

Univerzita Hradec Králové
Fakulta Informatiky a managementu
Katedra informačních technologií

Vývoj responsivní webové aplikace

Bakalářská práce

Autor: Daniel Kraus

Studijní obor: Informační management

Vedoucí práce: Ing. Pavel Čech, PhD.

Odborný konzultant: Ing. Jakub Plas

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 17. 8. 2015

Daniel Kraus

Poděkování

V první řadě děkuji Ing. Pavlu Čechovi, PhD. za vedení mé bakalářské práce a jeho hodnotné rady. Dále děkuji firmě STRV, s.r.o., pod jejíž záštitou jsem se podílel na vytvoření webové aplikace Ordr.cz a Ing. Jakubu Plasovi, řediteli Ordr.cz, za odborné konzultace při vývoji aplikace.

Anotace

Bakalářská práce se zabývá implementací a chováním responsivního web designu webové aplikace na různých zobrazovacích zařízeních (mobilní telefony, tablety, notebooky, stolní počítače) tak, aby se tato aplikace zobrazila korektně na všech těchto zobrazovacích zařízeních maximálnímu počtu uživatelů. Obsahem práce je zkoumání a vysvětlení optimálních způsobů dosažení tohoto cíle s ohledem a konkrétní implementací na webovou aplikaci Ordr.cz a analýza její využitelnosti.

Klíčová slova

Responsivní web design, webová aplikace, CSS, LESS, HTML, media queries, přizpůsobivý layout, zobrazovací zařízení, mobilní zařízení, Ordr.cz

Annotation

Title: Development of Responsive Web Application

The bachelor thesis deals with the implementation and behavior of a responsive web design of web application on different devices (ie. mobiles, tablets, laptops, PC desktops) with the aim to achieve correct application's display and maximize a number of its users. The content of the research is to explain the optimal ways that help to achieve the goal with regard to the implementation of these methods to the web application Ordr.cz and the analysis of its usage.

Keywords

Responsive web design, web application, CSS, LESS, HTML, media queries, responsive layout, display device, mobile device, Ordr.cz

OBSAH

1	Úvod	1
2	Volba responsivních technik	3
2.1	Mobile-first	4
	Použití na webu Ordr.cz.....	6
2.2	Grid systém	7
2.2.1	Bootstrap	8
2.2.2	Nastavení breakpointů	9
2.2.3	Chování grid systému a jeho nastavení.....	11
3	Frontend implementace.....	15
3.1	Struktura CSS (LESS).....	15
	Kořenový adresář LESS souborů	16
	Konvence pojmenování.....	18
3.2	Navigační menu.....	18
	Dotykové displeje a minimální velikosti prvků	19
	Navigační menu v aplikaci Ordr.cz.....	20
3.3	Práce s obrázky	21
3.3.1	Obrázky na retina displejích.....	22
	Použití v LESS.....	22
	Použití v HTML	23
3.3.2	Ikony jako font.....	25
	Nástroj pro vytvoření setu – IcoMoon.....	25
	Implementace ikony v HML.....	26
3.3.3	Kompresce obrázků	27
3.3.4	Lazy loading.....	28
	Knihovna Unveil.js.....	29
3.4	Možnosti uložení mobilního webu na plochu	30
	Ikona	31
	Nastavení jména aplikace	32
	Uvítací obraz.....	32
	Skrytí ovládacích prvků prohlížeče	33
	Zamezení otevírání odkazů mimo standalone režim	34
	Výsledek po uložení na plochu	34
3.5	Optimalizace touch událostí.....	35

Hover efekt.....	35
Časová prodleva události po dotyku na element	36
4 Analýza využívání.....	38
4.1 Webová platforma.....	39
4.1.1 Zařízení.....	39
Návštěvnost hlavních stránek	40
Množství objednávek podle kategorie zařízení	41
Mobilní zařízení podle výrobců	42
4.1.2 Velikost displeje	43
Rozlišení na desktopových zařízeních	43
Rozlišení na mobilních zařízeních	44
4.1.3 Operační systém.....	45
Mobilní operační systémy z pohledu návštěvnosti	45
Množství objednávek podle mobilního OS	46
4.1.4 Prohlížeč.....	47
Statistika používání prohlížečů celosvětově.....	47
Desktopové prohlížeče při návštěvnosti webu.....	48
Mobilní prohlížeče při návštěvnosti webu	48
4.2 Platformy iOS a Android	49
Počet stažení mobilních aplikací	49
Návštěvnost mobilních aplikací	50
Počet objednávek z mobilních aplikací	50
4.3 Komparace všech platforem	51
Návštěvnost – mobilní web vs. mobilní aplikace	51
Počet objednávek – web vs. mobilní aplikace	51
Počet objednávek – mobilní web vs. mobilní aplikace.....	52
5 Shrnutí výsledků.....	53
6 Závěr	58
7 Seznam použité literatury	60
8 Přílohy.....	62

SEZNAM OBRÁZKŮ

Obrázek 1: Vzhled domovské stránky a hlavní nabídky jídel.....	7
Obrázek 2: Nejčastější velikosti displejů dvaceti nepoužívanějších zařízení [7 str. 4]	10
Obrázek 3: Patička na průhledech větších než 767px.....	13
Obrázek 4: Mobilní patička	14
Obrázek 5: Způsoby hlavní navigace ve webové aplikaci Ordr.cz	20
Obrázek 6: Rozbalená mobilní navigace	21
Obrázek 7: Set ikon z Icomoon použitý v aplikaci.....	26
Obrázek 8: Hover efekt na ikoně položky v patičce.....	27
Obrázek 9: Web Ordr.cz po uložení na plochu iPhone 6	34
Obrázek 10: Porovnání staré a nové mobilní verze webu Ordr.cz	53
Obrázek 11: Porovnání navigací starého a nového mobilního webu	54
Obrázek 12: Porovnání mobilního webu, nativní aplikace pro iOS a pro Android	56

SEZNAM ZDROJOVÝCH KÓDŮ

Zdrojový kód 1: Import souborů Grid systému od Bootstrapu	9
Zdrojový kód 2: Proměnné deklarující breakpointy v aplikaci.....	11
Zdrojový kód 3: Proměnné pro práci s media queries	11
Zdrojový kód 4: Třídy hlavních kontejnerů.....	12
Zdrojový kód 5: Ukázka grid systému na patičce webové aplikace	13
Zdrojový kód 6: LESS mixin pro vložení obrázku s automatickou podporou retina rozlišení	22
Zdrojový kód 7: Použití atributu srcset na elementu img.....	24
Zdrojový kód 8: HTML kód s ikonou v položce patičky webu.....	27
Zdrojový kód 9: Použití lazy loadu obrázku v HTML.....	29
Zdrojový kód 10: jQuery inicializace lazy loadu na obrázek.....	30
Zdrojový kód 11: LESS deklarace umožňující fadeIn efekt při lazy loadu.....	30
Zdrojový kód 12: HTML deklarace uvítacího obrazu pro uložený mobilní web.....	33
Zdrojový kód 13: Inicializace FastClicku a jeho opravení pro Google Autocomplete ..	37

SEZNAM TABULEK

Tabulka 1: Porovnání kompresí obrázků pomocí nástrojů VS 2013 a Kraken.io	28
Tabulka 2: Chronologie expanze Ordr.cz do dalších oblastí	38

SEZNAM GRAFŮ

Graf 1: Návštěvnost webu dle kategorie zařízení.....	40
Graf 2: Návštěvnost konkrétních stránek webu dle kategorie zařízení	40
Graf 3: Počty objednávek z webu dle kategorií zařízení.....	41
Graf 4: Návštěvnost webu dle výrobců mobilních zařízení	42
Graf 5: Návštěvnost webu dle rozlišení desktopů.....	43
Graf 6: Návštěvnost webu dle mobilních rozlišení	44
Graf 7: Návštěvnost webu dle OS pro PC	45
Graf 8: Návštěvnost webu dle mobilních OS	45
Graf 9: Množství objednávek z webu podle mobilních OS.....	46
Graf 10: Návštěvnost webu dle desktopových prohlížečů.....	48
Graf 11: Návštěvnost webu dle mobilních prohlížečů.....	49
Graf 12: Počty stažení mobilních aplikací	50
Graf 13: Počty návštěv mobilních aplikací dle platforem iOS a Android.....	50
Graf 14: Počty objednávek z mobilních aplikací dle platforem iOS a Android	50
Graf 15: Srovnání návštěvnosti mobilního webu a mobilních aplikací	51
Graf 16: Srovnání počtu objednávek z webu a mobilních aplikací.....	52
Graf 17: Srovnání počtu objednávek z mobilního webu a mobilních aplikací.....	52

1 Úvod

Chytré telefony, tablety, čtečky, notebooky, stolní počítače, chytré televize a v poslední době často zmiňované tzv. wearable devices. Všechna zmíněná zařízení jsou dnes připojitelná k internetu a mohou zobrazovat veškerý dostupný internetový obsah od statických webových stránek po složité aplikace postavené na webových technologiích. Důvod, proč se zabývat responsivním designem webových stránek a webových aplikací, je dosažení optimálního zobrazení a zajištění uživatelsky maximálně komfortní interakce s těmito stránkami a aplikacemi na různých typech zařízení s ohledem na šířku, výšku, rozlišení a další parametry jejich zobrazovací plochy.

Trend responsivního web designu v posledních letech stále roste a není již ničím výjimečným. Stává se spíše nepsaným pravidlem. S tím přichází otázka „Může webová aplikace běžící v internetovém prohlížeči nebo na jeho jádru plně nahradit nativní iOS, Android nebo Windows Phone aplikaci?“. Položení zmíněné otázky je velice důležité již před začátkem samotného vývoje softwarových produktů začínajících firem, protože odpověď na ni může značně snížit náklady na celý vývoj. Ovšem její odpověď není tak jednoznačná. Bakalářská práce si proto klade mj. za cíl zodpovědět tento problém a doložit ho ostrými porovnávajícími daty získanými během 1. roku fungování provozu aplikace Ordr.cz od 1. července 2014 do 31. června 2015 na platformě webové a platformách iOS a Android.

Služba Ordr.cz ve svých aplikacích nabízí rozvoz jídla s garancí jeho dodání do 10 minut. V současné chvíli působí v širokém centru Prahy ve dvou směnech – oběd a večeře. Oběd je rozvážen v čase 11:15—13:30 a večeře 18:00—20:30. Do poloviny jejího prvního roku ostrého provozu nabídka v responsivním webu a v nativních aplikacích pro systémy iOS a Android obsahovala vždy pouze právě jedno jídlo k obědu. Další jedno jídlo určené k večernímu rozvozu bylo přidáno do druhé půle.

Hlavním cílem bakalářské práce je uplatnění technik využitelných pro vytvoření responsivní webové aplikace a jejich současná frontend implementace do nové verze webové aplikace Ordr.cz. Na ní jsou výsledky uplatnění pozorovány a dokládány za účelem dosažení co nejpodobnějšího chování webu nativním aplikacím a korektního

responsivního zobrazení na většině zařízení dle jejich parametrů. Práce se na svém začátku zabývá metodou prvotního grafického i obsahového návrhu responsivní aplikace, jejími výhodami a nastavením přizpůsobivého layoutu webu velikosti zobrazovacích ploch zařízení. Dále podrobněji rozebírá techniky při práci se soubory obsahující definice stylů, s obrázky a jejich optimalizacemi, nebo možnostmi, jak responsivní webové aplikace co nejvíce přiblížit vzhledu a chování nativních mobilních aplikací. Druhá část práce se zaměřuje na analýzu využívanosti první verze responsivní webové aplikace ve zmíněném časovém horizontu a pro zodpovězení otázek je provedena komparace s aplikacemi pro iOS a Android.

