS preprocesory SASS a LESS, jsou používány i jiné. Alespoň několik z nich je uvedeno v následujícím výčtu: Stylus, Myth, Clay, DtCSS, CSS-Crush nebo Rework [37].

3.1.7.2 Použití CSS preprocesorů

Tento nástroj je používán především k ulehčení zápisu složitých a rozsáhlých kaskádových stylů, které se stávají nepřehlednými, což umožňují vnořené definice. Mimo to také umožňují použití proměnných matematických výrazů, mixinů, nebo pracovat s barvami [40].

Vnořené definice

Vnořené definice slouží především k ulehčení zápisu složitých a rozsáhlých kaskádových stylů [3]. V ukázce pod odstavce je uveden rozdíl mezi klasickým CSS (vlevo) a LESS preprocesorem s použitím vnořených definic.

```
.box {
  width: 50%;
}

.box .text {
  font-size: 20px;
}

  .box .text p {
    padding: 5px;
  }
}
```

```
.box {
  width: 50%;

  .text {
    font-size: 20px;

    p {
      padding: 5px;
    }
  }
}
```

Proměnné

Možnost deklarace proměnných je obrovským přínosem preprocesorů oproti klasickým kaskádovým stylům. Proměnná je deklarována na začátku dokumentu (v případě LESS pomocí znaku @) a případná editace je pak provedena na všech místech, kde je použita.

Výše zmíněné lze tedy například použít pro hromadnou změnu velikosti písma nebo barvy [36], jak je uvedené v příkladu pod odstavcem (vlevo CSS bez použití proměnné a vpravo LESS). Není tedy nutné každý výskyt editovat postupně manuálně.

```
p {
    color: #000000;
}

h1 {
    color: #000000;
}
```

```
@font-color: #000000

p {
    color: @font-color;
}

h1 {
    color: @font-color;
}
```

Matematické výrazy

Pomocí preprocesorů je možné provádět základní matematické operace, jako je sčítání, odčítání, násobení a dělení. Ty lze použít například pro výpočet pixelů nebo jiné jednotky, kupříkladu i kódu barvy [36]. Zápis matematických výrazů pomocí LESS je zobrazený v kódu pod odstavcem.

```
@base: 30px;
@width-cell: @base * 2;
@height-cell: @base + @width-cell;
color: #222222 + #111111;
color: @base-color * 4;
```

Mixin

Mixin umožňuje ve své podstatě předávání vlastnosti jedné třídy jiné. Třída tedy může být automaticky rozšířena o další vlastnosti. Celý princip je založen na tom, že se založí třída s určitými vlastnosti a ta je poté celá vložena do nové třídy a tím jsou předány parametry [38]. Princip je zobrazen na příkladu kódu pod odstavcem, kde třídy *block1* a *block2* přijímají vlastnosti od třídy *border*. Zápis LESS je uveden vlevo a CSS výstup vpravo.

```
.border {
    border-top: 1px solid;
    border-bottom: 5px solid;
}

.block1 {
```

```
.block1 {
    Background-color: #fff;
    border-top: 1px solid;
    border-bottom: 5px solid;
}
```

```

    background-color: #fff;
    .border;
}

.block2 {
    .border;
}

```

```

block2 {
    border-top: 1px solid;
    border-bottom: 5px solid;
}

```

Práce s barvami

CSS preprocesor umožňuje provádět s barvami různé operace – zesvětlení, ztmavení, zvětšení/zmenšení průhlednosti, změna odstínu atd. Je možné pracovat se samotnou barvou nebo proměnnou a v ní uloženou barvou [36]. Příklad, který určitou barvu zesvětlí o 10% u nadpisů první úrovně, je zobrazen pod odstavcem.

```

@color: #000000;

h1 {
    color: lighten(@color, 10%);
}

```

3.1.7.3 Porovnání základní syntaxe CSS preprocesorů

Pro komparaci jsou vybrány preprocesory LESS a SASS, jelikož se jedná o nejčastěji používané nástroje a mají odlišnou syntaxi, deklarativní a imperativní. V porovnání je pro SASS použita novější syntaxe SCSS. Ve většině případů jsou zápisy velice podobné a liší se pouze v zástupném znaku.

Proměnné

Pomocí LESS je proměnná deklarována znakem @, za kterým je napsán název proměnné. U SASS, resp. SCSS, je jediný rozdíl – namísto @ je použit znak \$. Zbytek se nikterak od sebe neliší [37]. Deklarace proměnné *font-color* a její použití je zobrazeno v kódu pod odstavcem (vlevo LESS, vpravo SCSS).

```

@font-color: #000000;

p {
    color: @font-color;
}

```

```

$font-color: #000000;

p {
    color: @font-color;
}

```

Mixin

V případě SCSS není možné definovat třídu a tu následně zařadit do nové třídy, jako je tomu u LESS. Zde je zápis obsáhlejší. Při definování mixinu je nutné použít `@mixin` a jeho vložení provést příkazem `@include` [37]. Příklad je uveden níže (vlevo LESS, vpravo SCSS).

```
.border {
  border-top: 1px solid;
  border-bottom: 5px solid;
}

.block1 {
  background-color: #fff;
  .border;
}

.block2 {
  .border;
}
```

```
@mixin border {
  border-top: 1px solid;
  border-bottom: 5px solid;
}

.block1 {
  background-color: #fff;
  @include border;
}

.block2 {
  @include border;
}
```

3.1.8 CSS reset a CSS normalizace

Soubory CSS reset a CSS normalizace jsou soubory napsané v jazyce CSS. Jejich hlavní úkol je úprava defaultních stylů některých elementů v HTML dokumentu, které jsou nastavené ve webových prohlížečích. Tím je zabezpečeno, že při designování webu se nevyskytují výchozí (někdy nežádoucí) styly. V případě CSS resetu jsou všechny hodnoty stylů nastaveny na 0 a u normalizace jsou některé výchozí užitečné vlastnosti zachovány, případně upraveny [41].

Mezi nejpoužívanější CSS resety a normalizace lze zařadit: Yahoo! (YUI 3) Reset CSS, Eric Meyer's „Reset CSS“ 2.0, Universal Selector '*' Reset, HTML5 Doctor CSS Reset nebo Normalize.css 1.0 [12].

3.1.9 Framework

Obecně platí, že framework je struktura určená k tomu, aby sloužila jako podpora nebo průvodce pro budování něčeho, co samotný framework rozšiřuje na něco užitečného. V počítačových systémech je framework ve většině případů vrstvenou strukturou, na

jejímž základu může být vytvořen software. Některé frameworky počítačových systémů také zahrnují skutečné programy, specifikují programovací rozhraní nebo nabízejí programovací nástroje pro jejich samotné použití [42].

3.1.9.1 Typy frameworků

V oboru tvorby webových stránek a obecně v informačních systémech existují dva typy, které se používají: backend a frontend. Toto rozdělení je závislé na tom, kde má být framework využíván pro prezentační vrstvu nebo aplikační/logickou vrstvu [42].

Backend frameworky slouží především pro programátory pracující s jazykem PHP a databázemi. Naproti tomu frontend frameworky jsou určeny pro designéry, kteří pracují především s jazyky HTML, CSS a JavaScriptu. Tyto nástroje pro usnadnění práce se nazývají CSS frameworky [43].

3.1.9.2 CSS frameworky

Jedná se o sadu nástrojů vytvořených ze standardizovaných kódů HTML5, CSS3 a JavaScriptu. Jejich hlavním cílem je pomáhat designerům při kódování webových stránek tím, že není nutné webové stránky vytvářet úplně od začátku, jelikož jsou v něm obsaženy již základní prvky pro snadné rozvržení webových stránek i s odlišným vzhledem pro mobilní telefony a tablety. Tím je zrychlen celý proces prototypování a vývoje webů [43].

Obvyklé komponenty CSS frameworků jsou:

- CSS zdrojový kód pro vytvoření mřížky – umožňuje vývojáři umístit různé prvky, které vytvářejí design stránek jednoduchým a všestranným způsobem,
- definice typografického stylu pro prvky HTML,
- řešení případů nekompatibility prohlížeče, aby se stránky zobrazovaly správně ve všech prohlížečích a
- vytvoření standardních tříd CSS, které mohou být použity ke stylu pokročilých komponent uživatelského rozhraní [43].

3.2 Analýza CSS frameworků

Analýza a výběr CSS frameworku je prováděna metodami vícekriteriální analýzy variant. Pomocí ní lze určit nástroj, který je klasifikován jako nejlépe hodnocená kompromisní varianta.

3.2.1 Vícekriteriální analýza variant (VAV)

Vícekriteriální analýza variant řeší problematiku výběru jedné nebo více variant z množiny přípustných alternativ klasifikovaných podle kritérií. Řešení je nejlépe hodnoceným kompromisem na základě všech kritérií. Na základě variant a kritérií lze sestavit kritériální matici $Y=(y_{ij})$, jejíž prvky jsou tvořeny hodnocením i -té varianty podle j -tého kritéria [44].

Obrázek 14: Kritériální matice

$$\begin{array}{c} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_p \end{array} \begin{pmatrix} \mathbf{f}_1 & \mathbf{f}_2 & \dots & \mathbf{f}_k \\ \mathbf{y}_{11} & \mathbf{y}_{12} & \dots & \mathbf{y}_{1k} \\ \mathbf{y}_{21} & \mathbf{y}_{22} & \dots & \mathbf{y}_{2k} \\ \dots & \dots & \dots & \dots \\ \mathbf{y}_{p1} & \mathbf{y}_{p2} & \dots & \mathbf{y}_{pk} \end{pmatrix}$$

Zdroj: [46]

Kritéria mohou mít minimalizační nebo maximalizační charakter, což záleží na skutečnosti, zda usilujeme o vysoké nebo nízké hodnoty. Každé kritérium může mít jinou relativní důležitost (preferenci) vůči ostatním. Tuto podstatu zajišťuje váha kritérií, kterou lze vyjádřit číslem z intervalu 0 až 1. Ke stanovení vah i samotných kritérií je potřeba přistupovat co nejobektivněji. K určení vah je možné použít Saatyho metodu. Tato metoda je přesnější než ostatní, jelikož na základě kardinální informace zohledňuje i rozestupy mezi jednotlivými váhami, na což jiné metody vycházející ze žádné, nominální nebo ordinální informace neberou zřetel [45]. K samotné analýze jsou následně použity metody AHP a váženého součtu.

3.2.1.1 Saatyho metoda

Saatyho metoda je založena na párovém porovnání důležitosti objektů. Provádí se ve dvojrozměrné tabulce tzv. Saatyho matici, kde jsou v řádcích a sloupcích zapsané

hodnocené prvky, které jsou následně hodnoceny porovnáním každý s každým, k čemuž slouží stupnice celých čísel nabývajících hodnot 1 (rovnocennost) až 9 (absolutní preference). Standardně jsou používány pouze čísla 1, 3, 6, 7, 9 – jejich definování je následné:

- 1 - i a j jsou rovnocenná,
- 3 - i je slabě preferováno před j ,
- 5 - i je silně preferováno před j ,
- 7 - i je velmi silně preferováno před j a
- 9 - i je absolutně preferováno před j .

Ovšem je možné využít i mezistupně 2, 4, 6 a 8 pro jemnější rozlišení preference a dispreference. Při postupném porovnávání jsou objektům přiřazovány hodnoty dle vnímané důležitosti [45].

Saatyho matice je čtvercová a reciproční, což znamená, že při porovnání prvků je do pole na řádku preferovaného zapisováno celé číslo 1 až 9 a naopak do pole na dispreferovaném převrácená hodnota (například 1/9). Jelikož na diagonále jsou srovnávány dva stejné objekty, je hodnota daného pole rovna 1 (viz. obrázek č. 15)

Obrázek 15: Saatyho matice

$$\begin{matrix}
 & f_1 & f_2 & \dots & f_k \\
 f_1 & \left[\begin{array}{cccc}
 1 & s_{12} & \dots & s_{1k} \\
 1/s_{12} & 1 & \dots & s_{2k} \\
 \vdots & \vdots & & \\
 1/s_{1k} & 1/s_{2k} & \dots & 1
 \end{array} \right.
 \end{matrix}$$

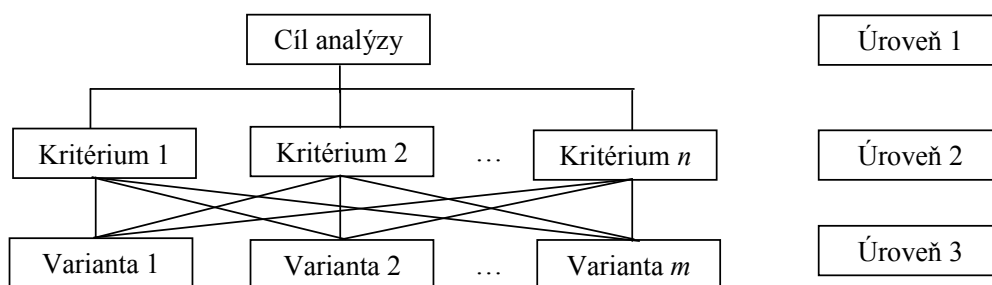
Zdroj: [46]

Po vyplnění celé matice je možno odvodit váhy jako normalizovaný geometrický průměr řádků. Geometrický průměr lze spočítat jako x -tou odmocninou součinu všech prvků na daném řádku, kde x je počet prvků na řádku. Následná normalizace je prováděna vydělením jednotlivých průměrů jejich celkovým součtem [44].

3.2.1.2 Metoda AHP

Metoda AHP (Analytic Hierarchy Process) byla navržena profesorem Saatyem v roce 1980. Tato metoda se snaží o zjednodušení složitých rozhodovacích problémů rozkladem složitých nestrukturovaných situací na jednodušší komponenty do několika úrovní, čímž vytváří hierarchický systém [46]. Na úrovni 1 je vždy cíl analýzy. Na úrovni 2 jsou kritéria, která hodnotíme podle stanoveného cíle. Tedy jakou měrou přispějí k naplnění cíle. Tato míra je určena váhami kritérií. Na 3. úrovni jsou varianty, jež jsou kvalifikovány na základě všech kritérií, také pomocí vah. Lze přidávat i další úrovně [44].

Obrázek 16: Schéma metody AHP



Zdroj: [46]

Metoda AHP je založená na postupném přerozdělování preferencí (vah) napříč celou hierarchickou strukturou. Úroveň 1 má vždy váhu 1. Tato preference je rozdělena mezi kritéria na 2. úrovni a následně se váha jednotlivých kritérií rozděluje mezi varianty na 3. úrovni [44]. Tímto způsobem metoda AHP postupně přiřazuje jednotlivým komponentám jejich důležitost. Na každé úrovni je vhodné pro větší přesnost použít dělení preferencí Saatyho metodu párového porovnání. Pro rozdělení vah z 2. na 3. úroveň se porovnávají hodnoty variant v rámci daného kritéria a následně se vypočtená normalizovaná váha varianty vynásobí normalizovanou vahou kritéria, podle kterého je právě prováděna komparace [45]. Po rozdělení všech preferencí se sečtou váhy kritérií pro každou variantu. Poté je vybrána varianta s maximální preferencí, tedy nejvyššího dosaženého součtu.