Bakalářská práce, jak již bylo zmíněno, ve svých praktických částech implementuje zkoumané techniky do nové verze webu, na které se začalo pracovat na jaře 2015 a její plánované vypuštění je v září 2015. Nová verze s kompletním redesignem přináší řadu novinek, jako je např. jednodušší objednávkový proces, nabídka ze dvou produktů, možnost pro zákazníka získávat SMS upozornění ve chvíli rozvozu jeho oblíbeného jídla, další stránky s informacemi o službě atp. Velmi podobný grafický design mobilní verze webu s jeho, rovněž aktualizovanými, sesterskými nativními aplikacemi je dalším cílem redesignu i bakalářské práce.

Jedním z hlavních zdrojů při zpracování bakalářské práce je americký webový designer a vývojář Ethan Marcotte a jeho kniha *Responsive Web Design*, ve které shrnuje a detailně popisuje používané praktiky při tvorbě responsivních webů s využitím HTML a CSS. Rovněž je to kniha, která položila základy responsivních webových technik. Důležitým autorem je Jonathan Fielding s knihou *Beginning Responsive Web Design with HTML5 and CSS3*, která je recenzovaná i nakladatelstvím Springer. Dalším zdrojem jsou autoři Craig Sharkie a Andrew Fisher a jejich publikace *Jump Start Responsive Web Design*, která se zaměřuje praktickým úvodem do vývoje modernějších webů. Velmi důležitým autorem je Scott Jehl a jeho kniha *Responsible Responsive Design*, která je jednou z posledních publikací na dané téma a tak skýtá mnoho nových trendů v oboru. Posledním klíčovým autorem je Luke Wroblewski a jeho kniha *Mobile first*, ve které představuje moderní metodu a přístup k vyvíjení nových webových stránek či aplikací. Při zpracování bakalářské práce je dále vycházeno z vlastních zkušeností jejího autora nabytých z pozice webového vývojáře a programátora ve firmě STRV, s.r.o.

2 Volba responsivních technik

Responsivní web design je termín, který poprvé použil Ethan Marcotte ve svém stejnojmenném článku v roce 2010. Marcotte responsivním designem nazývá takový návrh webové stránky a jejího stylování, které bude optimalizované pro všechny druhy nejrůznějších zařízení: mobilní telefony, netbooky, notebooky, tablety, osobní počítače i nadměrné displeje. [1]

K přizpůsobení webové stránky na její frontendové vrstvě jsou nutné podle Marcotta [1] tři základní ingredience:

- Proporcionální (flexibilní) layout – rozložení jednotlivých boxů a sekcí webové stránky je definováno relativně, flexibilně vůči svému okolí
- Proporcionální (flexibilní) obrázky a média
- Media queries – vzhled je přizpůsoben na základě parametrů cílového zařízení, typicky velikosti průhledu (ang. viewport), tj. dostupné plochy pro vykreslení stránky, nikoliv velikost displeje daného zařízení

Jedním z mnoha důvodů, proč responsivní techniky rozebírané v této kapitole vůbec do webů implementovat, není jen jejich samotné následné chování na mobilních zařízeních, ale rovněž přináší řadu faktorů ovlivňujících počty přístupů a návštěv stránek, které mohou značně ovlivnit obchodní hlediska služeb jako je právě Ordr.cz. Autoři Makino a Phan [2] ve svém článku *Rolling out the mobile-friendly update* na oficiálním blogu společnosti Google vydali 21. dubna 2015 oznámení o spuštění nové funkce při vyhledávání webových stránek ve svém vyhledávači, který přidá každé stránce štítek „mobile-friendly“. Štítek tedy říká, jestli je daný web přizpůsobený mobilním zařízením nebo ne a pokud je, indexovací robot stránku upřednostní v pořadí. Funkce přizpůsobivost rozlišuje pouze v mobilních vyhledávacích globálně napříč všemi jazyky a zohledňuje konkrétní stránky, nikoliv weby jako celky. Tato nová možnost rozlišovat responsivní weby je tedy jen dalším důkazem, proč je důležité se takovou vlastností při vývoji stránek a aplikací zabývat.

Vzhledem k rychlému nárůstu počtu responsivních webů se v posledních letech na open source trhu vyskytuje veliká škála frontendových frameworků, metod a technik, které urychlují, usnadňují a mnohdy ulehčují práci programátorům při vyvíjení

moderních webů. Obsahem této kapitoly je výběr takových základních technik a metod, které webové aplikaci Ordr.cz umožňují responsivní chování.

2.1 Mobile-first

Úplně na samotném začátku vývoje stojí UI a UX¹ návrh dané aplikace. Ještě před nástupem mobilních zařízení grafičtí designéři vždy začínali návrhem pro klasické desktopové a laptopové verze se zobrazovacím průhledem začínajícím na cca. 1080px. Na takových zobrazovacích plochách měli veliký prostor pro naplnění účelů webové stránky, ať už to byl web prezentační, veliké lákavé obrázky, spousta informačních textů, tabulky plné zajímavých dat atp., nebo aplikační - např. e-shop, kde bylo mnoho místa na umístění nabízených produktů. Velikosti prostorů se však v posledních letech dramaticky změnilo, resp. zmenšily a proto se, v případě responsivních a mobilních webů, začíná ustávat tzv. mobile-first přístup k designování stránek a aplikací.

S narůstajícím počtem mobilních zařízení narůstá fragmentace velikosti displeje, na kterém uživatelé zobrazují webové stránky. Malý displej představuje omezený prostor pro prezentaci dat, je tak třeba s dostupným prostorem nakládat odpovědně a co nejefektivněji a proto grafičtí designéři začínají aplikace navrhovat od mobilních verzí. Luke Wroblewski ve své stejnojmenné knize *Mobile First* provedl krátkou analýzu, ve které říká, že první vlna smartphone zařízení se systémem iOS, Android a WebOS měla většinou rozlišení 320x480px, což představovalo asi 80 % méně prostoru, který zde chyběl, oproti těm desktopovým velikostem. Takže až 80 % odkazů, reklam, textů, obrázků a dalších prvků z grafických designů pro veliká rozlišení muselo najít jinou interpretaci, nové místo nebo úplně zmizet, protože zkrátka pro ně nezbyl prostor. [3]

Dále v zamyšlení pokračoval a dospěl k závěru, že když se vezme v úvahu, kolik zbytečných a umělých „vycpávek“, které dříve typický webový design obsahoval, tak si každý uvědomí, proč je taková metoda návrhu výhodná jak pro podniky, tak zákazníky. [3 str. 19] S tím se rovněž ztotožňuje ve své knize Fielding, kde říká, že mobile-first přístup uživatelům nabídne právě ty funkce, které opravdu potřebují a na kterých záleží. [4]

¹ UI a UX – User Interface a User Experience

Na otázku „Proč tedy začínat mobilní verzí?“ odpovídá Luke Wroblewski několika výhodami. První výhodou je, že danému produktu nebo službě při definování jejich vlastností a funkcí takový přístup umožňuje reagovat na rychle přibývajícím nové příležitosti, které dnes mobilní přístroje na trhu svými parametry otevírají. Jednou z takových příležitostí může být například geolokace, kdy lze prostřednictvím mobilních prohlížečů a vestavěných GPS modulů v mobilních zařízeních zjistit polohu uživatele, čímž se dá cílit na konkrétnější skupiny zákazníků v určitých územích. Dále pak vyzdvihuje fakt nastíněný v předchozích odstavcích - takový přístup nutí identifikovat více klíčové vlastnosti vytvářeného produktu, které musejí být prezentovány či nabízeny ve zmenšeném mobilním prostoru zobrazovací plochy tak, aby byl jejich význam a pochopení stále zřejmý a lehce dostupný. [3]

Při návrhu mobilního webu je nutné se zaměřit na obsah, který uživatel nutně potřebuje k práci s aplikací nebo pochopení, co aplikace vlastně dělá, protože není více místa, kam taková data umístit. Dnešní mobilní zobrazovací plochy nejčastěji začínají šíří 320px a končí velikostmi nad 768px (ve vertikální poloze). Je velmi důležité také myslet na vyhnutí se vkládání zbytečně objemných dat, kterými mohou být např.:

- Videá hostovaná na vlastním serveru podléhající přímému stahování
- Vysoký počet obrázků nebo umělých grafických prvků tam, kde je lze nahradit CSS3 definicemi jako jsou např. gradient, border-radius, box-shadow, text-shadow aj.
- Vysoce objemné JavaScriptové knihovny, zvláště jsou-li použity pouze pro jednu nebo dvě funkce

Nejméně tyto 3 body značně zpomalují načítání a výkon webové mobilní aplikace pracující online. Proto je při mobilním návrhu vždy lepší přistupovat pesimisticky k rychlosti internetového připojení, neboť velmi často je mobilní web otevírán na cestách a mimo budovy, kde není dostupná Wi-Fi síť a na řadu přichází mobilní připojení. Pokud je načítací rychlost aplikace v pořádku na mobilním zařízení, dá se poté předpokládat, že bude i velmi rychlá na zařízení desktopovém. [3] Touto problematikou se zabývá ve své knize i Fielding, kde se odvolává na server *Web Performance Today*². Říká, že zatímco datové velikosti webových stránek rok od roku

² <http://www.webperformancetoday.com>

narůstají, očekávaný čas podle uživatelů pro načtení stránky klesá a na základě článku ze zmíněného serveru dokládá, že existuje vztah mezi výkonem webu a konverzí uživatele k nákupu produktů v něm nabízených. Uvádí příklad z roku 2014, kdy americký obchodní řetězec Walmart zjistil, že při každém zlepšení rychlosti načtení o 1s se konverze zvýší o 2 % nebo v případě produktové stránky Mozilla Firefox zvýšení rychlosti o 2,2s znamenalo nárůst počtu stažení o 15,4 %, což představovalo navýšení až o 10 milionů jednotek stažení. [4 str. 264]

Nevýhodou mobile-first přístupu při implementaci z technického hlediska je chybějící kompatibilita media query definice (@media) s prohlížeči Internet Explorer verze 8 a nižší. Definici tyto prohlížeče nerozeznají a poté se i na větším rozlišení stále zobrazuje zúžená mobilní velikost. Problémem se překvapivě Luke Wroblewski ve své knize nezabývá, ale řešení představuje Fielding v podobě jednoduché a datově velmi lehké JavaScriptové knihovny Respond.js³, mj. vytvořené autorem Scottem Jehlem. [4 str. 72]

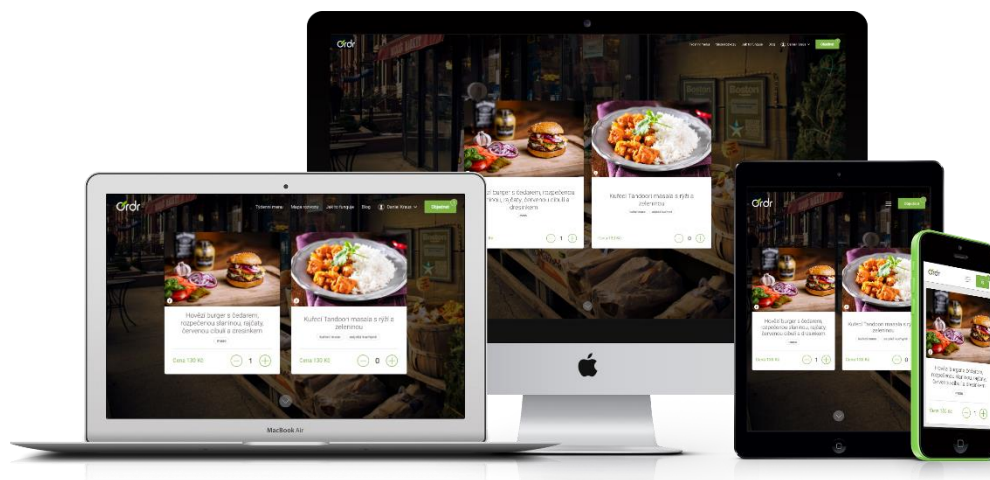
Použití na webu Ordr.cz

Použití mobile-first přístupu na webovou aplikaci Ordr.cz je zřejmé - klíčová je soustředěnost na nabídku produktů. Proto je nutné, aby uživatel měl co nejrychlejší, nejkratší a nejpohodlnější přístup k:

- Denní nabídce – 2 jídla (každé nabízené jídlo je doplněno fotografií, názvem, cenou a tlačítky pro regulaci množství objednávaných jídel v košíku)
- Přechodu do košíku – viditelné tlačítko v navigačním pásu u horního okraje viewportu (na mobilní verzi s ikonou košíku a na desktopové verzi s nápisem „Objednat“), tlačítko má dva následující stavy:
 1. šedivé tlačítko (nefunkční) – šedivé pouze na domovské stránce, kde lze regulovat množství v košíku a rovná-li se množství jídel v košíku 0, není tedy možný přechod do prázdného košíku
 2. zelené tlačítko (přesměrovávající do košíku) – na všech podstránkách pro snadný přechod přímo do košíku i za předpokladu, že košík je prázdný; nebo na domovské stránce je-li množství v košíku větší než 1

³ <https://github.com/scottjehl/Respond>

- Přehlednému navigačnímu menu na další podstránky webu (Pozn. autora: Navigační menu je podrobněji rozebráno v další kapitole.)



Obrázek 1: Vzhled domovské stránky a hlavní nabídky jídel

2.2 Grid systém

Pro vytváření responsivního layoutu je při velikosti tohoto projektu nutné myslet na jeho škálovatelnost v narůstání počtu dalších stránek a funkcionalit, které jsou v průběhu vývoje zasazovány do jednotného rozvržení mřížky napříč celým webem. Taková mřížka umožňující přizpůsobivé chování layoutu se nazývá grid systém. Není nutné vyvíjet vlastní systém od samého začátku, protože ho již obsahuje řada frontendových frameworků. Sharkie a Fisher [5] do nich zařazují a poměrně detailně porovnávají hojně používaný Bootstrap⁴, Skeleton⁵, Gridset⁶, 960 Grid System⁷, Semantic Grid System⁸ a 320 and up⁹ a Firdaus [6] k nim přidává ještě Foundation¹⁰. Použitím kteréhokoliv z nich je značně urychlen a usnadněn vývoj každého responsivní webu a rovněž vylepšují práci celého vývojového týmu, protože jsou ve většině z nich používány obecně známé konvence pojmenovávání tříd. Všechny z výčtu jsou postaveny na CSS pre-procesorech LESS či SASS a některé obsahují komplexní řešení v podobě dalších přídatných komponent (např. Bootstrap či Foundation). Stejně jako v případě autorů Sharkieho a Fishera nebo i Jonathana

⁴ <http://getbootstrap.com>

⁵ <http://www.getskeleton.com/>

⁶ <http://gridsetapp.com/>

⁷ <http://960.gs/>

⁸ <http://semantic.gs>

⁹ <http://stuffandnonsense.co.uk/projects/320andup/>

¹⁰ <http://foundation.zurb.com>

Fieldinga [4] je při vývoji webové aplikace Ordr.cz zvolen Bootstrap, který je postavený na pre-processoru LESS (ale i SASS), jehož právě aplikace využívá při vývoji stylů.

2.2.1 Bootstrap

Knihovna Bootstrap je známá díky tomu, že její vývoj původně pochází z roku 2010 od vývojářského týmu z Twitteru, kde byl používán pro interní účely a později byl uvolněn pro veřejnou vývojářskou komunitu na GitHubu¹¹. Responsivní chování implementoval až od své 2. verze a v současné verze nese pořadové označení 3.3.2. Na svých stránkách se prezentuje jako nejpopulárnější HTML, CSS a JS framework umožňující rychlý a intuitivní vývoj mobile-first aplikací na web. Obsahuje četné množství komponent, jako jsou navigační menu, elementy formulářů, tlačítka, různé panely a boxy, sety ikon aj. a dále i jQuery¹² rozšíření v podobě carouselů, modálních oken, záložek atp. [6]

Sharkie a Fisher sepsali výhody a nevýhody, které shledávají na této frontendové knihovně: [5 str. 44]

- Výhody
 - Popularita zajišťuje silnou vývojářskou komunitu, ve které jsou vývojáři s frameworkem spokojeni
 - Framework je díky svému stáří a nasazení na mnoho známých aplikacích a stránkách (např. www.twitter.com) intenzivně testován
 - Je přizpůsobitelný potřebám vývojáře, lze si tedy vygenerovat pouze ty funkce a komponenty, které vývojář chce a potřebuje pro svůj projekt
- Nevýhody:
 - Rozšiřuje navíc značkovací jazyk v porovnání s řešením na míru
 - Rozšiřuje velikost CSS souboru rovněž v porovnání s řešením na míru
 - Nepoužívá sémantické názvy tříd

S posledním bodem nevýhod lze dnes nesouhlasit, neboť od roku 2013, kdy autoři analyzovali tyto výhody a nevýhody, se sémantika tříd výrazně vylepšila právě díky otevřenosti celé knihovny široké veřejnosti.

¹¹ <http://github.com>

¹² <http://jquery.com>

Ve webu Ordr.cz není zahrnuta celá knihovna Bootstrap. Grafický design aplikace skýtá mnoho specifických prvků a bylo by pracné styly z knihovny upravovat. Avšak používá některé pojmenovávání tříd (např. prvky formulářů – form-group, input-icon, form-control, btn atp.), které je velmi jednoduché a umožňuje lepší představivost výstupu při kódování HTML. Ve vývoji aplikace je použit proto pouze grid systém této knihovny.

Grid systém od Bootstrapu obsahuje předdefinované třídy umožňující rychlé vytvoření řádků a sloupců v layoutu s responsivním chováním. Pro vygenerování tříd grid systému je nutné v projektu importovat soubory dostupné z Bootstrap knihovny, jak ukazuje Zdrojový kód 1.

```
// Import Layout Mixins used to generate the grid classes
@import "GridSystem/_grid-framework.less";
@import "GridSystem/_grid.less";
@import "GridSystem/_responsive-visibility.less";

// Add responsive utility classes
@import "GridSystem/_responsive-utilities.less";
```

Zdrojový kód 1: Import souborů Grid systému od Bootstrapu

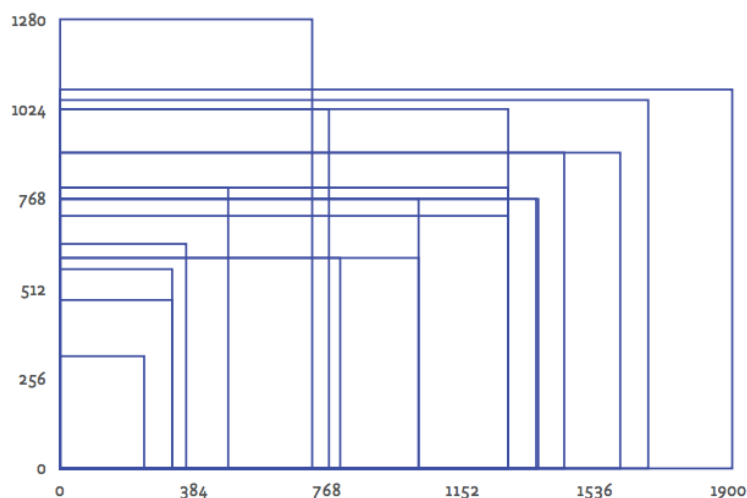
2.2.2 Nastavení breakpointů

Ještě předtím, než bude více přiblíženo chování grid systému, je nutné identifikovat horizontální body zlomu – breakpointy – umožňující zalomení nebo změnu layoutu s ohledem na šířku viewportu. K jejich identifikaci je vhodné znát cílové skupiny uživatelů, resp. jejich zařízení.

Scott Jehl ve své knize [7 str. 17] cituje Stephena Haya „*Začněte malými velikostmi displejů, poté stále rozšiřujte, dokud to nevypadá hrozně. JE ČAS PRO BREAKPOINT!*“¹³ (překlad autora). Obecně je takový přístup poměrně přirozený, vypadá-li stránka esteticky špatně, fonty jsou již malé a těžko čitelné, kolem kontejnerů je spousta místa atp., je na čase tyto a jiné elementy opravit tak, aby vypadaly opět hezky. Postupem času se body ustávají na stejných, nebo alespoň přibližných osvědčených velikostech. Jedním z bodu, na kterém se shoduje většina autorů je 768px, což je velikost displeje

¹³ „Start with the small screen first, then expand until it looks like shit. TIME FOR A BREAKPOINT!“ [7 str. 17]

iPadů ve vertikální poloze. Nejnižším bodem je podle Marcotta [1] 320px, což je šíře většiny menších smartphonů jako iPhone 4, 5 a 5s nebo Samsung Galaxy S3, S4 atp.



Obrázek 2: Nejčastější velikosti displejů dvaceti nejpoužívanějších zařízení [7 str. 4]

Breakpointy ve webové aplikaci Ordr.cz byly zvoleny na základě analýzy využitosti (viz kapitola 4). Názvy proměnných (např. `@screen-xs`, `@screen-xs-min`, `@screen-phone` a další.) jsou striktně dané grid systémem Bootstrapu, proto je nutné je dodržet jak uvádí Zdrojový kód 2, aby systém dokázal správně generovat CSS s ohledem na deklarované body. Důsledkem mobile-first přístup lze samozřejmě deklarovat styly i pro displeje menší než 480px. Jsou to začínající styly, které podléhají až dalším zlomovým bodům. Nadcházející velikosti layoutu, označené zkratkou jako `small (sm)`, se přizpůsobí mezi velikostmi 768px až 969px. Od šíře 970px začínají velikosti nazvané jako `medium (md)`, jsou to zejména tablety v horizontální (landscape) poloze a notebooky s menším displejem okolo 13". Pro větší (`lg`), např. stolní počítače, začínají změny layoutu od 1250px a pro extra veliké (`xlg`) zobrazovací plochy, jako jsou např. velké 27" počítače iMac, začínají změny od šíře 1920px.