3.2.1.3 Metoda váženého součtu

U každé varianty lze také ohodnotit, jaký přináší dílčí užitek na základě daného kritéria. Tyto dílčí užítky je možno sloučit do celkového užitku varianty a podle toho jednotlivé alternativy porovnat. Pro metody pracující na tomto principu ovšem platí podmínka, že

data musí v kritériální matici nabývat pouze číselných hodnot. Případné slovní hodnocení kritérií je tedy třeba převést na odpovídající hodnocení číselné. Na tomto principu pracuje mimo jiné i metoda váženého součtu.

Základem pro tuto metodu je určení ideální varianty H s ohodnocením (h_1, \dots, h_n) složenou z nejlepších hodnot kritérií a bazální varianty D s ohodnocením (d_1, \dots, d_n) složenou z nejhorších hodnot kritérií. Následně se vytvoří standardizovaná kritériální matice dílčích užitek R . Hodnota dílčího užtku r_{ij} , tedy i -té varianty podle j -tého kritéria lze vypočítat podle vzorce:

$$r_{ij} = \frac{y_{ij} - d_j}{h_j - d_j}.$$

Celkový užitek i -té varianty je vypočten jako součet dílčích užitek pomocí skalárního součinu standardizované matice a vektoru vah, tedy váhy všech kritérií. Za nejlepší variantu je zvolena ta, jejíž celkový užitek je nejvyšší [44].

4 Vlastní práce

V této části diplomové práce je nejprve proveden výběr CSS frameworků a posléze jsou tyto nástroje analyzovány pomocí vícekriteriální analýzy variant. Na závěr této části je aplikován vybraný nástroj, který je v analýze kvantifikován jako nejlépe hodnocená kompromisní varianta.

4.1 Analýza CSS frameworků

Analýza je zahájena stanovením kritérií a následně jim jsou přiděleny váhy Saatyho metodou. Poté je proveden výběr a charakterizování CSS frameworků. Posléze jsou zvolené softwary analyzovány pomocí vícekriteriální analýzy variant, konkrétně metodami AHP a váženého součtu.

4.1.1 Stanovení kritérií

V rozhodovacím procesu hraje klíčovou roli volba kritérií. Na jejich základě jsou vyhodnoceny varianty. Samozřejmě je možné vybrat mnoho různých kritérií pro hodnocení CSS frameworků, ale z důvodu orientace této práce jsou zvolena především kritéria se zaměřením na responzivní design a layout webových stránek podle dle subjektivního postoje autora práce vycházejícího zejména z vlastních pracovních zkušeností a konzultací s dvěma odborníky z praxe, kteří se zabývají designem webových stránek.

Pokud jde o vývojáře, jedná se především o jednoduchou implementaci CSS frameworků. Dále je nezanedbatelným prvkem dokumentace, která usnadňuje vývojářům vytvořit web. Také je velice zajímavá podpora CSS preprocesorů. Pro vývojáře je značnou výhodou, pokud je možné stáhnout zdrojový kód stylů frameworku v preprocesorech a s tím spojené přizpůsobení neboli customizace, zejména u rozsáhlých frameworků jako je Foundation nebo Bootstrap a další. U nich jsou jednotlivé úpravy defaultního nastavení pomocí klasických CSS stylů poměrně zdlouhavý proces. Dále je velmi důležitá podpora prohlížečů, zejména pro uživatele, kteří používají starší verze některých prohlížečů. Mimo uvedené problematiky je podstatná i podpora různých funkcí, například responzivní navigace nebo slideru. Konkrétní kritéria a jejich popis jsou uvedeny níže.

Velikost frameworku

Velikost frameworku je důležitá zejména kvůli mobilnímu připojení, kdy kvůli omezení FUP je podstatný téměř každý objem dat. Mimo to taky ovlivňuje rychlost načítání webových stránek. Proto je důležité, aby byl objem dat celého CSS frameworku co nejmenší. Do velikosti je zahrnut kompletní objem dat obsahující výchozí nastavení frameworku. Jedná se tedy především pouze o CSS soubor s kaskádovými styly a někdy také JS s JavaScriptem.

Implementace

Toto kritérium je zaměřeno především na jednoduchost implementace, podporu balíčkových systémů a počet způsobů, jakými lze framework implantovat dle oficiálních webových stránek jednotlivých nástrojů. Konkrétně jsou klasifikovány tři možnosti.

Kromě výchozího způsobu, který se ve většině případů snaží o co největší datovou úsporu a obsahuje pouze tedy CSS soubor s kaskádovými styly a někdy JS s JavaScriptem, je v analýze zohledňováno, zda je také možnost implantace pomocí CSS preprocesoru. Poslední možnost pro použití frameworku, která je v rámci tohoto kritéria klasifikována, je přes CDN server. Pomocí něho se zvyšuje rychlost načítání stránek, jelikož se při první návštěvě určitých webových stránek framework stáhne a uloží do cache webového prohlížeče. Při dalších návštěvách se tedy nemusí zdrojové soubory nástroje stahovat, a proto je zrychleno načítání webových stránek

Dle tohoto kritéria je dále velmi podstatné zohlednit i dokumentaci. Ta může velmi zjednodušit samotnou implementaci, pokud je kvalitní, rozsáhlá, popisující komponenty, obsahuje doporučení, tutoriály nebo instruktážní videa.

Customizace

Dle tohoto faktoru je hodnoceno především z hlediska možnosti stažení CSS frameworku i ve verzi podporující minimálně jeden CSS preprocesor. Pomocí nich je možné provádět hromadné úpravy jednotlivých stylů na celém webu zároveň. Dále je také zohledněna podpora samotné customizace před stažením frameworku. Tedy zda je možné vybrat jednotlivé komponenty nebo nastavit jejich vlastnosti na domovské stránce frameworku a až poté stáhnout upravenou variantu, nebo zda je možné stáhnout pouze jednu verzi.

Mřížka

Zde záleží především na tom, kolika sloupcová je výchozí mřížka. Velice dobře jsou hodnoceny varianty, které mají dvanáctisloupcovou mřížku (viz. literární rešerše, podkapitola Fluidní mřížka).

Zlomové body

Vhodně vytvořené breakpointy již ve výchozím nastavení frameworku mohou zjednodušit práci při tvorbě responzivního designu. Pokud je dostupný framework i jako varianta s CSS preprocesorem, je většinou možné zlomové body vytvořit vlastní či defaultní editovat. V několika málo případech je možné tyto operace provádět u systému i bez preprocesoru. Proto je hodnocen především počet breakpointů a hodnota (tedy umístění), jak moc se tyto veličiny přibližují ideálnímu rozvržení (viz. obrázek č. 12), popřípadě možnost jejich vytváření či editace především u nástrojů, které nemají ideální rozvržení. Tudíž zda je možné provádět avizované operace pouze s preprocesorem, bez něj nebo vůbec

Navigace

Obvyklý problém, který vývojáři potřebují vyřešit při vytváření webových prezentací, je vývoj responzivní navigace. Ta je podstatná pro prohlížení obsahu na webových stránkách. Standard je, že tyto navigační panely se u zařízení s malým rozlišením obrazovky změní na jedno tlačítko, které po aktivaci kliknutí zobrazí jednotlivé položky menu, a tak nezabírá na webu příliš místa

Proto je dle tohoto kritéria hodnoceno především to, zda se tento typ navigačního panelu u jednotlivých CSS frameworků vyskytuje, a pokud ano, tak i rozsah možností jeho nastavení a kvalita zobrazení.

Responzivita obrázků

Vysoce diskutované téma v odvětví responzivního designu webových stránek je používání obrázků. Vývojáři se při práci setkávají s několika zásadními problémy. Patří mezi ně například zvolení vhodných rozměrů používaných rastrů, jejich objemová náročnost nebo jejich rozlišení.

Vzhledem k výše uvedeným problémům je hodnoceno, zda CSS framework obsahuje řešení pro responzivitu médií a na jaké úrovni. Například zda dochází ke stažení pouze potřebného rastru, nebo jsou staženy všechny verze, což je nevhodný způsob, jelikož nepřináší žádnou datovou úsporu.

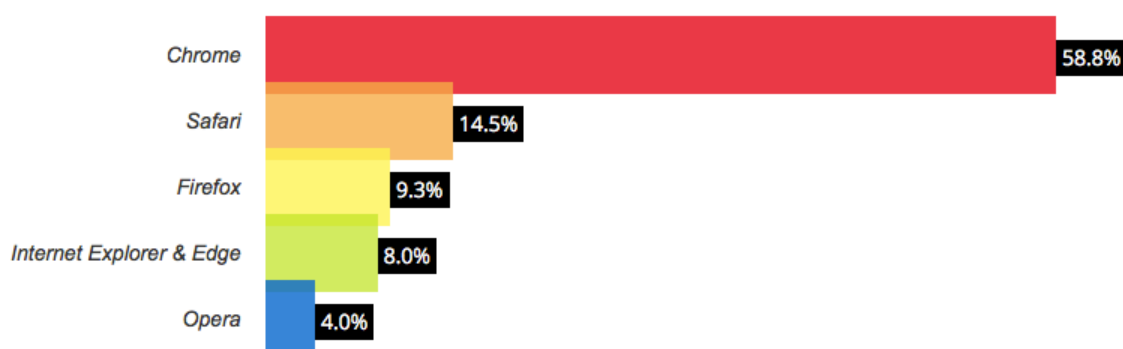
Carousel

Carousel neboli slider je prolínání jednotlivých obrazů pomocí různých efektů. Používá se především v hlavičce webových prezentací. Ačkoli slider přestává být pomalu trendem na poli responzivního designu, je stále ze strany klientů poměrně často vyžadován, jelikož je zde možné do záhlaví vměstnat více hlavních informací. Funkcionalita carouselu je většinou zabezpečena pomocí skriptovacího jazyka JavaScript. Ten ale nemá podstatný vliv na jeho zobrazení. Proto se samotná funkcionalita nebere v potaz, ale hodnoceno je pouze přítomnost či absence carouselu a kvalita jeho zobrazení, zejména na různých mobilních zařízeních.

Podpora prohlížečů

Pro vývojáře je podpora webových prohlížečů důležitá. Proto je podstatné toto kritérium neopomenout při analýze. V jeho rámci je zohledňována podpora pěti aktuálních i starších verzí nejvíce rozšířených webových prohlížečů (Google Chrome, Safari, Mozilla Firefox, Internet Explorer/Microsoft Edge a Opera), která je uvedena v grafu č. 2 pod odstavcem. Podpora starších verzí se týká zejména prohlížeče Internet Explorer, který je pro vývoj webových prezentací velmi specifický a je stále poměrně často používán, zejména mezi starší populací.

Graf 2: Pět nejvíce využívaných webových prohlížečů v procentech



Zdroj: [46]

Glyfy

Jelikož uživatelé začínají požadovat přehledné webové stránky, kde není příliš textu, začínají se některé textové prvky nahrazovat nebo doplňovat ikonami charakterizující daný textový element. Přítomnost takových ikon tedy může designérovi značně ulehčit práci na webových stránkách.

V rámci tohoto kritéria je zejména klasifikováno, zda CSS framework ve výchozí verzi obsahuje sadu takovýchto ikon (piktogramů) známých jako glyfy, tedy jejich počet. Jedná se tedy o kritérium kvantitativního charakteru.

4.1.2 Charakteristika variant dle kritérií

Pro analýzu jsou vybrány CSS frameworky Bootstrap, Foundation, Skeleton, Kube a Materialize. Hodnoty kritérií kvantitativní povahy, tedy *velikost frameworku* a *glyfy*, jsou získány z volně dostupných zdrojů. Kvalitativního jsou klasifikována na základě vlastních zkušeností řešitele s vybranými nástroji a důkladné studii dokumentací k daným systémům. V následujících podkapitolách jsou vybrané nástroje jednotlivě a postupně klasifikovány podle všech hledisek určených pro analýzu. Tabulka 4 uvádí všechna hlediska pro porovnávání včetně jejich jednotek. Míra bodového ohodnocení může nabývat hodnot od 1 do 10, kde 1 je nejhorší a 10 nejlepší.

Kromě kvantitativních kritérií *velikost frameworku* s maximalizačním charakterem a *ikony* s minimalizačním charakterem jsou všechna stanovená kritéria kvalitativní s maximalizačním charakterem.

Tabulka 4: Kritéria pro analýzu

Číslo	Kritérium	Jednotky
1	Velikost frameworku	kB
2	Implementace	bodové ohodnocení
3	Customizace	
4	Mřížka	
5	Zlomové body	
6	Navigace	
7	Responzivita obrázků	
8	Carousel	
9	Podpora prohlížečů	
10	Glyfy	

Zdroj: Vlastní zpracování

4.1.2.1 Bootstrap

Bootstrap je pokročilá frontendová struktura, která slouží nejen k vývoji responzivních webů. Tento framework byl původně vytvořen v roce 2010 Jacobem Thorntonem a Markem Ottou pro lepší implementaci a vývoj interních nástrojů na Twitteru. O rok později byl tento nástroj veřejně poskytnut všem vývojářům jako open source. Nejnovější verzi ke stažení je Bootstrap, verze 4.0, ale ta je v době psaní práce dostupná pouze ve formě beta. Proto je pro analýzu zvolena poslední dostupná klasická plná verze, což je varianta 3.3.7.

Velikost frameworku

Velikost: 177 kB

Implementace

Na oficiální webové stránce je Bootstrap k dispozici ve třech, respektive čtyřech, verzích, jelikož kromě výchozí verze a přes CDN server je možno nástroj získat se dvěma různými CSS preprocesory, konkrétně SASS a LESS. Bootstrap je podporován balíčkovými systémy Bower, npm a Composer.

Dokumentace obsahuje tutoriály s příklady, které slouží pro úvodní obeznámení s frameworkem Bootstrap. Na domovské stránce nástroje jsou také hotová řešení. Dále je zde uveden detailní přehled o systému mřížky, tabulkách, formulářích, glyfech, navigaci a dalších komponentách tohoto nástroje. Také jsou k dispozici příručky pro použití JavaScriptu a CSS preprocesorů LESS i SASS, ale schází instruktážní videa a možnost zkoušení kódu jednotlivých komponent online.

Hodnocení: 8 bodů

Customizace

Framework podporuje customizaci třemi způsoby kromě manuální editace v CSS souborech. První možnost je načtení konfiguračního souboru a poté automatické vygenerování kódu. Druhý způsob spočívá ve vyplnění formuláře se seznamem jednotlivých komponent na domovské stránce Bootstrapu, kde je možné nastavit barevné proměnné, typografii, barvu pro jednotlivé komponenty, tabulky, barvy tlačítek, formuláře, zlomové body pro dotazy na médium nebo jiné nastavení a poté nechat vygenerovat

editovanou verzi frameworku s vybranými komponentami a jejich editovanými atributy. Třetí možností je stažení nástroje s jedním ze dvou preprocesorů a poté provádět editaci.

Hodnocení: 10 bodů

Mřížka

Výchozí nastavení mřížky je nadefinováno na počet dvanácti sloupců, ale je možné nastavit jakýkoli počet sloupců v customizaci na domovské stránce a stažení editované verze nebo pomocí preprocesoru.

Hodnocení: 10 bodů

Zlomové body

Zlomové body jsou ve výchozím nastavení obsaženy tři, konkrétně 768px, 992px a 1200px, což umožňuje vytvářet odlišný vzhled pro čtyři různá zařízení (0-767px, 768-991px, 992-1199p, 1200px a více). Pro ideální rozmístění zlomových bodů tedy schází breakpoint na úrovni 600px (viz. obrázek č. 12).

Ve verzi frameworku s CSS preprocesorem je sice možné breakpoints vytvářet, ale jelikož s touto operací při vývoji nástroje pravděpodobně nebylo počítáno, není ve zdrojových souborech nic připravené, a je tedy poměrně komplikovaná.

Ovšem je možné editovat pozici breakpointů jednoduše nejen ve verzi s preprocesorem, ale i na domovské stránce frameworku a poté stáhnout upravenou verzi nástroje. Takto jednoduchou editaci nabízí jako jediný z vybraných systémů. Například je možné předefinovat breakpoints na hodnoty 600px, 900px a 1200px. Tyto hodnoty by měly být plně dostačující pro vývoj webových stránek. I přes neideální výchozí rozvržení breakpointů řešitel hodnotí Bootstrap vysokým hodnocením z důvodu jednoduché editace.

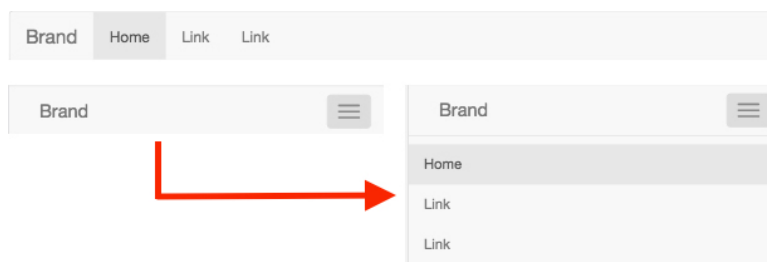
Hodnocení: 8 bodů

Navigace

Bootstrap obsahuje komponentu sloužící k vytvoření plně responzivní navigace. Má připravenou jasně definovanou strukturu, která je doplněna hotovými kaskádovými styly a skripty zajišťujícími funkčnost samotného panelu.

Responzivita menu je zde řešena tak, že se při určité šířce displeje všechny odkazy skryjí a navigační menu je následně zobrazeno vyjetím svrchu displeje po kliknutí na tlačítko (viz. obrázek č. 17).

Obrázek 17: Responzivní navigační panel frameworku Bootstrap (nahore verze pro desktopy, dole verze pro telefony)



Zdroj: Vlastní zpracování

Slabinami jsou absence možnosti editace výchozího řešení uspořádání rozložení položek v menu při prohlížení webu na zařízeních s menšími obrazovkami nebo zvolení zobrazení responzivního navigačního panelu formou vyjetí z boku obrazovky.

Hodnocení: 7 bodů

Responzivita obrázků

Responzivita obrázků je u Bootstrapu řešena dvěma způsoby. První možností je danému rastru přiřazení třídy *img-responsive*, která způsobí proměnlivost velikosti objektu v závislosti na rozměrech obalového elementu.

Druhou možností je vytvoření obrázku v několika variantách pro konkrétní zařízení a poté dané verze rastru skrýt třídou *hidden-X* nebo zviditelnit třídou *visible-X*, kde *X* prezentuje určité zařízení v závislosti na zlomových bodech.

Ať je zvolena první nebo druhá možnost, nedochází k žádné datové úspoře, jelikož je buď používán jeden objekt s variabilní velikostí pro všechna zařízení, nebo je používáno více verzí, ale jsou stahovány všechny formy rastru, které jsou na určitých zařízeních pouze skrývány. U obrázků s velkou datovou náročností je tedy vhodné použít spíše značku *picture* nebo nové atributy elementu *img* než defaultní možnosti CSS frameworku Bootstrap.

Hodnocení: 5 bodů

Carousel

V probíraném nástroji mají vývojáři připravený prvek nazývaný Carousel. Jeho součástí je i prostor pro responzivní titulek obrázku, ale v případě použití delšího textu je problematické zobrazení na menších mobilních zařízeních. Editace jeho velikosti není ošetřena ve výchozích zdrojových souborech, a je proto nutné tuto skutečnost opravit pomocí dotazů na médium.

Další slabinou komponenty Carousel je, že ve výchozím nastavení obraz vždy pokrývá celou šířku obalového elementu a jeho výška je dopočítávána podle poměru stran původního rastru. Nedochází tedy k přizpůsobení. Z toho vyplývá, že pokud se použijí k vytvoření slideru obrázky s různými rozměry, nebude jednotný vzhled jednotlivých položek carouselu. Ovšem tato skutečnost se vyskytuje prakticky téměř ve všech sliderech, které jsou dostupné. Tudíž ji nelze považovat za velký nedostatek. Navíc je možné této slabině předcházet tím, že rastry do carouselu uživatel připraví všechny ve stejných rozměrech.

Také je důležité zmínit, že Bootstrap ve výchozích zdrojových souborech má k dispozici pouze jeden efekt carouselu. Pro jiné animace je nutný pokročilý zásah do kaskádových stylů a v některých případech i JavaScriptu.

Ovšem dle názorů řešitele jsou zmíněné nedostatky drobnosti, které při použití obrázků pro slider ve stejných rozměrech nijak neovlivňují kvalitu responzivního designu webové stránky. Proto i přes tyto drobné nedostatky je nástroj Bootstrap klasifikován poměrně vysokým hodnocením.

Hodnocení: 8 bodů

Podpora prohlížečů

Podpora je zaručena pro všechny aktuální verze prohlížečů, dle kterých je kritérium hodnoceno, kromě Safari pro OS Windows. Internet Explorer je sice podporován už od verze 8, ale pro dotazy na médium je třeba importovat Respond.js nebo jinou knihovnu pro jejich funkčnost a není zaručena podpora většiny prvků CSS3 a HTML5.

Hodnocení: 9 bodů

Glyfy

Počet: cca 260 ks

4.1.2.2 Foundation

Foundation je CSS framework firmy ZURB, která se zabývá navrhováním, designem a vývojem webových stránek. Stejně jako jeho konkurent Bootstrap od společnosti Twitter, byl vyvinut jako interní nástroj, který podnik Zurb implementoval do všech svých projektů. Později, konkrétně v roce 2011, byl tento nástroj uvolněn a od té doby je pro veřejnost volně dostupný zcela zdarma. Je testována nejnovější verze 6.4.3.

Velikost frameworku

Velikost: 293 kB

Implementace

Tento framework je ve třech verzích – klasická s kaskádovými styly, přes CDN server nebo varianta s CSS preprocesorem SASS. Podpora je zaručena balíčkovými systémy Bower, npm, Composer.

Dokumentace je plně dostačující. Je v ní obsaženo vše potřebné pro implementaci i následující vývoj včetně informací o použitém JavaScriptu, CSS preprocesorech a kaskádových stylech týkajících se především komponent obsažených v nástroji Foundation. Také je dostupné téměř ke každému prvku instruktážní video sloužící jako tutoriál a dále je možné editování kódu dané komponenty online, což může vývojářům velice ulehčit a zrychlit tvorbu webových prezentací.

Hodnocení: 10 bodů

Customizace

Tento nástroj podporuje customizaci dvěma způsoby kromě manuální editace kaskádových stylů. Zaprvé je možné vyplnit formulář se seznamem jednotlivých komponent na domovské stránce Foundationu, kde lze vybrat jednotlivé komponenty nebo nastavit jejich vlastnosti a poté nechat vygenerovat editovanou verzi frameworku. Druhou variantou je stažení frameworku ve verzi s preprocesorem SASS a poté provádět editaci. Oproti některým konkurenčním je zde pouze možnost varianty s jedním preprocesorem a chybí

nahrání konfiguračního souboru s následným automatickým vygenerováním kódu a formulář pro editaci defaultní verze obsahuje poměrně málo komponent k přizpůsobení.

Hodnocení: 7 bodů

Mřížka

Výchozí nastavení mřížky je identické s frameworkem Bootstrap. Také v tomto případě je defaultní nastavení definováno na dvanáct sloupců s možností konfigurace gridu na libovolný počet sloupců pro různorodé projekty pomocí customizace u varianty s preprocesorem nebo na domovské stránce vyplněním položky ve formuláři a stažení editované verze.

Hodnocení: 10 bodů

Zlomové body

Zlomové body jsou ve výchozím nastavení definovány dva, konkrétně 640px a 1024px, což umožňuje responzivní design pro tři velikosti obrazovky. Podle názoru řešitele je sice umístění breakpointů poměrně blízko ideálním hodnotám, ale jejich počet, tedy dva, je příliš nízký, což může být značným nedostatkem, jelikož nelze vytvořit individuální design webových prezentací například při šířce obrazovky nad 1200px.

Ve verzi frameworku s CSS preprocesorem je sice možná velice jednoduchá editace, a dokonce elementární definování až dalších tří nových zlomových bodů, což zapříčiňuje možnost optimalizace designu až pro šest zařízení. Není ale možné tyto operace provádět v klasické verzi obsahující pouze soubory s kaskádovými styly a JavaScriptem.

Avizované skutečnosti mohou způsobit problémy zejména pro vývojáře, kteří neupřednostňují práci s variantou s preprocesorem, ale pouze klasickou defaultní formu CSS frameworku Foundation.

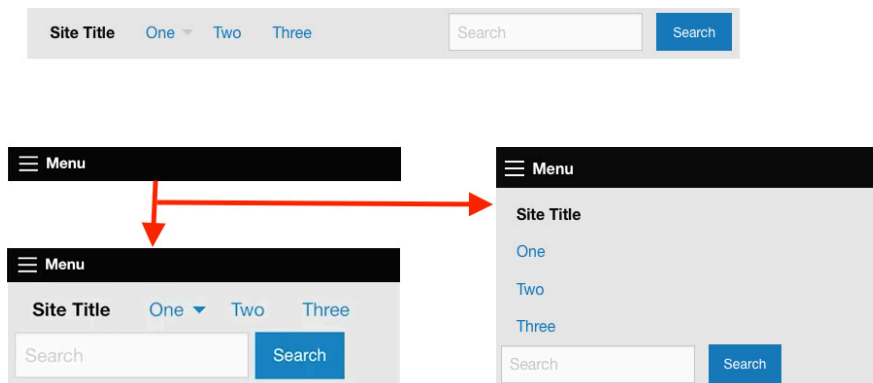
Hodnocení: 6 bodů

Navigace

Foundation řeší responzivní navigační panel standardním schováním položek menu a jejich následné zobrazení vyjetím svrchu displeje po kliknutí na tlačítko, ale například oproti

Bootstrap nabízí i možnost jiného rozložení položek (viz obrázek č. 18). Ovšem nenabízí možnost jednoduše zvolit responzivní navigační panel formou vyjetí z boku obrazovky.

Obrázek 18: Responzivní navigační panel frameworku Foundation (nahore verze pro desktopy, dole 2 verze pro telefony)



Zdroj: Vlastní zpracování

Hodnocení: 8 bodů

Responzivita obrázků

Responzivita nejen obrázků, ale i jiných médií je ve frameworku Foundation řešena samostatnou komponentou nazývanou Interchange. Pomocí ní je možné zobrazovat konkrétní obrázky, respektive jednotlivé verze obrázků, pro uživatele v závislosti na velikosti obrazovky.

Jsou zde používány dotazy na médium k identifikaci zařízení uživatele a objektu, který má být na daném zařízení zobrazen, a ten je používán. Tudíž je ze serveru stažena pouze potřebná varianta a dochází tedy k úspoře dat na mobilních zařízeních, kde jsou dostatečné rastry s menšími rozměry.

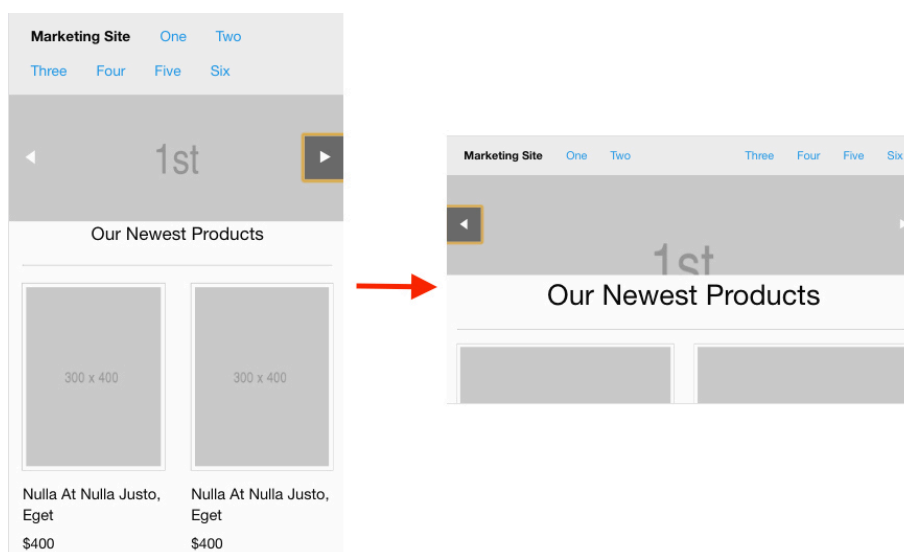
Hodnocení: 10 bodů

Carousel

Carousel je ve Foundationu řešen pomocí komponenty Orbit. Ta má oproti konkurenci výhodu, že obsahuje v základní verzi frameworku více nastavení efektu prolínání mezi jednotlivými slidy.

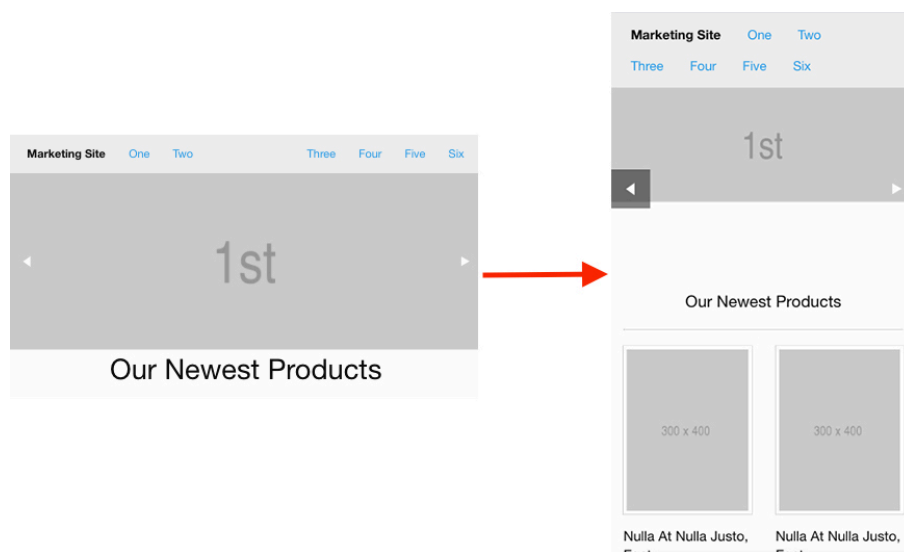
Orbit má obvyklé nedostatky sliderů identické například s komponentou Carousel frameworku Bootstrap. Ale mimo to má poměrně velkou slabinu. Orbit nedokáže po načtení webové stránky již přizpůsobovat výšku slideru. Tento jev může nastat kupříkladu při změně orientace obrazovky na telefonu nebo tabletu, tedy pokud dojde k přechodu z režimu landscape do portrait nebo naopak. V takovémto případě dojde k oříznutí rastru (viz. obrázek č. 19) nebo vzniku velkého odsazení (viz. obrázek č. 20). Pro správné zobrazení je nutné aktualizovat webovou prezentaci.