Jak si lze všimnout ve Zdrojový kód 2, hlavními jednotkami proměnných nejsou pixely, ale `em` jednotky. Použitím `em`¹⁴ jednotek při deklaraci breakpointů se zabývá Scott Jehl [7 str. 57] a usuzuje, že jejich použití je vhodnější, neboť vytváříme-li flexibilní layout, je nasnadě použít i flexibilní jednotky. Hodnoty breakpointů byly

¹⁴ 1em jednotka se rovná 16 pixelům.

stanoveny na základě analýzy provedené ve 4. kapitole v části věnující se počtu návštěvností webu Ordr.cz podle rozlišení desktopových a mobilních displejů.

```
// Extra small screen / phone
@screen-xs: 30em; // 480px
@screen-xs-min: @screen-xs;
@screen-xs-max: 47.938em; // 767px
@screen-phone: @screen-xs-min;
// Small screen / tablet
@screen-sm: 48em; //768px
@screen-sm-min: @screen-sm;
@screen-sm-max: 60.5625em; // 969px
@screen-tablet: @screen-sm-min;
// Medium screen / desktop
@screen-md: 60.625em; //970px
@screen-md-min: @screen-md;
@screen-md-max: 78.0625em; // 1249px
@screen-desktop: @screen-md-min;
// Large screen / lg desktop
@screen-lg: 78.125em; //1250px
@screen-lg-min: @screen-lg;
@screen-lg-desktop: @screen-lg-min;
// Extra Large screen
@screen-xlg: 120em; //1920px
```

Zdrojový kód 2: Proměnné deklarující breakpointy v aplikaci

Pro zjednodušení a urychlení psaní media queries definic jsou vytvořené proměnné ve Zdrojový kód 3, s kterými je zacházeno i v další části práce. Ukázka zdrojového kódu obsahuje proměnné pouze pro velikost xs. Pro pokračující velikosti sm, md, lg a xlg se opakuje stejný princip.

```
// Variables used for @media definitions
@xs-min: ~"screen and (min-width: @{screen-xs})";
@xs-only: ~"@{xs-min} and (max-width: @{screen-xs-max})";
@xs-max: ~"screen and (max-width: @{screen-xs-max})";
```

Zdrojový kód 3: Proměnné pro práci s media queries

2.2.3 Chování grid systému a jeho nastavení

Jak již bylo zmíněno v předcházející části, grid systém vygeneruje třídy pro nutné elementy layoutu. Není žádoucí zde vypisovat všechny třídy, protože k tomu slouží oficiální dokumentace knihovny, ale bude zde přiblížen princip chování systému.

Základním elementem layoutu je třída `.container`, která ještě není definovaná systémem, ale musí nést toto pojmenování. Kontejneru lze nastavit tedy vlastní šířku.

Ve webové aplikaci jsou 4 základní velikosti kontejnerů, viz Zdrojový kód 4. Základním a nejčastěji z nich používaným, je kontejner nesoucí třídu `.container`, který má až do velikosti 768px maximální šířku 590px. To znamená, že na displejích menších než 590px bude mít šířku rovnou 100%. Na větších zobrazovacích plochách má maximální šířku 920px. Mixin¹⁵ označený jako `.container()` pouze samotný kontejner centruje (přidává mu tedy `margin-left:auto; margin-right:auto`) a rovněž přidává `padding` po jeho levé i pravé straně.

Dalším nutným elementem pro používání grid systému je třída nazvaná jako `row`, tedy řádek. Každý řádek musí být vložen přímo v elementu nesoucí třídu `.container`. Do řádku lze pak již vkládat sloupce (kolonky), těch je maximálně 12, mají vždy flexibilní šíři v %, relativní pozici a plují doleva (`float:left`). Číselné označení sloupce 1-12 udává jeho šířku - 12 = 100%, 6 = 50%, 1 = 1/12%. Vygenerované třídy mají označení:

- `col-xs-{1-12}` - sloupec bude mít defaultně nastavenou šířku pro viewport menší nebo rovno než je proměnná `@screen-xs-max` (tedy 767px)
- `col-sm-{1-12}` - šířka sloupce bude aktivní pro pruhled od větší než `@screen-sm-min`
- `col-md-{1-12}` - šířka sloupce se projeví od `@screen-md-min`
- `col-lg-{1-12}` - šířka sloupce bude aplikována od `@screen-lg-min`

System dále vygeneruje třídy umožňující posouvání kolonek v řádku pomocí pojmenování `col-{xs;sm;md;lg}-{offset;pull;push}-{1-12}` a třídy pro

```
.container{
  max-width:36.875em; //590px
  .container();

  @media @sm-min{
    max-width:50em; //800px
  }
  @media @md-min{
    max-width:57.5em; // 920px
  }
}
.container-md {
  max-width:36.875em; //590px
  .container();

  @media @sm-min{
    max-width:50em; // 800px;
  }
  @media @md-min{
    max-width:51.25em;// 820px
  }
}
.container-sm{
  max-width:36.5625em;// 585px
  .container();
}
.container-xs{
  max-width:29.375em; //470px;
  .container();
}
```

Zdrojový kód 4: Třídy hlavních kontejnerů

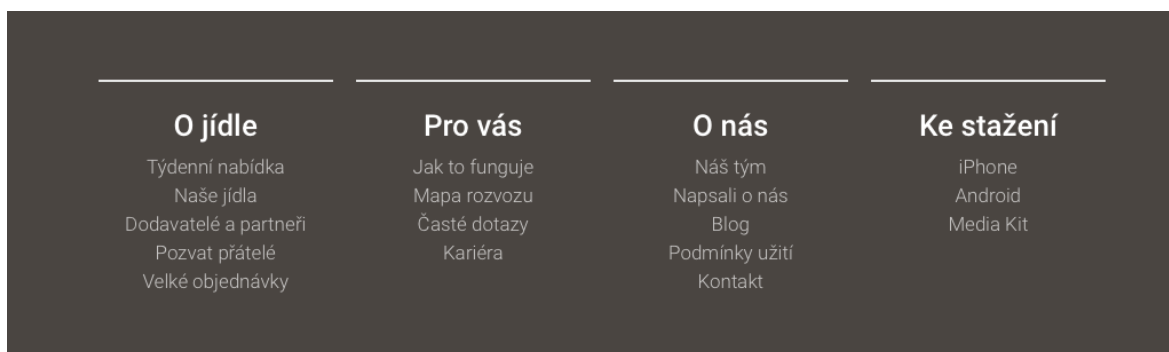
¹⁵ Mixin je obdobou funkce v programovacích jazycích. Výrazně šetří čas při psaní stylů, když seskupuje CSS definice nebo proměnné, které se po kompilaci vloží dané třídě, v níž je definovaný. [4 str. 180]

viditelnost `visible-{xs;sm;md;lg}-block` aj. Implementaci systému v HTML ukazuje Zdrojový kód 5 na příkladu patičky webu.

```
<div class="container">
  <div class="row clearfix">
    <div class="col-xs-6 col-sm-3">
      <h3 class="title-gamma">O jídle</h3>
      <!-- Navigation list placed here-->
    </div>
  </div>
  <div class="col-xs-6 col-sm-3">
    <div class="col">
      <h3 class="title-gamma">Pro vás</h3>
      <!-- Navigation list placed here-->
    </div>
  </div>
  <div class="clearfix visible-xs-block"></div>
  <div class="col-xs-6 col-sm-3">
    <div class="col">
      <h3 class="title-gamma">O nás</h3>
      <!-- Navigation list placed here-->
    </div>
  </div>
  <div class="col-xs-6 col-sm-3">
    <div class="col">
      <h3 class="title-gamma">Ke stažení</h3>
      <!-- Navigation list placed here-->
    </div>
  </div>
</div>
</div>
```

Zdrojový kód 5: Ukázka grid systému na patičce webové aplikace

Zdrojový kód 5 tedy říká, že na zobrazovací ploše větší než 767px bude patička mít 4 sloupce v jednom řádku. Jak se patička vykresluje v prohlížeči s šíří tohoto průhledu, ukazuje Obrázek 3. (Pozn. autora: Ukázka kódu neobsahuje z důvodu jeho zkrácení `` seznamy s navigačními odkazy na další podstránky, které jsou na obrázku již viditelné.)



Obrázek 3: Patička na průhledech větších než 767px

Na průhledech, menších a včetně zlomového bodu uvedeného výše, bude rozvržení mít 2 sloupce a 2 řádky. Původně 4 sloupce rozdělí prázdný `div` s třídou `.clearfix` a `.visible-xs-block`, kterému se nastaví definice `display:block`; pouze při této šířce průhledu. Vykreslení mobilní patičky představuje Obrázek 4.



Obrázek 4: Mobilní patička

3 Frontend implementace

V předešlé kapitole byly vyjmenovány nezbytné hlavní techniky a přístupy zajišťující responsivní chování webových aplikací. Nyní se v této části bude práce zabývat dalšími metodami a postupy, které přizpůsobivé chování rozšiřují a ještě hlouběji vylepšují, a které se při vývoji jejím programátorům dále naskýtají. Následující podkapitoly vycházejí z autorů, z kterých práce čerpá a rovněž z vlastních poznatků nabytých jejím autorem. Každé ze studovaných témat je doprovázeno praktickým zasazením do webové aplikace Ordr.cz.

3.1 Struktura CSS (LESS)

Jestliže je při vývoji webové aplikace použit celý frontendový framework jako např. Bootstrap nebo Foundation, je nutné si adaptovat jejich pojmenovávání tříd, protože se vyskytují napříč celou knihovnou. Knihovny rovněž samy řeší správu a rozdělení souborů obsahujících deklaraci stylů a dalších importů. Situace se mění ve chvíli, kdy nejsou použité celé knihovny, ale maximálně jejich části. Stejně tomu je v případě webového projektu Ordr.cz, kde ze zmíněné knihovny Bootstrap je použit pouze její grid systém, viz předchozí kapitola.

Projekty velikostí jako je tento, je velmi důležité udržovat přehledné již od samého začátku vytvoření prvních kódů a mít definovanou koncepci a normalizaci pojmenovávání tříd, jenž se budou vyskytovat napříč v jeho celé frontendové části. Je nutné počítat s příchodem dalších vývojářů, kteří se budou na projektu podílet. Je rovněž nutné počítat s dalším zvětšováním a rozrůstáním projektu, s přidáváním nových funkcí a nových stránek nebo jejich částí. To všechno jsou důvody, proč mít takové koncepty ujasněné a hlavně explicitně definované, aby je znal každý člen vývojového týmu projektu, ať už nový nebo i současný programátor. Jejich explicitním vyjádřením bude: [8]

- Nastaven stejný standard pro kvalitu kódu napříč celým obsahem
- Podporován soulad kódovaného obsahu
- Dáván vývojářům pocit známosti samotného obsahu
- Zvýšena produktivita (hlavně)

Harry Roberts, anglický frontendový architekt a konzultant společností jako jsou např. Google, Sky, BBC nebo i dokonce Organizace Spojených Národů a rovněž osoba, která výrazně přispívá celému oboru, jímž se práce zabývá, vytvořil veřejně publikovanou CSS příručku¹⁶ *High-level advice and guidelines for writing sane, manageable, scalable CSS* definující metody, postupy a techniky udržování CSS souborů při vývoji projektů jednotlivci i týmy. Autor sám říká, že „*CSS není pěkný jazyk. I když je velmi jednoduché se ho naučit a začít používat, brzy se stane problematickým v jakémkoliv rozumnějším měřítku. Není mnoho, co můžeme udělat pro změnu fungování CSS, ale můžeme provádět změny způsobů, kterým je píšeme a strukturujeme.*“¹⁷ (překlad autora) Rovněž zdůrazňuje, že to jsou ty metody, techniky a tipy, které by rozhodně doporučil jeho klientům a týmům, ale je na vlastním vkusu, jestli budou někým upraveny nebo ne. [8] Přestože v současné chvíli frontend část projektu vyvíjí pouze autor práce, je soubor vytvořený Harrym Robertsem hlavním zdrojem při vedení a strukturování CSS souborů, resp. při použití preprocesoru souborů LESS.

Kořenový adresář LESS souborů

S rychlým vzestupem preprocesorů jako jsou např. LESS, SASS, nebo Stylus, se v poslední době stalo poměrně běžným zvykem rozdělovat soubory zvlášť na jednotlivé kategorie. Konkrétní elementy a třídy jsou tak mnohem lépe přístupnější a dohledatelnější. Rovněž díky media queries definicím pro definování responsivního chování se soubory natahují a díky tomuto přístupu udržují efektivnější a přehlednější sémantiku. [8]

Rozdělení LESS souborů v projektu Ordr.cz bylo inspirováno a postaveno na základě architektury Harryho Robertse. Jednou z hlavních složek kořenu je Base, tedy místo, kam jsou umístěny základní definice jako deklaráce fontu, ikon, proměnných, nebo sprite obrázků – zkrátka elementy, s kterými je pracováno napříč všemi stránkami. Další složka pojmenovaná jako Components, obsahuje všechny důležité komponenty – hlavička s hlavní navigací, patička, tlačítka, formuláře a tabulky. Složka Objects seskupuje soubory, ve kterých jsou definovány obalové elementy (boxy, obaly,

¹⁶ <http://cssguidelin.es/>

¹⁷ „*CSS is not a pretty language. While it is simple to learn and get started with, it soon becomes problematic at any reasonable scale. There isn't much we can do to change how CSS works, but we can make changes to the way we author and structure it.*“ [8]

odsazení atp.), nebo pozadí, která jsou umístována v úvodních obrazech hlavních stránek (např. domovská stránka, jak to funguje, mapa rozvozu, týdenní nabídka atp.). Do složky Pages jsou shromažďovány soubory pro bližší upravování elementů na konkrétních stránkách. Každá stránka má na svém `body` elementu ID selektor nebo třídu, přes jejichž vnořené deklarační je další specifitější stylování dané stránky. Složka Plugins obsahuje vlastní soubory konkrétních rozšíření a knihoven, jako např. Toastr.js¹⁸, který je používán pro frontendové notifikace, nebo Slick.js¹⁹ pro carousel. Poslední složka Settings seskupuje soubory, ve kterých se nacházejí styly pro resetování některých elementů, nebo nastavení pro práci v LESS souborech pomocí knihovny Lesshat²⁰, která obsahuje 86 mixinů značně ulehčujících práci při psaní stylů v jazyce LESS. Shrňme-li seřazení složek a souborů, pak výsledný kořenový adresář ve chvíli vývoje a uváděním této práce vypadá následovně:

- Styles >
 - Less >
 - Base >
 - GridSystem >
 - _colors.less
 - _fonts.less
 - _icons.less
 - _sprites.less
 - _variables.less
 - Components >
 - _buttons.less
 - _footer.less
 - _forms.less
 - _header.less
 - _tables.less
 - Objects >
 - _backgrounds.less
 - _wrappers.less
 - Pages >
 - (example: _nameOfPage.less)
 - Plugins >
 - _toastr.less
 - _slickjs.less
 - Settings >
 - _custom-mixins.less
 - _global.less
 - _lesshat.less

¹⁸ <http://codeseven.github.io/toastr/>

¹⁹ <https://github.com/kenwheeler/slick>

²⁰ <http://lesshat.madebysource.com>

- styles.less

Konvence pojmenování

Další zajímavé rady v příručce se týkají pojmenovávacích konvencí, které jsou velmi užitečné při vytváření kódu, jenž se poté stává transparentnější a více informativní. Dobré pojmenování vývojáři řekne, co daná třída s elementem dělá, kde může být umístěna a s čím dalším může být spojena. [8]

Harry Roberts doporučuje tzv. BEM-like pojmenování. To je velmi přínosné pro větší a vzájemně propojené kusy UI, které vyžadují určitý počet tříd. BEM znamená – blok, element, modifikátor. Blok má vždy jednoslovné pojmenování, např. třída `.meal`. Element uvnitř bloku nese pojmenování vytvořené spojením názvu jeho nadřazeného bloku a svého vlastního pojmenování odděleného a mezi ně jsou vložena dvě podtržítka, např. `meal__footer`. Modifikátor je rozšířením určitého elementu nebo i bloku, např. chceme-li blok `.meal` mít menší, může mu být přidána třída `.meal--sm`.

Modifikátory jsou mj. rovněž často používány i k rozlišení tlačítek. Ve webu `Ordr.cz` se nachází dva druhy hlavní tlačítek. Každé z nich nese třídu `.btn`. Zelená tlačítka jsou vždy spojená s objednávkou a nesou modifikační třídu `.btn--primary`, která tlačítko obarví do zelena. Jsou primárním cílem služby – objednat jídla. Oranžová tlačítka pak nesou svou modifikační třídu `.btn--secondary` a jsou použity např. při uložení adresy, platební karty, nebo při přihlášení či registraci. Při vývoji HTML je vhodné se vyvarovat pojmenování jako např. `.btn--green` nebo `.btn--orange`, protože v případě změn hlavních barev webu nebo služby jako takové, by se musely přepisovat všechny třídy tlačítek napříč všemi HTML kódy. Při vyvarování se tohoto zlovyku poté pouze stačí změnit barvu v LESS proměnné a to se promítne do všech tlačítek s těmito třídami. Pojmenování konkrétní barev je tedy vhodné nepoužívat nejen na tlačítka, ale na všechny elementy.

3.2 Navigační menu

Velmi důležitou komponentou celého webu je vždy navigační menu. Před příchodem responsivního web designu nebyly nabídky nijak omezené. Daly se umístit v podstatě kamkoliv a v jakémkoli provedení. Po příchodu razantně menších zobrazovacích ploch ještě nějakou dobu grafičtí designéři nevěděli, kam a hlavně jak nabídky

zobrazovat při takových omezeních a velmi často se stávalo, že někdy až 60 položek víceúrovňové navigace bylo umístěno do prostoru, který uživatel po příchodu na mobilní web viděl jako první a vlastně ani neviděl samotný obsah, který by očekával. Takový přístup pochopitelně nebyl vhodný a bylo nasnadě se tématem dále zabývat a vyvinout lepší a přístupnější principy. [3]

Dotykové displeje a minimální velikosti prvků

Scott Jehl ve své knize zmiňuje „palcové pravidlo“, když říká: „Zařízení, z kterých uživatelé na váš web přistupují, mohou i nemusí mít dotykový displej, ale vždy elementy navrhujte tak, jako když ho mají“²¹ (překlad autora). Dále odkazuje na článek *Finger-Friendly Design: Ideal Mobile Touchscreen Target Sizes* publikovaný serverem Smashing Magazine, který se týká prsty snadno ovládaných velikostí elementů na displejích s dotykovým ovládaním. [7]

Příručky pro designery napříč všemi mobilními platformami uvádí rozdílné velikosti minimálních rovinných útvarů, které jsou, podle nich, bezproblémově ovladatelné konečky prstů. Apple ve svém dokumentu uvádí minimální velikost 44x44 pixelů. Microsoft pro vývoj Windows Phone návrhů vyměřil 34x26px, podle Nokie 28x28px a co není v článku obsaženo, je velikost pro systémy Android, která podle Google příručky²² s metrikami činí 48x48dp, což je v přepočtu na pixely nejméně 36x36px. [9]

Rozdílnost minimálně doporučených velikostí výrobci přináší tomuto tématu nepřehlednost a navíc se většina z nich neshoduje s reálnou velikostí konečků prstů lidské ruky. Právě tím se zabývala studie provedená na MIT v Cambridge už v roce 2003, ze které vychází citovaný článek ze serveru Smashing Magazine. Vědci v ní zjistili, že průměrná velikost konečků prstů dospělého člověka je 1,6 až 2 centimetry, což je po převedení na pixely 45-57 a to je více než výrobci software doporučují. Článek doplňuje i velikost palců, která je průměrně okolo 2,5cm. Ideálně ovládaný prvek palci by podle článku měl být 72 pixelů široký. Je tedy zřejmé, že čím menší prvek, tím hůře se stává ovladatelným, resp. dotknutelným, a tím více vzniká riziko

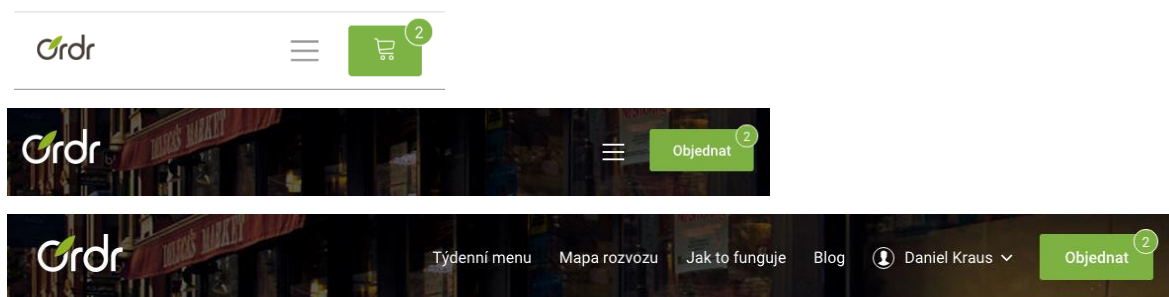
²¹ „One rule of thumb(s): the devices accessing your site may or may not have touch screens, but always design as if they will.“ [7 str. 27]

²² <https://www.google.com/design/spec/layout/metrics-keylines.html>

chyb. [9] Při implementaci responsivního designu do webové aplikace je na tento faktor velmi důležité myslet.

Navigační menu v aplikaci Ordr.cz

Analýza velikostí zmíněných v předešlých odstavcích vede k určení šíře a výše prvků tlačítek v mobilní navigaci, resp. v pásu, ze kterého je mobilní navigace dostupná. Obrázek 5 ukazuje všechny možné velikosti a režimy pásu s navigační nabídkou. Na obrázku první pás (shora) představuje mobilní verzi, která nese daný vzhled až do velikosti < 768px. Druhý pás je aplikovaný pro velikost od 768px až po 970px, tedy velikost tabletů. Třetí, poslední pás, představuje již klasickou ihned viditelnou navigaci na desktopových velikostech průhledů.