Obrázek 19: Špatné zobrazení slideru u komponenty Orbit při změně z režimu portrait do landscape



Zdroj: Vlastní zpracování

Obrázek 20: Špatné zobrazení slideru u komponenty Orbit při změně z režimu landscape do portrait



Zdroj: Vlastní zpracování

Zmiňovanou slabinu je nutné opravit editací kaskádových stylů a JavaScriptu nebo celou komponentu Orbit nahradit jiným sliderem se správným zobrazením, ale tím je potlačována samotná podstata použití frameworku jako komplexního řešení pro responzivní design webových stránek.

Hodnocení: 6 bodů

Podpora prohlížečů

Podpora je zaručena pro všechny aktuálně používané webové prohlížeče, podle kterých je toto kritérium hodnoceno. Internet Explorer od společnosti Microsoft je podporován už od verze 9.

Hodnocení: 9 bodů

Glyfy

Počet: cca 170 ks

4.1.2.3 Skeleton

Skeleton je dvanáctisloupcový framework obsahující základní typografii, minimum kaskádových stylů pro běžné HTML elementy, jako jsou tlačítka, odkazy apod. Jeho síla spočívá především v jednoduchosti, jelikož se jedná o nenáročný CSS framework, který je tvořen pouze 400 řádky kódu. Analyzována je nejnovější dostupná verze nástroje nazývaná Skeleton 2.0.4

Velikost frameworku

Velikost: 11 kB

Implementace

Skeleton je z oficiálních stránek možné získat ve dvou, respektive třech, formách. Defaultní varianta s kaskádovými styly nebo verze s CSS preprocesory SASS i LESS. Na oficiální webové stránce není dostupná možnost implementace přes CDN server. Skeleton je podporován balíčkovými systémy Bower, npm a Composer.

Samotný framework není příliš rozsáhlý ani dokumentace není objemná, ale obsahuje popis všech komponent, které nástroj nabízí. Tedy grid, typografii, tlačítka, formuláře,

dotazy na médium a zlomové body. Ke všemu jsou uvedeny příklady i se zdrojovými kódy, ale schází instruktážní videa a možnost zkoušení kódu jednotlivých komponent online.

Hodnocení: 7 bodů

Customizace

Kromě manuální editace kaskádových stylů je customizace možná pouze při použití varianty s CSS preprocesorem. Jiné možnosti nejsou dostupné, například ani editace nástroje na domovské webové stránce a stažení upravené verze.

Hodnocení: 4 body

Mřížka

Skeleton má ve výchozím nastavení dostupnou dvanáctisloupcovou mřížku. Tu ale nelze lehce editovat, jak je tomu u dvou předchozích nástrojů. Není tedy možné jednoduše změnit defaultní počet sloupců.

Hodnocení: 7 bodů

Zlomové body

Zlomových bodů je obsaženo 5 – konkrétně 400px, 550px, 750px, 1000px, 1200px. Ty jsou zvoleny ideálně. Tři zlomové body se pohybují okolo ideálních hodnot (viz. obrázek č. 12), konkrétně 550px, 1000px a 1200px. Breakpointy 400px a 750px jsou tedy ve své podstatě navíc, ale u některých projektů určitě využitelné. Případně je také možné provádět jejich úpravy ve variantě nástroje s CSS preprocesorem SASS. Jediný nedostatek tedy může být absence jednoduché editace breakpointů pro klasickou CSS verzi frameworků. Ovšem kvůli ideálnímu rozložení a velkému počtu ve výchozí variantě řešitel tuto skutečnost považuje za minimální slabinu.

Hodnocení: 9 bodů

Navigace

Skeleton neobsahuje ve výchozí verzi žádné řešení pro responzivní navigační panel.

Hodnocení: 1 bod

Responzivita obrázků

Responzivitu obrázku je možné řešit pouze přiřazením třídy *u-max-full-width*, která způsobí proměnlivost velikosti závislou na obalovém elementu. Nedochozí tedy k žádné datové úspoře, jelikož je používán jeden objekt s variabilní velikostí. U obrázků s velkou datovou náročností je proto vhodné používat značku *picture* nebo *img* novými atributy.

Hodnocení: 4 body

Carousel

Skeleton neobsahuje ve výchozí verzi žádné řešení pro Carousel.

Hodnocení: 1 bod

Podpora prohlížečů

Podpora je zaručena pro všechny aktuálně používané webové prohlížeče, podle kterých je toto kritérium hodnoceno. Internet Explorer od společnosti Microsoft je podporován už od verze 9.

Hodnocení: 9 bodů

Glyfy

Počet: 0 ks

4.1.2.4 Kube

Kube je kompletní CSS framework vytvořený firmou Imperavi. Tento nástroj je datově nenáročný se zaměřením na vyváženou typografii a obsahuje mnoho užitečných komponent. Analyzována je nejnovější dostupná verze Kube 6.5.2.

Velikost frameworku

Velikost: 87 kB

Implementace

Na domovské stránce jsou dostupné dvě verze frameworku Kube. První je klasická výchozí varianta s kaskádovými styly a JavaScriptem. Druhým způsobem je získání

nástroje s CSS preprocesorem LESS. Na oficiálním webu tedy není dostupná možnost implementace přes CDN server.

V dokumentaci je obsažen popis mřížky, tabulek, typografie a jiných používaných komponent. Jsou taky velice dobře zpracovány příklady použití jednotlivých součástí. Dokumentace také poskytuje vysvětlení použitých JavaScriptových funkcí, ale nejsou dostupné instruktážní videa a možnost zkoušení kódu jednotlivých komponent online.

Hodnocení: 7 bodů

Customizace

Kromě manuální editace kaskádových stylů je customizace možná pouze při použití varianty s CSS preprocesorem LESS. Jiné možnosti nejsou dostupné, například ani editace nástroje na domovské webové stránce a stažení upravené verze.

Hodnocení: 4 body

Mřížka

CSS framework Kube má ve výchozím nastavení dostupný pouze dvanáctisloupcový grid systém. Ten ale nelze lehce editovat. Není tedy možné jednoduše změnit defaultní počet sloupců ani pomocí CSS preprocesoru LESS, stejně jako je tomu u konkurenčního frameworku Skeleton.

Hodnocení: 7 bodů

Zlomové body

U frameworku Kube je situace podobná jako u nástroje Bootstrap. Také v tomto případě jsou breakpoint obsaženy s vhodným a podobným rozložením – konkrétně 768px, 1024px a 1200px. Schází tedy také ještě jeden definovaný breakpoint na úrovni přibližně 600px.

Tuto situaci lze tedy řešit stejně jako u systému Bootstrap, a to předefinováním defaultních zlomových bodů. Ovšem u Kube je jejich editace možná pouze u verze s preprocesorem, což je značný nedostatek, který působí problémy vývojářům upřednostňujícím klasickou variantu frameworku.

Hodnocení: 5 bodů

Navigace

Responzivní navigace je řešena pomocí schování položek pod panel, který se zobrazí po kliknutí na tlačítko. Kube jde opačnou cestou než jeho konkurenti Bootstrap nebo Foundation, jelikož navigační panel je zobrazován vyjetím z boku obrazovky zleva nebo zprava a nikoli svrchu. Druhou zmiňovanou alternativu ani nelze jednoduše definovat přiřazením třídy a také chybí možnost změny rozložení jednotlivých položek menu.

Hodnocení: 7 bodů

Responzivita obrázků

Framework Kube neobsahuje komponentu, která řeší responzivitu obrázků.

Hodnocení: 1 bodů

Carousel

Framework neobsahuje ve výchozí verzi žádné řešení pro Carousel.

Hodnocení: 1 bod

Podpora prohlížečů

Podpora je zaručena pro všechny aktuálně používané webové prohlížeče, podle kterých je toto kritérium hodnoceno. Internet Explorer od společnosti Microsoft je podporován až od verze 11.

Hodnocení: 7 bodů

Glyfy

Počet: 0 ks

4.1.2.5 Materialize

Jedná se o poměrně nový nástroj mezi front-end frameworky. Vyvíjen je společností Google. Obsahuje poměrně dost CSS komponent i JS pro různé animace. Analyzována je nejvyšší dostupná verze Materialize 1.0.

Velikost frameworku

Velikost: 326 kB

Implementace

Dostupnost frameworku Materialize je shodná s jeho konkurenčním nástrojem Foundation. Také je dostupný ve třech verzích – klasická s kaskádovými styly, implementace přes CDN server a varianta s CSS preprocesorem SASS. Podpora je zaručena balíčkovými systémy Bower a npm.

Dokumentace je také na stejné úrovni. Je v ní obsaženo vše potřebné pro implementaci i následující vývoj včetně informací o použitém JavaScriptu, CSS preprocesoru a kaskádových stylech týkajících se především komponent. Ovšem schází instruktážní videa a možnost zkoušení kódu jednotlivých komponent online

Hodnocení: 8 bodů

Customizace

Kromě manuální editace kaskádových stylů je customizace možná pouze při použití varianty s CSS preprocesorem. Stejně jako je tomu u nástrojů Skeleton a Kube. Jiné možnosti nejsou dostupné, například ani editace frameworku na domovské webové stránce a stažení upravené verze.

Hodnocení: 4 body

Mřížka

Také co se týče mřížky, je nástroj Materialize shodný s dvěma předchozími frameworky. Rovněž má ve výchozím nastavení dostupný pouze dvanáctisloupcový grid systém. Ten ale nelze lehce editovat. Není tedy možné jednoduše změnit defaultní počet sloupců ani pomocí CSS preprocesoru.

Hodnocení: 7 bodů

Zlomové body

Ve výchozím řešení frameworku Materialize jsou celkem tři Breakpointy, stejně jako je tomu u konkurenčních nástrojů Kube a Bootstrap, ale s tím rozdílem, že první zlomový

bod je zvolen dle řešitele téměř ideálně, a to na hodnotě 600px. Zbylé dva zlomové body jsou definovány na pozici 992px a 1200px, tedy také ideálně.

Ačkoli není možné jednoduše zlomové body editovat pro klasickou CSS verzi frameworky, ale pouze u varianty s preprocesorem, a přidávání je složité, uživatel nepovažuje tento nedostatek za velkou slabinu jako u konkurentů Kube nebo Foundation, jelikož počet breakpointů a jejich rozložení je prakticky ideální. (viz. obrázek č. 12)

Hodnocení: 9 bodů

Navigace

Responzivní navigace je také řešena pomocí schování položek pod panel, který se zobrazí po kliknutí na tlačítko. Tato operace je řešena stejně jako u nástroje Kube vyjetím z boku obrazovky a nikoli svrchu. Druhou zmiňovanou alternativu nelze jednoduše definovat a chybí možnost změny rozložení jednotlivých položek menu.

Hodnocení: 7 bod

Responzivita obrázků

Responzivita obrázků je u frameworku Materialize řečena pomocí přiřazení třídy *img-responsive*, která způsobí proměnlivost velikosti objektu v závislosti na rozměrech obalového elementu.

Nedochází tedy k žádné datové úspoře, jelikož je používán jeden objekt s variabilní velikostí pro všechna zařízení. U obrázků s velkou datovou náročností je proto vhodné používat raději značku *picture* nebo nové atributy elementu *img*, než defaultní možnosti.

Hodnocení: 4 body

Carousel

Carousel je v Materialize řešen pomocí stejnojmenné komponenty. Ta nabízí, stejně jako Orbit u Foundation, více efektů. A navíc kromě obvyklých nedostatků, například problematického zobrazení titulků, se zde nevyskytuje slabina přizpůsobení výšky při změně režimu z portrait do landscape a naopak, jako je tomu u Orbitu.

Hodnocení: 9 bodů

Podpora prohlížečů

Podpora je zaručena pro všechny aktuálně používané webové prohlížeče, podle kterých je toto kritérium hodnoceno. Internet Explorer od společnosti Microsoft je podporován už od verze 9.

Hodnocení: 9 bodů

Glyfy

Počet: 932 ks

4.1.3 Ohodnocení variant dle kritérií

Klasifikace variant podle jednotlivých hledisek je jedním ze stěžejních úkonů u vícekritériální analýzy variant. Samotné ohodnocení dle kritérií je uvedeno v předchozí kapitole u každého frameworku. V této kapitole je tedy v tabulce č. 5 komplexní souhrn hodnocení, které je založeno především na zkušenostech řešitele s vybranými nástroji.

Tabulka 5: Ohodnocení variant dle jednotlivých kritérií

X			Kritérium									
			1	2	3	4	5	6	7	8	9	10
Varianta	1	Bootstrap	177	8	10	10	8	7	5	8	9	260
	2	Foundation	293	10	7	10	6	8	10	6	9	170
	3	Skeleton	11	7	4	7	9	1	4	1	9	0
	4	Kube	87	7	4	7	5	7	1	1	7	0
	5	Materialize	326	8	4	7	9	7	4	9	9	932

Zdroj: Vlastní zpracování

4.1.4 Stanovení vah kritérií

Výsledné údaje ukazuje tabulka 6. Normalizované hodnoty váhy všech kritérií jsou vypočteny pomocí Saatyho metody párového srovnání a následně použity ve výpočtech metodami AHP i váženého součtu.

Stanovení preference a dispreference jednotlivých kritérií je založeno na vlastních zkušenostech řešitele. Jak je patrné ve výsledné tabulce, mají pro řešitele v této analýze největší váhu kritéria 4 (*mřížka*), poté 3 (*customizace*) a 5 (*zlomové body*).

Tabulka 6: Ohodnocení variant dle jednotlivých kritérií

X	Kritérium										Váha Kritéria	Normalizovaná hodnota váhy kritéria	
	1	2	3	4	5	6	7	8	9	10			
Kritérium	1	1	1/7	1/9	1/9	1/7	1/7	1/3	1/3	1/5	3	0,274	0,017
	2	7	1	1/3	1/3	1/3	1/3	7	7	5	9	1,690	0,105
	3	9	3	1	1/3	1	3	7	7	5	9	3,002	0,186
	4	9	3	3	1	3	5	7	7	5	9	4,393	0,272
	5	7	3	1	1/3	1	3	7	7	5	9	2,928	0,181
	6	7	3	1/3	1/5	1/3	1	7	7	5	9	2,001	0,124
	7	3	1/7	1/7	1/7	1/7	1/7	1	3	1/3	5	0,496	0,031
	8	3	1/7	1/7	1/7	1/7	1/7	1/3	1	1/3	5	0,398	0,025
	9	5	1/5	1/5	1/5	1/5	1/5	3	3	1	7	0,795	0,049
	10	1/3	1/9	1/9	1/9	1/9	1/9	1/5	1/5	1/7	1	0,178	0,011
$\Sigma =$											16,156	1	

Zdroj: Vlastní zpracování

4.1.5 Analýza metodou AHP

Základem této metody je přerozdělení jednotlivých normalizovaných vah kritérií na jednotlivé varianty. Tedy porovnat framework každý s každým postupně v rámci všech kritérií. Pro přerozdělení vah je využita Saatyho metoda párového porovnání. Preference a dispreference jsou určeny na základě ohodnocení nástrojů z tabulky č. 5. Pro příklad jsou znázorněny komparace jednotlivých variant podle kritéria 1 (*velikost frameworku*) v tabulce č. 7 a 2 (*implementace*) v tabulce č. 8.

Tabulka 7: Komparace variant podle kritéria 1 (*velikost frameworku*)

Kritérium 1 = 0,017	Varianta					Váha varianty	Normalizovaná hodnota váhy varianty	Normalizovaná hodnota váhy varianty podle kritéria 1	
	1	2	3	4	5				
Varianta	1	1	5	1/7	1/5	7	1,000	0,116	0,002
	2	1/5	1	1/9	1/7	3	0,394	0,046	0,001
	3	7	9	1	3	9	4,427	0,516	0,009
	4	5	7	1/3	1	9	2,537	0,295	0,005
	5	1/7	1/3	1/9	1/9	1	0,226	0,026	0,000
$\Sigma =$						8,584	1	0,017	

Zdroj: Vlastní zpracování

Tabulka 8: Komparace variant podle kritéria 2 (*implementace*)

Kritérium 2 = 0,105	Varianta					Váha varianty	Normalizovaná hodnota váhy varianty	Normalizovaná hodnota váhy varianty podle kritéria 2	
	1	2	3	4	5				
Varianta	1	1	1/3	2	2	1	1,059	0,177	0,019
	2	3	1	4	4	3	2,702	0,453	0,047
	3	1/2	1/4	1	1	1/2	0,574	0,096	0,010
	4	1/2	1/4	1	1	1/2	0,574	0,096	0,010
	5	1	1/3	2	2	1	1,059	0,177	0,019
$\Sigma =$						5,969	1	0,105	

Zdroj: Vlastní zpracování

Správnost výpočtů dokazuje, že suma součtu jednotlivých normalizovaných vah v rámci jednoho kritéria je rovna normalizované váze daného kritéria, které jsou vypočteny a uvedeny v tabulce č. 6. Například u kritéria 1 (*velikost frameworku*) je to hodnota 0,017.

Analogickým způsobem je provedena postupně komparace jednotlivých CSS frameworků podle všech deseti kritérií, ale pro úsporu místa je uvedena pouze tabulka č. 9, kde jsou vyobrazeny výsledné normalizované váhy jednotlivých variant v rámci kritérií.

Tabulka 9: Souhrnná tabulka normalizovaných vah jednotlivých variant podle všech kritérií

X	Kritérium										
	1	2	3	4	5	6	7	8	9	10	
Varianta	1	0,002	0,019	0,105	0,099	0,036	0,025	0,005	0,009	0,011	0,003
	2	0,001	0,047	0,043	0,099	0,015	0,045	0,018	0,005	0,011	0,001
	3	0,009	0,010	0,013	0,025	0,060	0,004	0,003	0,001	0,011	0,001
	4	0,005	0,010	0,013	0,025	0,010	0,025	0,001	0,001	0,005	0,000
	5	0,000	0,019	0,013	0,025	0,060	0,025	0,003	0,009	0,011	0,006

Zdroj: Vlastní zpracování

Po sečtení normalizovaných vah určité varianty podle všech kritérií je dosažena výsledná hodnota váhy dané varianty. Nástroj, který má výslednou váhu nejvyšší, je vyhodnocen jako nejlépe hodnocená kompromisní varianta. Výsledné pořadí je uvedeno v tabulce č. 10.

Tabulka 10: Výsledné pořadí analýzy metodou AHP

Varianta	Výsledná hodnota váhy varianty	Pořadí
Bootstrap	0,313	1.
Foundation	0,285	2.
Skeleton	0,136	4.
Kube	0,095	5.
Materialize	0,171	3.
$\Sigma =$	1	-

Zdroj: Vlastní zpracování

Jak je patrné, dle této metody je nejlépe klasifikován framework Bootstrap těsně před nástrojem Foundation. Mezi následující variantou v pořadí Materialize je poměrně značný rozestup. Poté následují Skeleton a Kube.

4.1.6 Analýza metodou váženého součtu

Základem u analýzy metodou váženého součtu je určení ideální a bazální varianty podle všech kritérií. V případě prvního kritéria, tedy *velikost frameworku*, je ideální varianta

nejnižší možná hodnota a bazální naopak nejvyšší. U ostatních kritérií je situace opačná. Přehled je uveden v tabulce č. 11.

Tabulka 11: Ideální a bazální varianty

X	Kritérium									
	1	2	3	4	5	6	7	8	9	10
Ideální varianta	11	10	10	10	9	8	10	9	9	932
Bazální varianta	326	7	4	7	5	1	1	1	7	0

Zdroj: Vlastní zpracování

Pomocí hodnot ideálních a bazálních variant je možné vypočítat standardizovanou kritériální matici R, kde jsou uvedeny dílčí užitky každé z variant dle jednotlivých kritérií (viz tabulka č. 12). Pro ilustraci je uveden výpočet dílčího výpočtu varianty 1 podle kritéria 1 dosazením do vzorce (viz. literární rešerše):

$$r_{ij} = \frac{177 - 326}{11 - 326} = 0,473$$

Hodnota 177 označuje hodnotu varianty 1 podle kritéria 1, 326 hodnotu bazální a 11 ideální. Číslo 0,473 značí výsledek, což je dílčí užitek varianty 1 dle kritéria 1 (viz. tabulka č. 12 – 1. řádek, 1. sloupec)

Tabulka 12: Standardizovaná kritériální matice R

X		Kritérium									
		1	2	3	4	5	6	7	8	9	10
Varianta	1	0,473	0,333	1,000	1,000	0,750	0,857	0,444	0,875	1,000	0,279
	2	0,105	1,000	0,500	1,000	0,250	1,000	1,000	0,625	1,000	0,182
	3	1,000	0,000	0,000	0,000	1,000	0,000	0,333	0,000	1,000	0,000
	4	0,759	0,000	0,000	0,000	0,000	0,857	0,000	0,000	0,000	0,000
	5	0,000	0,333	0,000	0,000	1,000	0,857	0,333	1,000	1,000	1,000

Zdroj: Vlastní zpracování

Pro získání celkových užiteků jednotlivých frameworku je nutné provést skalární součin dílčích užiteků ze standardizované kritériální matice R a normalizovaných vah kritérií. Pro příklad je uveden postup pro výpočet celkového užtku nástroj Bootstrap:

$$0,473 * 0,017 + 0,333 * 0,105 + 1 * 0,186 + 1 * 0,272 + 0,6 * 0,181 + 0,857 * 0,124 + 0,556 * 0,031 + 0,875 * 0,025 + 1 * 0,049 + 0,279 * 0,011 \doteq 0,806$$

Výsledné celkové užítky všech nástrojů, a tedy i pořadí metodou váženého součtu, jsou uvedeny v tabulce č. 13. Jak je patrné, pořadí je stejné jako u metody AHP. Oběma metodami je tedy dosaženo stejných výsledků.

Tabulka 13: Výsledné pořadí analýzy metodou váženého součtu

Varianta	Celkový užitek variant	Pořadí
Bootstrap	0,830	1.
Foundation	0,738	2.
Skeleton	0,258	4.
Kube	0,119	5.
Materialize	0,417	3.

Zdroj: Vlastní zpracování

4.2 Aplikace vybraného nástroje

Analýzou oběma metodami je vyhodnocena jako nejlépe hodnocená kompromisní varianta framework Bootstrap, proto pro aplikaci na webových stránkách je vybrán právě avizovaný nástroj. Pro úsporu místa je ve všech následujících kódech v elementech, kde se opakuje kód pouze s obměnami textu, názvu ikon apod., nahrazen tečkami.

Pomocí výchozí klasické verze Bootstrapu je vytvořena smyšlená webová prezentace Clubu C s prostým rozložením v tzv. one page designu dle grafického návrhu (viz příloha č. 1). V následujících podkapitolách není popsán postup tvorby webových stránek, ale pouze implementace, aplikace vybraných komponent a customizace elementů. V těchto procesech řešitel poukazuje na důležité skutečnosti nebo odstraňuje vybrané nedostatky, které se vyskytují v klasické defaultní variantě frameworku.

4.2.1 Implementace

Jelikož se jedná o jednoduchý web, je dostačující klasická verze frameworku Bootstrap a není nutné implementovat variantu s jedním CSS preprocesorů. Pro implementaci tedy stačí stáhnout zdrojové soubory kaskádových stylů a JavaScriptu z domovské stránky a odkázat na ně v HTML dokumentu.

```
<!-- Bootstrap Core JavaScript -->
<script src="js/bootstrap.min.js"></script>

<!-- Bootstrap Core CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet">
```

4.2.2 Mřížka a zlomové body

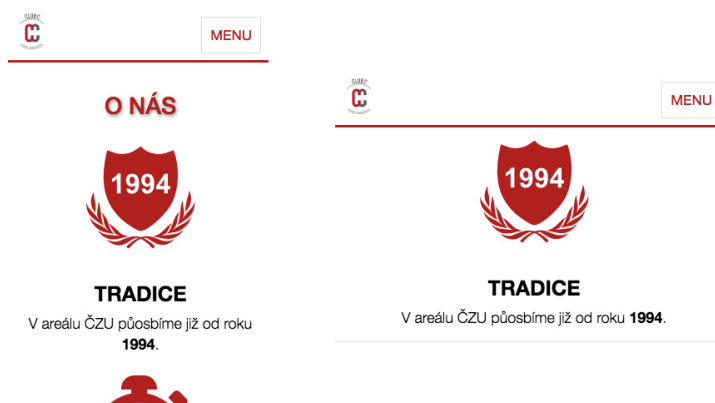
U webové prezentace je použitý nejrozšířenější styl rozložení a je tedy využita výchozí dvanáctisloupcová mřížka. Z tohoto důvodu není nutné defaultní grid systému nijak upravovat.

Jak již bylo řečeno u analýzy, zlomové body jsou ve výchozím nastavení obsaženy tři, konkrétně 768px, 992px a 1200px, což umožňuje vytvářet odlišný layout pro čtyři různá zařízení (0-767px, 768-991px, 992-1199px, 1200px a více). Jednotlivé intervaly jsou definovány přiřazením tříd. Pro nejmenší *col-xs-X*, následuje *col-sm-X*, *col-md-X* a pro největší třída *col-lg-X*, kde X je číslo, jež označuje počet sloupců, které mají být použity z výchozí mřížky pro jeden sloupec na určité velikosti displeje. Tedy například přiřazením třídy *col-sm-6* je vzato pro jeden sloupec 6 sloupců z defaultní mřížky a tím pádem je na daném zařízení dvousloupcový layout, jelikož $12 \text{ (počet sloupců výchozí mřížky)} / 6 = 2$. Viz ukázka kódu níže, kde u tříd *col-lg-3* a *col-md-3* je vytvořen čtyřsloupcový layout ($12/3=4$), pro *col-sm-6* dvousloupcový a *col-xs-12* jednosloupcový.

```
<section id="about" class="about-section">
  <div class="container">
    <div class="row">
      <div class="col-xs-12">
        <h2 class="section-heading">0 nás</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-lg-3 col-md-3 col-sm-6 col-xs-12 text-center about-element">
        <div class="icon">
          <i class="icon-tradice"></i>
        </div>
        <h3 class="black">tradice</h3>
        <p class="black lead">V areálu ČZU působíme již od roku <span class="bold">1994</span>.</p>
      </div>
      <div class="col-lg-3 col-md-3 col-sm-6 col-xs-12 text-center about-element">
        .
      </div>
      <div class="col-lg-3 col-md-3 col-sm-6 col-xs-12 text-center about-element">
        .
      </div>
      <div class="col-lg-3 col-md-3 col-sm-6 col-xs-12 text-center about-element">
        .
      </div>
    </div>
  </div>
</section>
```

Ovšem jak již řešitel avizoval v analýze, první zlomový bod má příliš vysokou hodnotu, což může působit problémy například při individualizaci obsahu pro mobilní telefony. V minulosti měly mobilní aparáty velikost displeje i v režimu landscape kupříkladu 480px, například iPhone 4. Kvůli těmto malým rozměrům nebylo nutné nijak individualizovat obsah pro telefony v režimech portrait a landscape (viz obrázek č. 21).

Obrázek 21: Zobrazení části vytvořené webové prezentace v režimech portrait i landscape na telefonu s malými rozměry obrazovky (konkrétně iPhone 4)



Zdroj: Vlastní zpracování

Ovšem s příchodem nových telefonů, například iPhone 7 nebo 8, které mají šířku displeje v režimu portrait 375px a landscape módu 667px, je vhodné přizpůsobovat obsah zvlášť, jelikož při orientaci na šířku mají velikost obrazovky téměř stejnou jako tablety v režimu portrait. Tato skutečnost může způsobit nevhodné zobrazení na většině momentálně vyráběných mobilních telefonech v jednom z avizovaných módů, což je možné vidět na obrázku č. 22. Rozvržení elementů je vytvořeno zdrojovým kódem uvedeným výše pomocí třídy *col-xs-12*. Na mobilním telefonu při orientaci na šířku se vyskytuje příliš prázdného nevyužitého prostoru po stranách.

Obrázek 22: Vhodné zobrazení třídou *col-xs-12* v režimu portrait (vlevo) a nevhodné v režimu landscape (vpravo) na mobilním telefonu iPhone 7



Zdroj: Vlastní zpracování

Při změně použití třídy z *col-xs-12* na *col-xs-6* se sice vytvoří z jednosloupcového layoutu dvousloupcový u mobilních telefonů, což je vhodnější pro režim landscape, ale problém na

druhou stranu nastává u orientace na výšku, kde je naopak málo místa pro tento styl, jak je patrné na obrázku č. 23.

Obrázek 23: Nevhodné zobrazení třídou *col-xs-6* v režimu portrait (vlevo) a vhodné v režimu landscape (vpravo) na mobilním telefonu iPhone 7



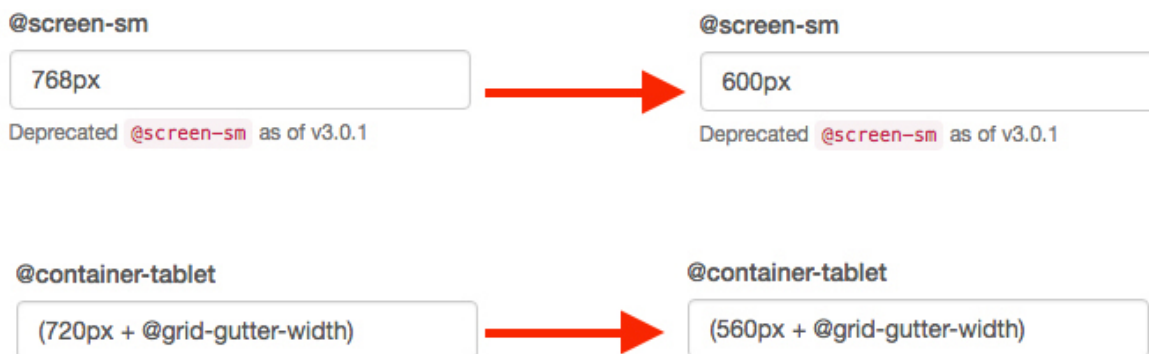
Zdroj: Vlastní zpracování

Na obrázku č. 23 je možné vypožorovat, že jednotlivé elementy jsou v režimu portrait rozmístěny příliš blízko u sebe. V tomto případě lze tento problém řešit dotazy na médium definováním jiných velikostí textu, ikon, odsazení a dalších vlastností. Ale například u portfolií nebo fotogalerií je tento postup nevhodný, jelikož by příliš malý náhled rastrů nebyl téměř čitelný. Proto je lepší pro tyto případy vhodně zvolit odlišné rozložení zlomových bodů, a tak jednoduše individualizovat rozložení webové prezentace pro mobilní telefony větších rozměrů v režimu portrait.