Obrázek 5: Způsoby hlavní navigace ve webové aplikaci Ordr.cz

V prvních dvou případech navigačního pásu je vlevo od zeleného tlačítka Objednat umístěné tlačítko (tzv. hamburger) s trigger vlastností, která po kliknutí rozbalí nebo zavře navigační nabídku, viz Obrázek 6. Trigger tlačítko má velikost 44x44px, tedy velikost uvedenou v příručce Apple grafickým designerům. Samotný prvek tří malých proužků pod sebou, v režimu před rozbalením nabídky, je o 20px užší, avšak kolem jeho obvodu se i tak opticky prázdné místo nachází v dotykovém poli zabezpečující správnou funkcionalitu tlačítka. Další zajímavou vlastností tohoto prvku je jeho vytvoření pouze pomocí elementu `span` vnořeného v elementu `button` a dále upravením v CSS, nikoliv pomocí obrázku. Po kliknutí na tlačítko se opět pomocí CSS a přidáním třídy `active` pomocí jQuery prvek změní na křížek nebo naopak. Následně rozbalená nabídka má nastaveno `position:fixed;` a 0px vzdálenost po všech stranách, čímž se roztáhne přes celou obrazovku průhledu a všechny její položky jsou velmi pohodlně a přehledně dostupné.

Při návrhu mobilní navigace bylo nejdříve nutné si ujasnit, kde bude umístěné tlačítko pro přechod do košíku a tlačítko pro rozbalení navigační nabídky. Umístění vpravo nahoře i ve své analýze míst dobře dostupných ovládním telefonu jednou (pravou) rukou označuje Scott Jehl [7 str. 73] jako v pořádku a dále je obecně známým faktem, že uživatelé si nejdříve všimají prvků umístěných na pravé straně.

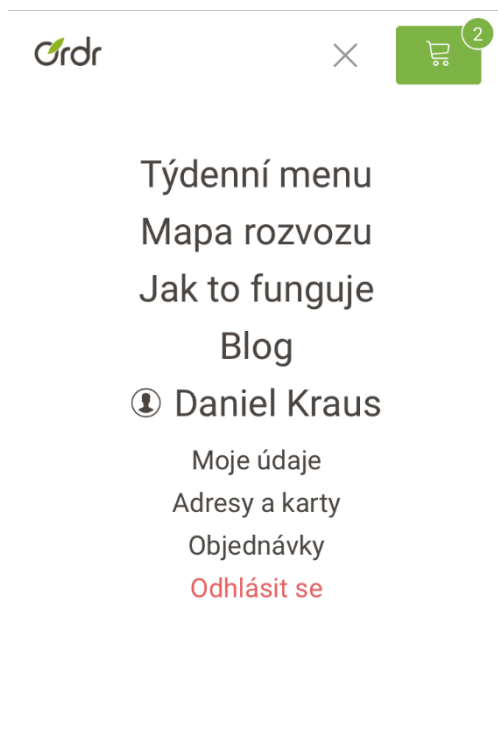
Tlačítko pro objednání se v pásu nachází pouze v případě, že je aktivní rozvoz jídla. Pokud není, tlačítko se pro v tu chvíli nedostupný košík stává nepotřebným, zmizí a trigger tlačítko se posouvá do pravé strany. Velikost tlačítka s košíkem je na mobilním zařízení

široké 64 pixelů a vysoké 44 pixelů. Z pohledu šířky je tedy určitým kompromisem vybraným mezi průměrnými velikostmi prstů a palců, uvedených v předešlé části podkapitoly a je dobře ovladatelné.

Velmi důležitým faktem při vývoji navigačního menu je zachování jednoty HTML kódu tak, aby se nemusely určité prvky dále přesouvat v těle dokumentu pomocí JavaScriptu, nebo aby mobilní navigace měla dokonce vlastní kód. Tomu bylo v případě webové aplikace Ordr.cz předejito a celý navigační pás, jak ukazuje Obrázek 5, tvoří pouze jeden HTML kód, viz Příloha 1, který má pro různé velikosti jiné CSS definice.

3.3 Práce s obrázky

Je zřejmé, že každý grafický design obsahuje řadu obrázků a pokud jde o velikosti souborů, které se při načítání stránek stahují přes HTTP požadavky ze serveru, jsou právě obrázky často největším nepřítelem. Podle Scotta Jehla každou stránku s průměrným objemem 1,7 MB obrázky tvoří 61 % a problém se nadále začal zhoršovat s tím, jak se široká škála velikostí a rozlišení zobrazovacích zařízení stala různorodější. Tuto kritickou kategorii naštěstí napomáhají řešit prohlížeče,



Obrázek 6: Rozbalená mobilní navigace

protože oproti CSS a JS souborům se obrázky stahují ze serveru asynchronně, nebo alespoň neblokují vykreslování stránky. I přesto se vyskytuje ve spojení s obrázky řada faktorů, které mohou výkony stránek omezit a na něž se frontend vývojáři musí zaměřovat. [7 str. 127] V následující části bude zkoumáno a implementováno několik technik a postupů, které je nutné při vývoji webu řešit.

3.3.1 Obrázky na retina displejích

Obrázky lze do webového obsahu dostat dvěma hlavními cestami - deklarací v CSS nebo vložením pomocí patřičných tagů přímo v HTML. Do chvíle, kdy nebyly displeje s větší hustotou pixelů, byly obě cesty velmi snadné. S příchodem prvních retina displejů se obě varianty zkomplikovaly a bylo nutné najít jejich řešení.

Použití v LESS

První možností, jak do webového obsahu vložit obrázek s podporou vyšší hustoty rozlišení, je pomocí CSS, resp. LESS. Scott Jehl se tématem ve své knize taktéž zabývá a ukazuje možné řešení. „*Definice `min-resolution` v media query nám dovoluje cílit na zařízení s konkrétní, nebo vyšší hustotou pixelů obrazovky (záložní definice `-webkit-min-device-pixel-ratio` pomáhá řešit podporu jiných prohlížečů). V příkladu jsem použil 144dpi, protože je to 2x více než standardní obraz s 72dpi a dobrý základ pro HD obrazovky (které nyní jdou ještě výše než je 144dpi).*“²³ (překlad autora)

```
.image2x(@image, @repeat: no-repeat) {
  @filename : ~`/(.*)\.(jpg|jpeg|png|gif)/.exec(@{image})[1]`;
  @extension : ~`/(.*)\.(jpg|jpeg|png|gif)/.exec(@{image})[2]`;
  background-image: ~`url(/Content/img/{filename}.{extension})`;
  background-repeat: @repeat;

  @media
    (-webkit-min-device-pixel-ratio: 1.44),
    (min-device-pixel-ratio: 1.44),
    (min-resolution: 144dpi) {
    // on retina will be used image scaled by 2
    background-image: ~`url(/Content/img/{filename}@2x.{extension})`;
  }
}
```

Zdrojový kód 6: LESS mixin pro vložení obrázku s automatickou podporou retina rozlišení

²³ „*The min-resolution media query lets us target devices with screens at or above a particular dpi (vendor-specific fallbacks such as `-webkit-min-device-pixel-ratio` help round out support here too). I've used 144dpi in this example because it's twice the standard screen definition of 72 dots per inch, and a good baseline for HD screens (which now often go even higher than 144 dots per inch)*“ [7 str. 128]

Zdrojový kód 6 vytvořený autorem práce je přepracováním příkladu Scotta Jehla pro snadnější a hlavně efektivnější použití ve stylovacím jazyce LESS. Vlastní mixin vytvořený v souboru `_custom-mixins.less` v kořenovém adresáři projektu umožňuje velmi snadnou deklaraci `background-image` s automatickou podporou HD rozlišení. Konkrétnímu elementu stačí jedním řádkem vložit mixin v podobě:

```
.image2x („filename.png“)
```

Ještě před zmíněným mixinem je nutné vložit definici `background-size` s originální velikostí obrázku, nebo s takovou velikostí, ve které se bude obrázek v obsahu vykreslovat. Po kompilaci je automaticky v media query vložená definice s cestou v adresáři na jeho dvojnásobnou velikost. Je pouze nutné, aby dvojnásobně větší obrázek měl ve svém jméně na konci vložené označení `@2x`. Poté v případě, je-li detekováno zařízení s vyšší nebo rovnou hustotou pixelů rozlišení 144dpi, bude použit dvojnásobně větší obrázek, který se pomocí `background-size` zmenší na jeho minimálně poloviční velikost a tím se zajistí jeho ostrost na těchto displejích. Původní obrázek bude serverem ignorovaný a nebude se z něho stahovat.

Použití v HTML

O něco problematičtější je podpora vyšších hustot rozlišení, má-li být obrázek vkládán v HTML. Tím se Scott Jehl ve své knize také zajímá a to mnohem detailněji, než v předchozím případě.

Obrázky, které jsou vkládány přímo do HTML, dávají větší sémantickou hodnotu samotnému obsahu. Svůj význam mají např. v článcích nebo seznamu jídel, kde v první řadě plní informační účel. Pro zajištění responsivního chování obrázků z pohledu jejich šířky jim stačí pouze definovat `max-width:100%;` nebo `width:100%;`, čímž je zajištěna přizpůsobivost velikosti buď šíře layoutu, nebo jeho nadřazeného blokového elementu (záleží, kde je obrázek umístěn). [7]

Obvyklým řešením problému, které někteří vývojáři do svých webových stránek implementují, je nahrání největší velikosti a následně její zmenšení při vykreslení v prohlížeči, což zajistí jeho ostrost. Scott Jehl o tomto přístupu říká, že „*taková*

*praktika vkládání těch největších obrázků úplně všem není příliš odpovědná – lidé musí stahovat mnohem víc dat, než jejich zařízení potřebují*²⁴ (překlad autora)

Možnost vkládání responsivních obrázků přímo v HTML začalo řešit už v roce 2012 W3C konsorcium a rozhodlo se HTML5 specifikaci přidat nové elementy a atributy, jenž problém velkého množství vývojářů, vyřeší. Zasedající skupina jazyku přidala `picture` element a k němu spojené nové atributy `srcset`, `sizes`, `media` a `type`, které byly postupně přidávány do seznamů podporovaných elementů moderními prohlížeči. Pro ty, jenž elementy a atributy nepodporují, vytvořil autor Scott Jehl velmi jednoduchou a datově lehkou knihovnu `Picturefill.js`²⁵, kterou mj. představuje i ve své knize a rovněž nové prvky detailně popisuje. [7] Knihovnu `Picturefill` doporučuje rovněž i ve své knize Fielding [4].