Dle obrázku č. 12 – Ideální rozvržení zlomových bodů je vhodné zvolit první breakpoint na hodnotu 600px, jelikož dnešní mobilní telefony mají při orientaci na šířku minimálně právě avizovaný rozměr a tím je docíleno individualizace vzhledu při prohlížení webu v závislosti na orientaci momentálně vyráběných mobilních zařízení. Jejich režim portrait je pak stejný jako pro telefony s menšími rozměry bez ohledu na orientaci a landscape je spojen s tablety používanými na výšku.

Změny zlomových bodů u nástroje Bootstrap lze docílit, mimo CSS preprocesorů, i pomocí customizace editací některých položek ve formuláři, stažením upravené verze systému a nahráním nového souboru s kaskádovými styly. Na obrázku č. 24 jsou vyobrazeny dvě položky, které je nutné změnit pro editaci prvního bodového zlomu na hodnotu 600px.

Obrázek 24: Editace prvního zlomového bodu



Zdroj: Vlastní zpracování

Po této editaci jsou intervaly pro zobrazení 0-599: *col-xs* a 600-991: *col-sm*. Tím je docíleno avizované individualizace rozložení u mobilních telefonů. Režim portrait je stejný se zařízeními menších rozměrů bez ohledu na orientaci a landscape mód je spojen s tablety v portrait režimu. Po použití tříd *col-xs-12* a *col-sm-6* je odlišný layout zobrazen na obrázku č. 25.

Obrázek 25: Odlišný layout na mobilních telefonech po změně prvního zlomového bodu v režimech portrait vytvořený třídou *col-xs-12* a landscape vytvořený třídou *col-sm-6*



Zdroj: Vlastní zpracování

4.2.3 Navigace

Navigace je vytvořena dle dokumentace (zdrojový kód uveden pod odstavcem). U navigačního panelu není vytvořena žádná třída navíc, která není důležitá dle oficiálních webových stránek nástroje. Vzhledu zobrazeného na obrázku č. 26 je docíleno pouze úpravou kaskádových stylů výchozích tříd.

```

<nav class="navbar navbar-default navbar-fixed-top" role="navigation">
  <div class="container">
    <div class="navbar-header page-scroll">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-ex1-collapse">
        <span class="sr-only">Toggle navigation</span>
        <span>MENU</span>
      </button>
      <a class="navbar-brand page-scroll" href="#page-top">
        
      </a>
    </div>
    <div class="collapse navbar-collapse navbar-ex1-collapse">
      <ul class="nav navbar-nav">
        <li>
          <a class="page-scroll" href="#about">0 nás</a>
        </li>
        .
        .
      </ul>
    </div>
  </div>
</nav>

```

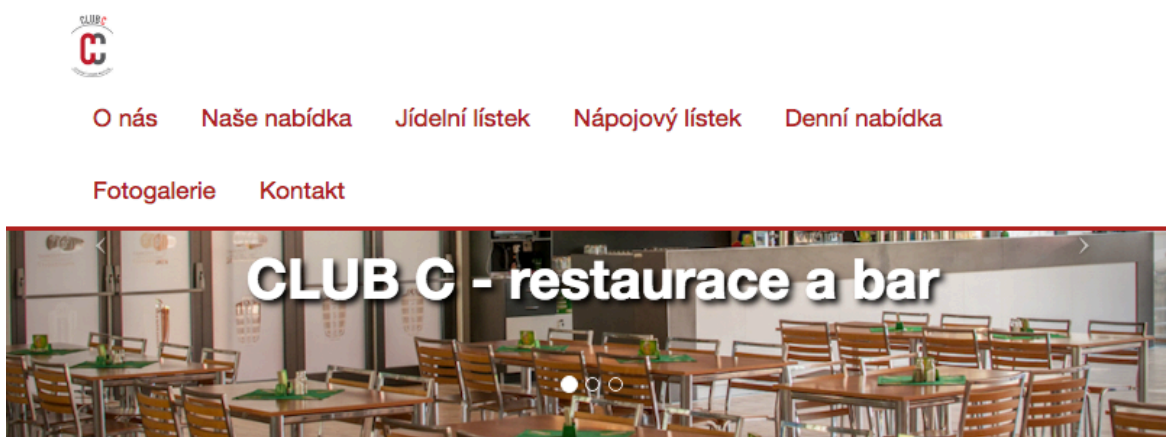
Obrázek 26: Design responzivní navigace na desktopových zařízeních



Zdroj: Vlastní zpracování

Ovšem jak je patrné na obrázku č. 27, dochází ke špatnému zobrazení na mobilních zařízeních s většími rozměry displeje, například iPhone X v režimu landscape, jelikož ke schování navigačního panelu dochází na hodnotě prvního zlomového bodu a nižší, což je 768px ve výchozí variantě, respektive 600px v upravené verzi frameworku řešitele, ale iPhone X má rozměr při orientaci na šířku 812px. Proto je nutné navýšit hodnotu pro schování navigačního panelu, aby nedocházelo k nežádoucímu jevu špatného zobrazení.

Obrázek 27: Špatné zobrazení responzivní navigace na iPhone X v režimu landscape



Zdroj: Vlastní zpracování

U systému Bootstrap je zmiňovaná změna možné provést opět pomocí formuláře pro customizaci. Postup je tedy obdobný s individualizací zlomových bodů. Pouze se edituje jiná položka, konkrétně `@grid-float-breakpoint`. Ta označuje bod, při kterém je navigační panel schován. Výchozí hodnota je první zlomový bod, který je nutné navýšit. První možností je změna na druhý, potažmo třetí, breakpoint, nebo je možné definovat jakoukoli číselnou hodnotu, například 800px. Autor práce v tomto případě situaci řeší editací na druhý zlomový bod, tedy v tomto případě 992px (viz. obrázek č. 28).

Obrázek 28: Editace bodu pro schování navigačního panelu



Zdroj: Vlastní zpracování

Po avizované editaci zlomového bodu je docíleno požadované změny. Tedy navigační panel je schován na hodnotě druhého zlomového bodu a nižší. K dané změně dochází již při rozměru obrazovky 992px a nižší. V tomto intervalu se nachází i 812px, tedy velikost displeje při orientaci na šířku iPhone X, na kterém je problém s navigačním panelem demonstrován (viz. obrázek č. 29)

Obrázek 29: Správné zobrazení responzivní navigace na iPhone X



Zdroj: Vlastní zpracování

4.2.4 Carousel

Pro carousel je použita stejnojmenná komponenta nástroje Bootstrap, tedy s názvem Carousel. Je vytvořen dle dokumentace pomocí zdrojového kódu, který je uveden pod odstavcem.

```

<header id="myCarousel" class="carousel slide carousel-fade">
  <!-- Indicators -->
  <ol class="carousel-indicators">
    <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
    <li data-target="#myCarousel" data-slide-to="1"></li>
    <li data-target="#myCarousel" data-slide-to="2"></li>
  </ol>

  <!-- Wrapper for Slides -->
  <div class="carousel-inner">
    <div class="item active">
      <!-- Set the first background image using inline CSS below. -->
      <div class="fill" style="background-image:url('img/carousel/slide1.jpg');"></div>
      <div class="carousel-caption">
        <h1>
          CLUB C - restaurace a bar
        </h1>
      </div>
    </div>
    <div class="item">
      . . .
    </div>
    <div class="item">
      . . .
    </div>
  </div>

  <!-- Controls -->
  <a class="left carousel-control" href="#myCarousel" id="left-arrow" data-slide="prev">
    <span class="icon-prev"></span>
  </a>
  <a class="right carousel-control" href="#myCarousel" data-slide="next">
    <span class="icon-next"></span>
  </a>
</header>

```

Pro carousel je použitý pouze základní efekt. U této komponenty je možné nastavit i automatické spuštění prolínání mezi jednotlivými slidy a i čas mezi přechody skriptem uvedeným pod odstavcem. Hodnota 3000 znamená dobu mezi přechody 3000ms, tedy 3 vteřiny.

```

<script>
  $(' .carousel').carousel({
    interval: 3000
  })
</script>

```

Jak již autor zmiňoval při analýze, tento nástroj neumí automaticky přizpůsobovat velikost titulku u jednotlivých prvků carouselu. Proto je nutné tento problém ošetřit, například dotazy na médium. V kódu kaskádových stylů pod odstavcem je uvedeno řešení autorem práce, a to zmenšením velikosti textu v titulku. Jak je patrné, řešitel využil strategii desktop first.

```

.carousel-caption h1 {
  color: white;
  text-shadow: 3px 3px 6px rgba(0,0,0,1);
  line-height: 1.2;
  font-weight: bold;
  font-size: 50px;
}

@media (max-width: 991px) {
  .carousel-caption h1 {
    font-size: 38px;
  }
}

```

4.2.5 Responzivita obrázků

Jelikož framework Bootstrap nenabízí efektivní řešení pro responzivitu rastrů, je vhodné zvolit některou z možností v HTML5. Řešitel pro tuto konkrétní webovou prezentaci využívá značku *picture*. Konkrétně použití demonstruje na rastru v sekci nazvané Kontaktní informace pod logem Club C (viz. příloha č. 1)

Na zdrojovém kódu pod odstavcem je patrné, že autor práce pro použití různých verzí rastru využívá hodnoty, které jsou identické se zlomovými body této webové prezentace. Tedy pro velikost dispeje do 600px je stažen ze serveru rastr s nejmenšími rozměry s názvem *contact_xs.jpg*, do 992px *contact_sm.jpg*, do 1200px *contact_md.jpg* a nad 1200px *contact_lg.jpg*. Poslední jmenovaný rastr je použit jako výchozí, a proto se uživatelům, kteří používají některé ze starších verzí prohlížečů nepodporující značku *picture*, zobrazí právě on bez závislosti na rozměrech používané obrazovky.

```

<div class="row">
  <div class="col-lg-12">
    <picture >
      <source media="(max-width: 600px)" srcset="img/contact_xs.jpg">
      <source media="(max-width: 992px)" srcset="img/contact_sm.jpg">
      <source media="(max-width: 1200px)" srcset="img/contact_md.jpg">
      
    </picture>
  </div>
</div>

```

4.3 Otestování vytvořené webové prezentace

Vytvořená webová prezentace Clubu C je otestována na všech aktuálních verzích prohlížečů, které byly zohledňovány v analýze, a browseru Internet Explorer 8. Pro ten je ze strany tvůrců Bootstrapu také garantována podpora. Na všech prohlížečích je vzhled

a chování webové stránky identické (viz. přílohy č. 2 až 5) kromě Internetu Explorer 8. Sice pomocí knihovny Respond.js jsou funkční dotazy na médium a tím pádem je dodrženo rozložení celé stránky, ale kvůli neúplné podpoře kaskádových stylů, konkrétně nejnovější verze CSS3, není v avizovaném prohlížeči docíleno identického vzhledu. Například není podporována vlastnost *border-radius*, pomocí které je docíleno kruhového podkladu pod ikonami v sekci Naše nabídka (viz. příloha č. 1). Z důvodu absence podpory avizované vlastnosti je toto pozadí čtvercového tvaru (viz. obrázek č. 30).

Obrázek 30: Špatné zobrazení webové prezentace ve webovém prohlížeči IE 8



Zdroj: Vlastní zpracování

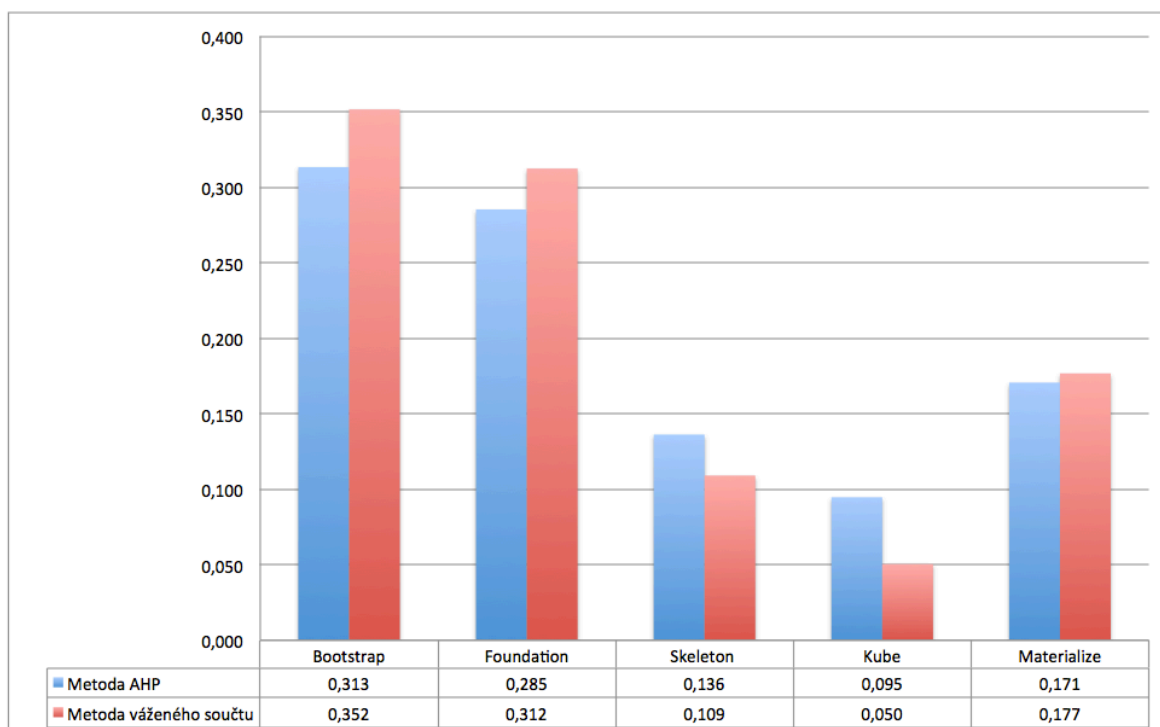
Dále nejsou podporovány značky používané od verze HTML5. Například tag *picture* pro responzivitu obrázků. Z toho důvodu je v sekci nazvané Kontaktní informace pod logem Club C (viz. příloha č. 1) používán vždy výchozí rastr bez závislosti na rozměrech používané obrazovky. V tomto případě se jedná o obrázek s největšími rozměry. Ovšem tuto skutečnost nepovažuje řešitel za příliš velký problém, jelikož browser IE 8 je výhradně pro desktopy, zatímco tematika flexibility obrázků se řeší především z důvodu úspory dat na mobilních zařízeních.

Oba zmiňované nedostatky týkající se IE 8 nepovažuje autor práce za chyby samotného nástroje, jelikož nijak závažně neovlivňují responzivní design webových stránek, což je primární účel front-end CSS frameworků.

5 Výsledky a diskuse

Výsledky jsou prezentovány jako porovnání konečných hodnot analýzy oběma metodami. Pro porovnání skóre byly dodatečně hodnoty metody váženého součtu znormalizovány, tedy celkový užitek každé varianty byl vydělen jejich součtem. Výsledky porovnání jsou zobrazeny v tabulce č. 14 doplněné grafem.

Tabulka 14: Výsledné pořadí analýzy metodou váženého součtu



Zdroj: Vlastní zpracování

Na základě analýzy provedené řešitelem pomocí obou metod jsou jednoznačně určeny jako nejlépe hodnocené kompromisní varianty Bootstrap na prvním místě a Foundation na druhém. Rozestup mezi prvně jmenovanými nástroji a ostatními softwary, tedy Materialize na třetím, Skeleton na čtvrtém a na pátém místě Kube, je poměrně značný.

Dle názoru řešitel je prvním důležitým faktorem vítězství Bootstrapu především nejlepší možné ohodnocení podle kritéria customizace, které má druhou nejvyšší normalizovanou hodnotu váhy.

Dalším důležitým činitelem určení jako nejlépe hodnocené kompromisní varianty právě softwarové struktury pro designování webových stránek Bootstrap je patrně velice vysoké

hodnocení v závislosti na kritériu zlomové body. Ačkoli zmiňovaný nástroj neobsahuje ve výchozí variantě všechny breakpointy na ideálních pozicích, je velice jednoduchá editace i v klasické verzi bez CSS preprocesoru, což je velké pozitivum například oproti největšímu konkurentu frameworku Foundation a avizované kritérium má třetí nejvyšší normalizovanou hodnotu váhy.

Jak je již uvedeno v odstavci výše, řešitel považuje za největšího oponenta nejlépe vyhodnoceného nástroje softwarovou strukturu Foundation, která nakonec obsadila druhou příčku oběma metodami, ale výsledky nejsou tak těsné, jak byly očekávány.

Menší rozestup v rezultátech byl očekáván především z důvodu několika jednoznačně nejlepších hodnocení ze všech variant dle několika kritérií. Ovšem vyšší známky než konkurenční nástroje obdržel framework Foundation především pouze u kritérií s nízkou normalizovanou hodnotou váhy, například responzivita obrázků, což je sice jeden z pilířů responzivního designu, ale pomocí HTML5 jde tato problematika řešit velice jednoduše i bez CSS frameworku. Proto patrně autor práce při Saatyho metodě párového porovnání avizované kritérium spíše dispreferoval, z čehož vznikla poměrně nízká normalizovaná hodnota váhy, konkrétně 0,031.

Po shrnutí čtyř předchozích odstavců lze tedy říci, že Foundation, největší konkurent nástroje Bootstrap, je výrazně lépe ohodnocen pouze dle kritérií s nízkou normalizovanou hodnotou váhy. Naopak u důležitých kritérií jsou avizované nástroje vyrovnané nebo je s vyšším hodnocením klasifikován framework Bootstrap. Tato skutečnost má s největší pravděpodobností za následek vítězství nástroje od společnosti Twitter s ne tak těsným rozdílem, jaký byl očekáván.

Největším překvapením ve výsledcích je jednoznačně poslední místo frameworku Kube, který má být dle názoru autora systému velice schopný a konkurenceschopný s nejlepšími nástroji na poli frameworků pro designování responzivních webových prezentací. Nakonec se ale před ním umístil i Skeleton, který je i dle samotného vývojáře označován jako minimalistická softwarová struktura zaměřující se na jednoduchost a pouze základní funkce responzivního designu.

Špatné umístění nástroje Kube je dle autora způsobeno především nízkým hodnocením podle stěžejních kritérií, například customizace nebo zlomové body, ale také i v absenci

řešení pro responzivitu obrázků nebo carousel. Ačkoli mají tyto dvě poslední jmenovaná kritéria nízkou normalizovanou hodnotu váhy, totální absence pro řešení zmiňovaných prvků má samozřejmě za následek klasifikaci hodnotou 1. Tudíž nejnižší možné hodnocení, které se patrně projevuje i navzdory relativní nedůležitosti avizovaných kritérií.

Přímá konfrontace výsledků dosažených výzkumem, který provedl řešitel, a dříve publikovaných je prakticky nemožná, protože téměř v žádných nalezených zdrojích se nevyskytuje komparace právě těchto pěti softwarů, ale pouze porovnání například dvou nebo tří s jinými systémy.

Ovšem velice často se ve výzkumech vyskytují frameworky Bootstrap a Foundation. Ty jsou dle těchto avizovaných studií ve většině případů klasifikovány nejlépe, ale výsledky jsou nejednoznačné. V jednom případě je lépe vyhodnocen Bootstrap a v další analýze je naopak lepší Foundation, což lze považovat za srovnatelné výsledky dosažené řešitelem, jelikož také v této analýze jsou nejlépe vyhodnoceny oba probírané nástroje pro tvorbu responzivního designu webových stránek a poměrně s malým rozdílem ve výsledcích ve prospěch frameworku Bootstrap od společnosti Twitter v případě obou provedených metod vícekritériální analýzy variant, tedy AHP a váženého součtu.

6 Závěr

Diplomová práce byla tematicky zaměřená na tvorbu responzivního designu webových stránek. Tato problematika je v posledních letech stále více aktuální zejména od příchodu chytrých telefonů. Popularita tohoto odvětví je umocněna zlepšujícím se mobilním internetovým připojením a z toho vyplývající možnosti procházení webových stránek kdekoli a téměř na jakémkoli současném mobilním zařízení.

Cílem práce bylo charakterizovat CSS frameworky, zmapovat jejich současnou nabídku, analyzovat je a nakonec aplikovat včetně otestování vybraného nástroje pro tvorbu webových stránek. Analýza a výběr byly prováděny pomocí vícekritériální analýzy variant, konkrétně metodami AHP a váženého součtu podle kritérií.

Na základě zvolených kritérií byly klasifikovány jako nejlépe hodnocené kompromisní varianty frameworky Bootstrap a Foundation. Tyto dva nástroje dosáhly velmi dobrého hodnocení, ale prvně jmenovaný byl s malým rozdílem klasifikován lépe oběma metodami, kterými byla prováděna vícekritériální analýza, tedy AHP a váženého součtu. Proto byl zvolen jako nejlépe hodnocenou kompromisní variantou právě avizovaný nástroj.

Vybraný nástroj Bootstrap byl následně aplikován pro tvorbu smyšlených ilustrativních webových stránek Clubu C s prostým rozložením v tzv. one page designu dle grafického návrhu (viz přílohy č. 1-5). V kapitolách týkajících se aplikace nástroje autor práce nepopisuje postup tvorby webových stránek, ale pouze implementaci, aplikaci vybraných komponent a customizaci elementů. V těchto procesech řešitel poukazuje na důležité skutečnosti nebo odstraňuje některé nedostatky, které se vyskytují v klasické defaultní neupravené variantě frameworku Bootstrap.

Dále bylo provedeno testování vytvořené webové prezentace Clubu C s použitím webových prohlížečů, které byly brány v potaz při samotné vícekritériální analýze dle jednoho z kritérií. Na všech prohlížečích byly vzhled a chování webové prezentace téměř identické kromě Internetu Explorer 8. Sice pomocí knihovny Respond.js byly funkční dotazy na médium a tím pádem bylo dodrženo rozložení celé stránky, ale kvůli neúplné podpoře kaskádových stylů, konkrétně nejnovější verze CSS3, nebylo v avizovaném prohlížeči docíleno identického vzhledu. Vybraná alternativa tedy splňovala ve své

podstatě všechny požadavky nástroje pro tvorbu responzivního designu webových prezentací.

7 Použitá literatura

1. DAWSON, Alexander. *Výjimečný webdesign: jak tvořit osobité, přitažlivé, použitelné weby*. Brno: Computer Press, 2012. ISBN 978-80-251-3719-2.
2. KADLEC, Tim. *Responzivní design profesionálně*. Brno: Zoner Press, 2014. Encyklopedie Zoner Press. ISBN 978-80-7413-280-3.
3. SHARKIE, Craig a Andrew FISHER. *Responzivní webdesign: okamžitě*. Brno: Computer Press, 2015. ISBN 978-80-2514-384-1.
4. MARCOTTE, Ethan. *Responsive Web design. 2e tir.* Paris: Eyrolles, 2011. ISBN 9782212133318.
5. Responsive Web Design. *Defsys* [online]. 2017 [cit. 2017-07-29]. Dostupné z: <https://www.defsys.com.au/wp-content/uploads/2017/01/responsive-web-design.jpg>.
6. Mobilní stránky nebo responzivní web. *Artweby* [online]. 2013 [cit. 2017-07-30]. Dostupné z: <https://www.artweby.cz/blog/mobilni-stranky-nebo-responzivni-web>.
7. Víte jaký je rozdíl mezi mobilním a responzivním webem. *Webdesign Plzeň* [online]. 2015 [cit. 2017-07-30]. Dostupné z: <http://www.webdesign-plzen.cz/clanky/vite-jaky-je-rozdil-mezi-mobilnim-a-responzivnim-webem>.
8. MARCOTTE, Ethan a [FOREWORD BY JEREMY KEITH]. *Responsive web design*. New York: A Book Apart, 2011. ISBN 9780984442577.
9. Mobile First VS Desktop First: How To Choose A Responsive Strategy. *Brainleaf* [online]. 2017 [cit. 2017-08-02]. Dostupné z: <https://www.brainleaf.com/blog/brainleaf-news/mobile-first-vs-desktop-first-how-to-choose-a-responsive-strategy/>.

10. Mobile first vs Desktop first. *Francisco Aguilera G.* [online]. 2015 [cit. 2017-08-02]. Dostupné z: <https://franciscoamk.com/mobile-first-vs-desktop-first/>.
11. Understanding the difference between mobile-first, adaptive and responsive design. *Frederic Gonzalo* [online]. 2017 [cit. 2017-08-03]. Dostupné z: <http://fredericgonzalo.com/en/2017/03/01/understanding-the-difference-between-mobile-first-adaptive-and-responsive-design/>.
12. LAZARIS, Louis. *CSS okamžitě*. Brno: Computer Press, 2014. ISBN 978-80-251-4176-2.
13. Délkové jednotky v CSS. *Jak psát web* [online]. 2017 [cit. 2017-08-23]. Dostupné z: <https://www.jakpsatweb.cz/css/css-jednotky.html>.
14. CSS pixel. *Vzhůru dolů* [online]. 2017 [cit. 2017-08-23]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/css-pixel>.
15. CASTRO, Elizabeth a Bruce HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
16. Layout (rozložení stránky) - Český HTML 5 manuál. *I Net work* [online]. 2013 [cit. 2017-08-04]. Dostupné z: <https://www.itnetwork.cz/html-css/html-manual/rozlozeni/html-layout-rozlozeni-stranky-cesky-manual/>.
17. GILLENWATER, Zoe Mickley. *Flexible web design: creating liquid and elastic layouts with CSS*. Berkeley, CA: New Riders, c2009. ISBN 978-0321553843.
18. GASSTON, Peter. *Moderní web*. Přeložil Ondřej BAŠE. Brno: Computer Press, 2015. ISBN 978-80-251-4345-2.

19. Webdesign - Mřížkové systému. *Kurzy grafiky* [online]. 2015 [cit. 2017-08-15]. Dostupné z: <http://www.kurzygrafiky.cz/tvorba-webu/135-webdesign-mrizkove-systemy>.
20. HOGAN, Brian P. *HTML5 a CSS3: výukový kurz webového vývojáře*. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1.
21. CASARIO, Marco. *CSS3 solutions: essential techniques for CSS3 developers*. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2012. ISBN 978-1-4302-4335-9.
22. GEARY, David M. *Core HTML5 canvas: graphics, animation, and game development*. Upper Saddle River, NJ: Prentice Hall, c2012. ISBN 978-0-13-276161-1.
23. Jak vytvořit responzivní design. *WhiteHat* [online]. 2017 [cit. 2017-08-20]. Dostupné z: <https://www.whitehat.cz/jak-vytvorit-responzivni-design/#more-161>.
24. Jaké breakpointy zvolit v responzivním webdesignu? *Vzhůru dolů* [online]. 2013 [cit. 2017-08-23]. Dostupné z: <http://kratce.vzhurudolu.cz/post/46416507703/jaké-breakpointy-zvolit-v-responzivn%C3%ADm-webdesignu>.
25. The 100% correct way to do CSS breakpoints. *FreeCodeCamp* [online]. 2016 [cit. 2017-08-23]. Dostupné z: <https://medium.freecodecamp.org/the-100-correct-way-to-do-css-breakpoints-88d6a5ba1862?gi=c1e8e6938a0>.
26. Screen resolution stats. *Statcounter* [online]. 2017 [cit. 2017-08-23]. Dostupné z: <http://gs.statcounter.com/screen-resolution-stats>.
27. Layout. *Get Bootstrap* [online]. 2017 [cit. 2017-08-23]. Dostupné z: <https://v4-alpha.getbootstrap.com/layout/overview/>.

28. GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. *HTML5 a CSS3 pro webové designéry*. Brno: Zoner Press, 2011. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.
29. DVOŘÁKOVÁ, Zdenka. *DTP a předtisková příprava: kompletní průvodce od grafického návrhu po profesionální tisk*. Vyd. 1. Brno: Computer Press, 2008, 288 s. ISBN 978-80-251-1881-8.
30. Obrázky. Jak psát web [online]. 2015 [cit. 2017-08-13]. Dostupné z: <https://www.jakpsatweb.cz/html/obrazky.html>.
31. (Snad už) Definitivní responzivní obrázky – srcset a sizes. *Zdroják* [online]. 2015 [cit. 2017-08-13]. Dostupné z: <https://www.zdrojak.cz/clanky/snad-uz-definitivni-responzivni-obrazky-srcset-sizes/>.
32. Srcset-sizes. *Vzhuru Dolů* [online]. 2017 [cit. 2017-08-14]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/srcset-sizes>.
33. HTML Tag. *W3 Schools* [online]. 2017 [cit. 2017-08-17]. Dostupné z: https://www.w3schools.com/tags/tag_picture.asp.
34. Nový HTML element by měl zrychlit web. *LinuxEXPRES* [online]. 2014 [cit. 2017-08-17]. Dostupné z: <https://www.linuxexpres.cz/novinky/novy-html-element-by-mel-zrychlit-web>.
35. Picture, nová značka pro vkládání obrázků. *Vzhůru dolů* [online]. 2017 [cit. 2017-08-20]. Dostupné z: <https://www.vzhurudolu.cz/prirucka/picture>.
36. CSS preprocesory: méně psaní, vyšší efektivita. *Zdroják* [online]. 2011 [cit. 2017-08-30]. Dostupné z: <https://www.zdrojak.cz/clanky/css-preprocesory-mene-psani-vyssi-efektivita/>.