Nové atributy není nutné používat pouze na elementu `picture`, který umožňuje hlubší integraci a specifikaci responsivních obrázků, např. podle jejich různých formátů atp. (více rozebírá Scott Jehl [7] i Jonathan Fielding [4]). Atribut `srcset` lze použít i na známý element `img`. Implementaci na konkrétním příkladu v aplikaci `Ordr.cz` ukazuje Zdrojový kód 7, jenž pomocí atributu vkládá obrázek ilustrující mobilní aplikace pro iOS a Android na domovské stránce v sekci s odkazy na jejich stažení. Přínos a jednoduchost použití atributu jsou ze zdrojového kódu zřejmé. Za každým vloženým obrázkovým souborem je definovaná hustota pixelů obrazovky, na nichž se obrázek má zobrazit a vždy je použita vlastní originální velikost toho obrázku, který je definovaný s označením 1x. Není-li detekovaná konkrétní hustota, druhý soubor je serverem ignorovaný.

```
<img id="platforms-mockup"
  srcset="@Url.Content("/Content/Img/platforms-mockup.png") 1x,
         @Url.Content("/Content/Img/platforms-mockup@2x.png") 2x"
  alt="Ordr Mobile Apps">
```

Zdrojový kód 7: Použití atributu `srcset` na elementu `img`

W3C spolu se Scottem Jehlem, který elegantně vyřešil podporu nových elementů a atributů díky jeho JavaScriptové knihovně následně importované v dokumentu

²⁴ „Unfortunately, this practice of serving large images to everyone isn't very responsible—people end up loading far more data than their device needs“ [7 str. 132]

²⁵ <https://scottjehl.github.io/picturefill/>

HTML, výrazně pomohly vyřešení problému a lze tak nadále vytvářet sémanticky správný kód.

3.3.2 Ikony jako font

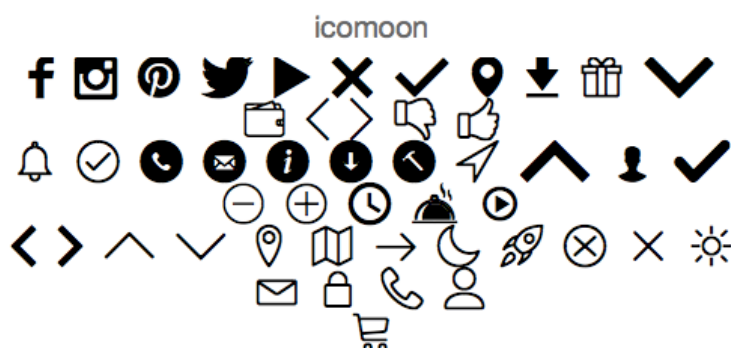
Na každém webu se vždy vyskytne velké množství drobných ikon, jako jsou např. šipky, křížky, ikony sociálních sítí, ikony ilustrující siluety uživatelů, obálky atp. Je vysoce nepraktické tyto ikony vkládat jako samostatné obrázky nebo sprite obrázky, zvláště existuje-li jiná cesta, jak k tomuto případu přistupovat. Řešení problému použitím setů ikon jako fontů není žádnou novinkou. Tato metoda se již využívala ještě před nástupem responsivního web designu. V tu dobu jedinou nevýhodou byla kompatibilita s prohlížeči, ale nyní se postupem času tento nedostatek vyřešil (ve většině případů) sám a nasazení způsobu nahrazení drobných vektorových obrázků fontem se stalo velmi populárním. Navíc z hlediska výkonu stránky a délky jejího načítání, je metoda výbornou volbou, protože soubor s fontem obsahujícím ikony se stahuje přes jediný HTTP požadavek. [7] Další výhodou, proč se pro takové použití rozhodnout, je možnost libovolného přebarvení ikon pomocí CSS přes definici `color` nebo nastavení jejich velikostí pomocí `font-size`, což je v případě obrázků nebo spritů nemožné.

Nástroj pro vytvoření setu – IcoMoon

V případě frameworků Bootstrap či Foundation se vývojáři otázkou vytvoření setu ikon pro jejich weby zabývat nemusí, protože jsou v nich obsaženy, ale je vysoce pravděpodobné, že ikony se v síti webových stránek budou opakovat i na ostatních webech, čili nebudou tak unikátní. Webová aplikace Ordr.cz, která má svůj grafický design dělaný na míru, nemá rovněž striktně všechny ikony vlastní, ale valnou většinu ano. Proto je nutné set ikon ex post vytvořit. K tomu slouží velice šikovný online nástroj nazvaný IcoMoon App dostupný z <https://icomoon.io>. Aplikace umožňuje nahrání jednoduchých vektorových obrázků ve formátu svg, jejich následnou úpravu (velikosti, zvětšení průhledných míst, pojmenování atp.) v prohlížeči a poté set sama vygeneruje. Vygenerovaný balík obsahuje CSS soubor obsahující import souborů fontu pomocí `@font-face` a deklarace tříd jednotlivých ikon. Soubory obsahující samotné ikony jako fonty jsou v balíku ve všech potřebných formátech:

- .woff – formát woff
- .ttf – formát true type
- .eot – formát embedded opentype
- .svg – formát svg

Další užitečnou vlastností nástroje je vytvoření a uložení projektů pro jejich pozdější správu. To znamená, že kdykoliv grafický designer nebo i vývojář budou chtít přidat do webu nové ikony, vloží se do stávajícího setu v nástroji, nově vygenerují a aktualizují se nimi soubory v projektu. Set ikon vytvořených pomocí nástroje IcoMoon ilustruje Obrázek 7.



Obrázek 7: Set ikon z Icomoon použitý v aplikaci

Implementace ikony v HML

Vygenerovaný CSS soubor je převeden do kořene s LESS soubory uvedeného na začátku kapitoly jako `_icons.less`. Poté je už velmi jednoduché pracovat s ikonami v HTML, jak ukazuje Zdrojový kód 8 na příkladu s vloženou ikonou emailu na elementu `span` v položce patičky webu. Umístění právě na element `span` není náhodné. Je to jeden z nevhodnějších kandidátů na toto použití. Například framework Bootstrap vkládá ikony primárně na tag `<i>`, což je sémantický problém, protože tento element je určen na práci s čistým textem, kterému pak obvykle aplikuje kurzívu. Oficiální dokumentace W3C o tomto tagu říká, že „*může být použit k označení technického pojmu, citaci z cizího jazyka, myšlenky, vlastního jména atp.*“²⁶ (překlad autora). Z toho je jasné, že Bootstrap tento tag pro ilustraci ikony používá chybně.

²⁶ „The `<i>` tag can be used to indicate a technical term, a phrase from another language, a thought, or a ship name, etc.“ [13]

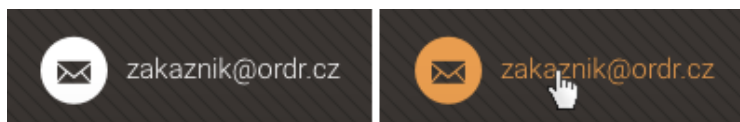
```

<li>
  <a class="mail" href="mailto:zakaznik@ordr.cz">
    <span class="icon icon-circle_email"></span>zakaznik@ordr.cz
  </a>
</li>

```

Zdrojový kód 8: HTML kód s ikonou v položce patičky webu

Velmi výhodným bonusem tohoto řešení je, že ikony se dají snadno přebarvovat i pro selektor `:hover`. To je v případě použití obrázků značně omezující a taková situace musí být řešena např. přes `background-position`, což je v podstatě použití sprite obrázku, ve kterém jsou oba barevné stavy, a při najetí kurzorem myši se v elementu obrázek posune na přebarvenou ikonu. Jiným možným efektem může být redukce průhlednosti pomocí `opacity`, avšak je to stále velmi omezující. Výsledek použití ikony jako fontu ilustruje Obrázek 8.



Obrázek 8: Hover efekt na ikoně položky v patičce

3.3.3 Kompresie obrázků

Při frontend implementaci je nutné myslet na to, aby byly do prohlížečů posílány obrázkové soubory tak lehké na datový objem, jak je jen možné. Toho lze docílit jejich kompresí, lépe řečeno bezztrátovou kompresí. Ta neubírá na vlastní kvalitě obrázku a pouze redukuje jeho datový objem. Tématem se zabývají rovněž Jonathan Fielding i Scott Jehl ve svých knihách, kde představují mj. nástroje ImageOptim²⁷, který kompresi provádí. [7 str. 101; 4 str. 270]

Webová aplikace Ordr.cz, včetně jeho administračního prostředí, je vyvíjena v ASP.net, což znamená, že s projektem se pracuje v prostředí Visual Studio (konkrétně ve verzi 2013), do kterého lze nahrát rozšíření Web Essentials²⁸ pro webové vývojáře. Toto rozšíření mimo jiné také umožňuje kompresi obrázků a to velmi jednoduše, pouze jedním kliknutím na daný obrázkový soubor v kořenovém adresáři.

²⁷ <https://imageoptim.com>

²⁸ <http://vswebessentials.com>

Tabulka 1 představuje výsledky pokusu, při němž bylo vybráno 5 obrázkových souborů ve formátech jpg nebo png. Porovnává ušetření datového objemu obrázků provedeného pomocí Web Essentials a online webového nástroje Kraken.io²⁹. Z tabulky je patrné, že nástroj Kraken.io z celkového počtu datového objemu všech pěti obrázků ušetřil až 60 % dat a rozšíření implementované ve Visual Studiu 2013 pouze přibližně 25 %. Bylo rovněž provedeno testování nástroje doporučeného autory Jehlem a Fieldingem. V jeho případě bylo ušetřeno pouhých 9 % dat.

Formát obrázku	Originální velikost	Komprese - VS 2013	Komprese - Kraken
jpg	156,74 kB	147,92 kB	121,88 kB
png	327,65 kB	238,6 kB	82,19 kB
png	30,99 kB	21,88 kB	11,89 kB
png	838,79 kB	494,99 kB	484,04 kB
jpg	614,10 kB	573,04 kB	83,39 kB
Celkem:	1968,27 kB	1476,43 kB	783,39kB

Tabulka 1: Porovnání kompresí obrázků pomocí nástrojů VS 2013 a Kraken.io

Zdarma dostupný online software Kraken s jednoduchou drag-and-drop vlastností importování obrázků lze po analýze označit jako nejlepší variantu pro práci s kompresí z porovnávaných nástrojů. I přestože je nutné při vývoji přecházet do jiného prostředí, značné ušetření datových objemů dokázané v Tabulka 1 může velmi ovlivnit očekávanou optimalizaci.

3.3.4 Lazy loading

Jednou z dalších optimalizačních metod pro zlepšení rychlosti načtení stránky snížením datového objemu a počtu HTTP requestů na server je metoda nazvaná lazy loading. Autoři Sharkie a Fisher ve své knize [5 str. 115] tuto techniku řadí mezi jeden z velmi důležitých postupů při vývoji responsivního webu. Je to způsob, jak velmi jednoduše a inteligentně zobrazovat obrázky uživatelům teprve až ve chvíli, kdy je potřebují. To znamená, že když uživatel scrolluje dolů po směru obsahu stránky, obrázky se postupně načítají až v případě, překročí-li hranici viditelné části průhledu.

²⁹ <https://kraken.io>

Knihovna Unveil.js

Ve stejnojmenném článku [10] se touto metodou zabývá i český autor s 14 letou praxí v oboru Martin Michálek, ve kterém představuje knihovnu Unveil.js³⁰ umožňující implementaci metody do webu. Stejně jako autoři Sharkie a Fisher říká, že „z pohledu responzivního webdesignu jde o nepostradatelnou techniku“. Knihovna je inspirovaná pluginem nesoucím název *Lazy Load Plugin for JQuery*³¹, který v podstatě umožňuje to samé, ale v případě Unveil.js se autorovi podařilo ještě zmenšit její objem pod 1kB a rovněž přidat podporu retina displejů.