37. Průvodce CSS preprocesory: který vybrat? *Vzhůru dolů* [online]. 2014 [cit. 2017-08-30]. Dostupné z: <https://www.vzhurudolu.cz/blog/15-css-preprocesory-4>.
38. Dynamický jazyk pro tvorbu stylesheetů. *LESS* [online]. 2012 [cit. 2017-08-30]. Dostupné z: <https://www.lesscss.cz>.
39. NETHERLAND, Wynn. *Sass and Compass in action*. Shelter Island, N.Y.: Manning, c2013. ISBN 978-1-617290-14-5.
40. Průvodce CSS preprocesory: co a jak? *Vzhůru dolů* [online]. 2014 [cit. 2017-08-30]. Dostupné z: <https://www.vzhurudolu.cz/blog/12-css-preprocesory-1>.
41. MILLS, Chris. *Practical CSS3: develop and design*. Berkeley, CA: Peachpit Press, 2013. Develop and design. ISBN 978-0-321-82372-4.
42. What is framework? *WhatIs.com* [online]. 2015 [cit. 2017-12-27]. Dostupné z: <http://whatis.techtarget.com/definition/framework>.
43. What are Frameworks? 22 Best Responsive CSS Frameworks for Web Design. *AWWWARDS* [online]. 2016 [cit. 2017-12-27]. Dostupné z: <https://www.awwwards.com/what-are-frameworks-22-best-responsive-css-frameworks-for-web-design.html>.
44. ŠUBRT, Tomáš. *Ekonomicko-matematické metody*. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, 2011, 351 s. ISBN 978-80-7380-345-2.
45. Moodle. *Kurz: Ekonomicko matematické metody I - KS - ZS 13/14* [online]. 2013 [cit. 2017-11-26]. Dostupné z: <https://moodle.czu.cz/course/view.php?id=805>
46. Browser & Platform Market Share. *W3Counter: Global Web Stats* [online]. 2017 [cit. 2018-01-02]. Dostupné z: <https://www.w3counter.com/globalstats.php>.

8 Seznam obrázků, tabulek a příloh

8.1 Obrázky

Obrázek 1: Schéma responzivního designu	14
Obrázek 2: Desktopová (vlevo) a mobilní (vpravo) verze webu společnosti Alza a.s.....	15
Obrázek 3: Verze klasická (vlevo) a optimalizovaná pro telefony (vpravo) webu Seznam.cz	16
Obrázek 4: Schéma strategie desktop first.....	19
Obrázek 5: Schéma strategie mobile first.....	20
Obrázek 6: Poměr mezi HW a CSS pixely	21
Obrázek 7: Mřížka (3x3) se základními pojmy	24
Obrázek 8: Schéma mřížky (vlevo) a použití mřížky (vpravo)	25
Obrázek 9: Plovoucí mřížka pro responzivní design webu – vpravo desktop, uprostřed tablet, vlevo telefon.....	26
Obrázek 10: Schéma syntaxe dotazů na médium	27
Obrázek 11: Rozíl mezi vlastnostmi <i>device-width</i> a <i>width</i>	28
Obrázek 12: Ideální rozložení zlomových bodů	30
Obrázek 13: Zlomové body CSS frameworku Bootstrap	31
Obrázek 14: Kriteriaální matice	42
Obrázek 15: Saatyho matice	43
Obrázek 16: Schéma metody AHP	44
Obrázek 17: Responzivní navigační panel frameworku Bootstrap (nahore verze pro desktopy, dole verze pro telefony).....	53
Obrázek 18: Responzivní navigační panel frameworku Foundation (nahore verze pro desktopy, dole 2 verze pro telefony).....	57
Obrázek 19: Špatné zobrazení slideru u komponenty Orbit při změně z režimu portrait do landscape.....	58

Obrázek 20: Špatné zobrazení slideru u komponenty Orbit při změně z režimu landscape do portrait.....	58
Obrázek 21: Zobrazení části vytvořené webové prezentace v režimech portrait i landscape na telefonu s malými rozměry obrazovky (konkrétně iPhone 4).....	72
Obrázek 22: Vhodné zobrazení třídou <i>col-xs-12</i> v režimu portrait (vlevo) a nevhodné v režimu landscape (vpravo) na mobilním telefonu iPhone 7	72
Obrázek 23: Nevhodné zobrazení třídou <i>col-xs-6</i> v režimu portrait (vlevo) a vhodné v režimu landscape (vpravo) na mobilním telefonu iPhone 7	73
Obrázek 24: Editace prvního zlomového bodu.....	74
Obrázek 25: Odlišný layout na mobilních telefonech po změně prvního zlomového bodu v režimech portrait vytvořený třídou <i>col-xs-12</i> a landscape vytvořený třídou <i>col-sm-6</i>	74
Obrázek 26: Design responzivní navigace na desktopových zařízeních	75
Obrázek 27: Špatné zobrazení responzivní navigace na iPhone X v režimu landscape.....	75
Obrázek 28: Editace bodu pro schování navigačního panelu	76
Obrázek 29: Správné zobrazení responzivní navigace na iPhone X	76
Obrázek 30: Špatné zobrazení webové prezentace ve webovém prohlížeči IE 8.....	79

8.2 Tabulky

Tabulka 1: Srovnání základních parametrů jednotlivých řešení tvorby webových stránek	18
Tabulka 2: Typy médií.....	27
Tabulka 3: Vlastnosti médií.....	28
Tabulka 4: Kritéria pro analýzu	50
Tabulka 5: Ohodnocení variant dle jednotlivých kritérií.....	66
Tabulka 6: Ohodnocení variant dle jednotlivých kritérií.....	67
Tabulka 7: Komparace variant podle kritéria 1 (<i>velikost frameworku</i>)	67
Tabulka 8: Komparace variant podle kritéria 2 (<i>implementace</i>)	67

Tabulka 9: Souhrnná tabulka normalizovaných vah jednotlivých variant podle všech kritérií.....	68
Tabulka 10: Výsledné pořadí analýzy metodou AHP	68
Tabulka 11: Ideální a bazální varianty.....	69
Tabulka 12: Standardizovaná kriteriální matice R	69
Tabulka 13: Výsledné pořadí analýzy metodou váženého součtu	70
Tabulka 14: Výsledné pořadí analýzy metodou váženého součtu.....	80

8.3 Grafy

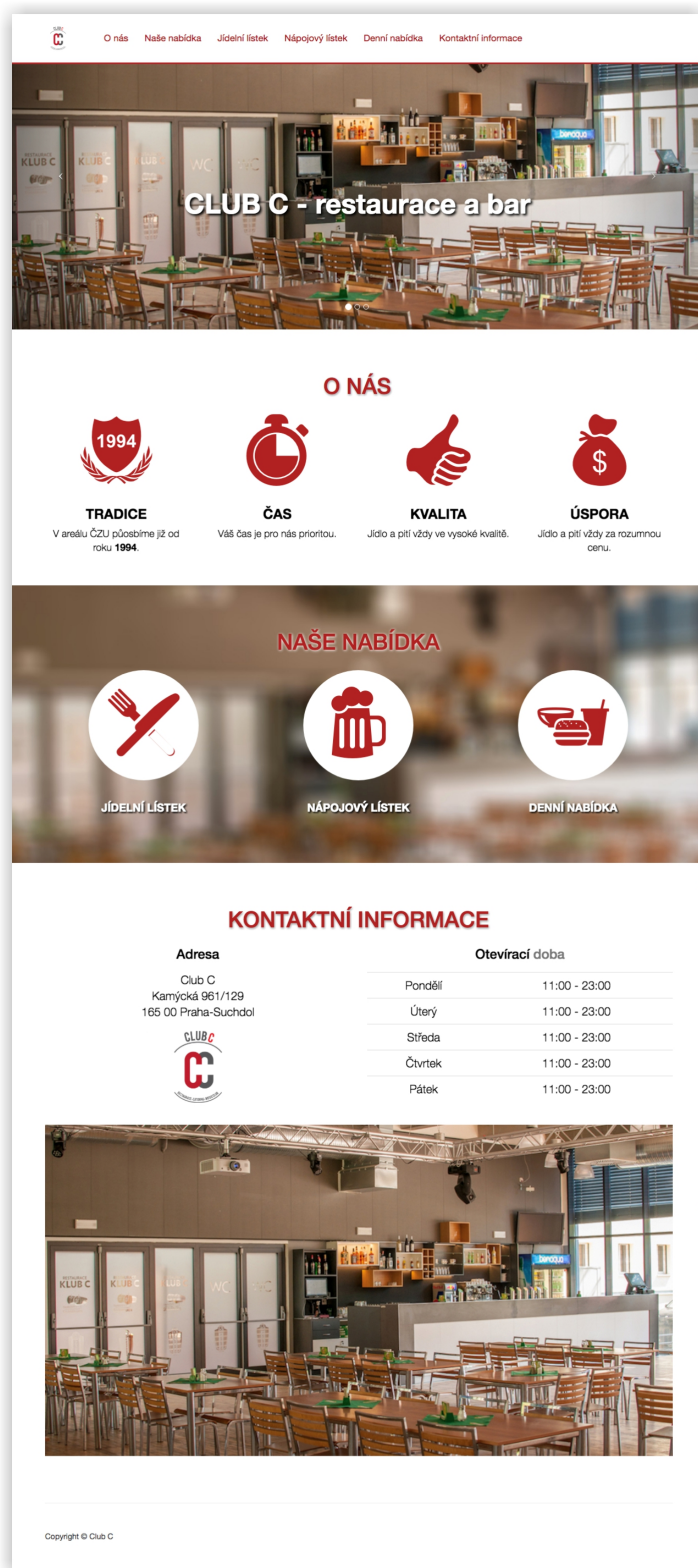
Graf 1: Procentuální poměr zařízení, na kterých jsou zobrazovány webové stránky.....	30
Graf 2: Pět nejvíce využívaných webových prohlížečů v procentech	49

8.4 Přílohy

Příloha 1: Grafický návrh pro tvorbu webové prezentace	93
Příloha 2: Výsledná webová prezentace pro zařízení se šířkou obrazovky 1200px a více (screenshot pořízen na MacBooku se šířkou 1280px).....	94
Příloha 3: Výsledná webová prezentace pro zařízení se šířkou obrazovky od 992px do 1199px (screenshot pořízen na iPadu v režimu landscape se šířkou 1024px).....	95
Příloha 4: Výsledná webová prezentace pro zařízení se šířkou obrazovky od 600px do 991px (screenshoty pořízeny na zařízeních - vlevo iPad v režimu portrait se šířkou 768px, vpravo iPhone 7 v režimu landscape se šířkou 667px).....	96
Příloha 5: Výsledná webová prezentace pro zařízení se šířkou obrazovky do 599px (screenshot pořízen na iPhone 7 v režimu portrait se šířkou 375px).....	97

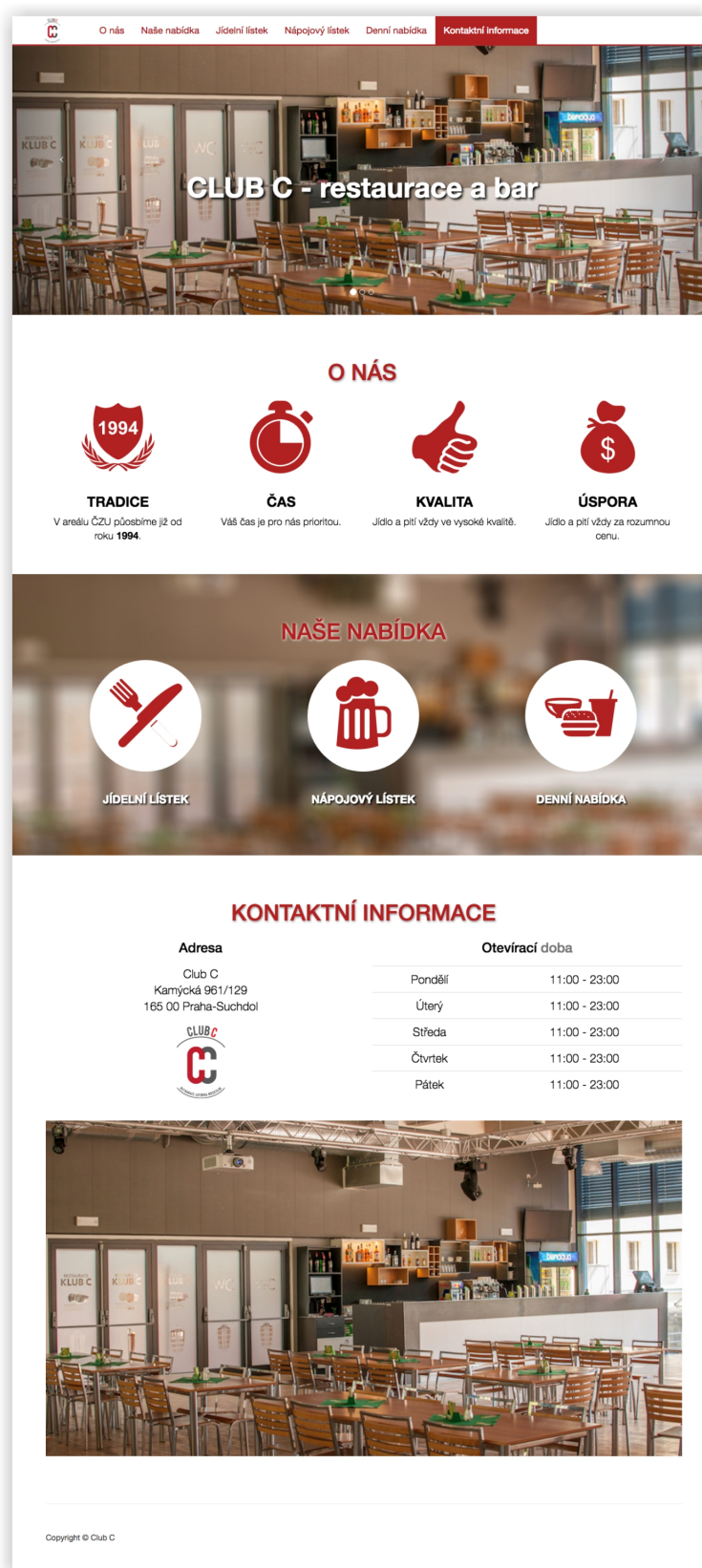
9 Přílohy

Příloha 1: Grafický návrh pro tvorbu webové prezentace



Zdroj: Vlastní zpracování

Příloha 2: Výsledná webová prezentace pro zařízení se šířkou obrazovky 1200px a více (screenshot pořízen na MacBooku se šířkou 1280px)



Zdroj: Vlastní zpracování

Příloha 3: Výsledná webová prezentace pro zařízení se šířkou obrazovky od 992px do 1199px (screenshot pořízen na iPadu v režimu landscape se šířkou 1024px)



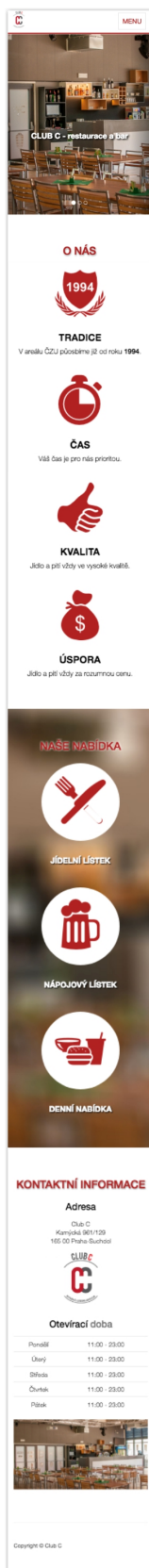
Zdroj: Vlastní zpracování

Příloha 4: Výsledná webová prezentace pro zařízení se šířkou obrazovky od 600px do 991px (screenshoty pořízeny na zařízeních - vlevo iPad v režimu portrait se šířkou 768px, vpravo iPhone 7 v režimu landscape se šířkou 667px)



Zdroj: Vlastní zpracování

Příloha 5: Výsledná webová prezentace pro zařízení se šířkou obrazovky do 599px (screenshot pořízen na iPhone 7 v režimu portrait se šířkou 375px)



Zdroj: Vlastní zpracování