Ve webové aplikaci Ordr.cz byla tato metoda implementována na stránkách s týdenní nabídkou nebo seznamem nabízených jídel. V obou případech se vyskytují 400x275px velké obrázky (v maximální desktopové verzi) s průměrným objemem okolo 50kB. Na stránce s týdenní nabídkou se vždy vyskytuje maximálně 10 těchto obrázků, tzn. přibližně 500kB. Stránka Naše jídla představuje až 80 jídel, což je v převedení na jejich datový objem až 4MB a to už je enormní ulehčení při načítání této části webu. Jak byla metoda implementována v HTML, ukazuje Zdrojový kód 9. Zdrojový kód 10 pak ukazuje inicializaci funkce LazyLoad na tomto elementu v jQuery.

```
<div class="img-wrap img-wrap--meal img-wrap--link">
  <a href="@Url.Action("Detail","Meals", new { id = Model.ElementAt(i).Id})"
    class="img-wrap__overlay">
    <span class="overlay__more">Zjistit více o jídle</span>
  </a>
  
</div>
```

Zdrojový kód 9: Použití lazy loadu obrázku v HTML

„Důležitý tip: řada webařů (včetně autora Unveil v demonstračním příkladu) propadla vášnivě lásce k preloaderům a do `src=""` jako výchozí obrázek dává takovýten točící se kolovrátek. Nedělejte to prosím. Na uživatele působí daleko lépe, když tam před načtením obrázku vidí barevnou plochu mírně odlišenou od barvy pozadí.“, říká ve svém článku Martin Michálek [10]. Stejně tak nebyl preloader použit ani v případě aplikace. Umístění obrázku se třídou `.img--lazy` je obalené v elementu `div`, který má

³⁰ <http://luis-almeida.github.io/unveil/>

³¹ <http://www.appelsiini.net/projects/lazyload>

na pozadí placeholder s obrázkem poklice na talíři a celý tento element rovněž slouží jako odkaz na detail jídla. Ten je zajištěn hyperlinkem s třídou `.img-wrap__overlay` a absolutní pozicí přizpůsobenou velikosti `img` elementu. Efekt odkazu se světlou vrstvou a textem „Zjistit více o jídle“ se zobrazí při selektoru `:hover`.

```
.img--lazy{                                $( ".img--lazy" ).unveil(200, function () {
  .opacity(0);                               $(this).load(function () {
  .transition(opacity 0.3s ease-in);         this.style.opacity = 1;
}                                             });
}                                             });
```

Zdrojový kód 11: LESS deklarace umožňující fadeIn efekt při lazy loadu

Zdrojový kód 10: jQuery inicializace lazy loadu na obrázek

Pro dosažení poutavého efektu, který umožní jemné prolnutí elementu, lze obrázku nastavit nulovou hodnotu `opacity` a definovat `transition`, jak ilustruje Zdrojový kód 11. Ve chvíli, kdy se obrázek přiblíží hranici viditelného viewportu ve vzdálenosti 200px se třídě `.img--lazy` nastaví `opacity` na hodnotu 1, což provede daný efekt, viz Zdrojový kód 10.

3.4 Možnosti uložení mobilního webu na plochu

Stejně jako v běžných případech, kdy si uživatelé chtějí uložit svůj oblíbený web jako záložku v prohlížeči na svých počítačích, lze ukládat weby i na smartphonech. Díky vývojářům z Applu lze uložit web nejen do prohlížeče, ale i přímo na plochu zařízení. Vytvořili pro to meta tagy, které se již staly standardem, ale ne všechny z nich jsou příliš známé. Autoři se možnostmi této funkce téměř nezabývají, zmiňují ji pouze okrajově [5 str. 134; 6 str. 48] a dále nezkoumají, jaká jsou další rozšíření, kterých lze dosáhnout pomocí těchto tagů.

V oficiální dokumentaci Applu pro iOS developery³² je kapitola o možnostech konfigurace webových aplikací [11]. Uvádí, že u těchto aplikací lze dosáhnout toho, aby vypadaly a chovaly se velmi podobně těm nativním, například tím, že web je možné zvětšit na celou obrazovku zařízení, odstranit nativní ovládací prvky prohlížeče Safari a vytvořit na ploše ikonu jako mají rovněž nativní aplikace. S čím se ovšem dokumentace rozporuje s realitou, je tvrzení „*Tato všechna volitelná*

³² <https://developer.apple.com/library/ios/navigation/>

nastavení, která lze přidat do vašeho webového obsahu, jsou ignorována ostatními platformami.“³³ (překlad autora) To totiž již delší dobu neplatí a některá, ne všechna, nastavení fungují i pro platformu Android. V nativních prohlížečích ji umožňují jeho poslední verze. V následujících částech této kapitoly byla pro potvrzení tohoto tvrzení provedena testování pomocí emulátorů systémů Android v softwaru BrowserStack³⁴.

Jedním z možných scénářů a důvodů nutnosti se touto funkcí zabývat, je případ, kdy samotná nativní aplikace již nepodporuje konkrétní OS a po implementaci patričních meta tagů ji tak lze na daném systému nahradit alternativou - webem. Takový scénář může nastat i v případě nativních aplikací Ordr.cz, protože na nižších verzích operačních systémů iOS 7 a Android 2.3.3 je není možné nainstalovat – nepodporují takové systémy. Dalším důvodem uložení webu Ordr.cz může být i fakt, že mobilní nativní aplikace nejsou tak komplexní a není v nich dostupných tolik informací, jako je nabízených na webu. Jestli se uživatelé chovají jako v těchto dvou případech, je možné zjistit pomocí JavaScriptu, kde lze detekovat případ standalone režimu ve window.navigator a poté odeslat událost s údajem např. do Google Analytics. To bohužel nebylo v případě webové aplikace Ordr.cz v průběhu prvního roku jejího nasazení implementováno a nelze tedy takové hypotézy potvrdit.

Ikona

Vzhledem k široké škále mobilních rozlišení na trhu se vyskytuje řada velikostí, které ikony mohou mít. Velikosti ikon aplikací pro iOS a konkrétní zařízení lze zjistit v Apple příručce³⁵ pro grafické designery. Vložení následujícího tagu do hlavičky HTML se tato funkce aktivuje [11].

```
<link href="apple-touch-icon.png" rel="apple-touch-icon-precomposed" sizes="152x152">
```

Je možné definovat více velikostí, ale každá velikost musí být umístěná ve vlastním elementu link a poté je až přístrojem vybrána ta nejvíce (nejblíže) vyhovující velikost pro dané zařízení. Často se v praxi stává, že je definována ta největší (192x192px) z důvodu zajištění ostrého zobrazení na retinových displejích a na menší rozlišení se

³³ „These are all optional settings that when added to your web content are ignored by other platforms“ [11]

³⁴ <https://www.browserstack.com/>

³⁵

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/IconMatrix.html>

nebere ohled, protože se ikona zmenší sama. Tento přístup neškodí aplikaci samotné, neprojevuje se na rychlosti jejího chodu atp., ale může se stát, že v případě graficky náročnějších ikon, se na menších rozlišení automatickým zmenšením znehodnotí její kvalita. Jak poté ikona v případě webu Ordr.cz vypadá na konkrétním zařízení, ukazuje Obrázek 9 (první telefon zleva). Testování potvrdila, že ikony se na plochu zařízení ukládají jak na platformě iOS, tak Android.

Nastavení jména aplikace

Při ukládání webové aplikace na plochu operační systém vždy zobrazí nabídku pro libovolné pojmenování aplikace. Není-li v HTML obsažen následující kód, systém do řádku pro editaci názvu vloží hlavní meta titulek stránky.

```
<meta name="apple-mobile-web-app-title" content="Ordr.cz">
```

V tomto případě bude při ukládání na plochu na iOS předvyplněno pojmenování Ordr.cz, které si ovšem uživatel může v případě potřeby změnit, ale pouze při ukládacím procesu, ex post editace názvu už není možná. Testování provedené na nejnovějších verzích Android ukázalo, že tato hodnota atributu v meta tagu je systémem ignorována a je přiložen obsah z hlavního meta titulku webu.

Uvítací obraz

Velmi podobného chování uloženého webu nativní aplikaci lze rovněž dosáhnout nastavením uvítací obrazovky, která se zobrazuje vždy při spouštění aplikace. Oficiální dokumentace Applu [11] není při nastavení této možnosti příliš podrobná a neukazuje podmínky použití pro více jejich zařízení. Rovněž žádný z autorů se tímto rozšířením vůbec nezabývá.

Zde už bohužel neplatí možnost nahrání největšího obrázku, který se bude na různých menších rozlišení zmenšovat a přizpůsobovat jako tomu bylo v případě ikony. Podstatný je při vývoji fakt, že tato funkce se projeví v současnosti pouze na zařízeních s operačním systémem iOS, ostatní ji ignorují. Proto je důležité znát cílová rozlišení těch Apple zařízení, na kterých má být otevírání doprovázeno efektem uvítacího obrazu. Na základě rozlišení všech Apple mobilních iOS zařízení³⁶ byl sestaven následující seznam velikostí zobrazovacích ploch bez zahrnutí stavového

³⁶ <http://ios-resolution.com>

pásku (status baru), který má výšku 10px na klasickém rozlišení a 20px na retina rozlišení. (Pozn. autora: V závorkách jsou uváděny logické velikosti, tedy velikosti bez ohledu na hustotu pixelů.):

- 1536x2008 (768x1004) - iPad s retina displejem
- 768x1004px - iPad bez retina displeje
- 1242x2148 (414x716) - iPhone 6 Plus
- 750x1294 (375x647) - iPhone 6
- 640x1096 (320x548) - iPhone 5 a 5S a iPod Touch 5. generace
- 640x920 (320x460) - iPhone 4 a 4S
- 320x460 - iPhone 3G a 3GS

Právě v těchto velikostech je nutné připravit obrázkové soubory a importovat je v HTML způsobem, jak uvádí Zdrojový kód 12. Představu následného spouštění aplikace s tímto efektem ilustruje Obrázek 9 (druhý telefon zleva).

```
<!-- Startup image for iPhone 6 -->  
<link href="@Url.Content("/Content/Assets/Apple_Startups/apple-startup-750x1294.png")"  
      media="(device-width: 375px) and  
            (device-height: 667px) and  
            (-webkit-device-pixel-ratio: 2)"  
      rel="apple-touch-startup-image">
```

Zdrojový kód 12: HTML deklarace uvítacího obrazu pro uložený mobilní web

Skrytí ovládacích prvků prohlížeče

V pokračování optimalizace uložení webu na plochu, za dosažením co nejpodobnějšího vzhledu nativních aplikací, lze rovněž odstranit ovládací prvky prohlížeče jako je pole pro psaní URL adresy a panel s tlačítky na spodní straně prohlížeče pomocí následujícího kódu. [11]

```
<meta name="apple-mobile-web-app-capable" content="yes">
```

Web se poté otevírá ve fullscreen (standalone) režimu a zůstává pouze status bar. Zde se oficiální dokumentace pro iOS vývojáře opět mýlí, když uvádí, že tato funkce platí pouze pro operační systém mobilních zařízení Apple, protože provedeným testováním např. na zařízení Samsung Galaxy S5 s verzí Androidu 4.4 i 5, bylo zjištěno, že se tyto komponenty po spuštění aplikace přímo z plochy odstranily taktéž.

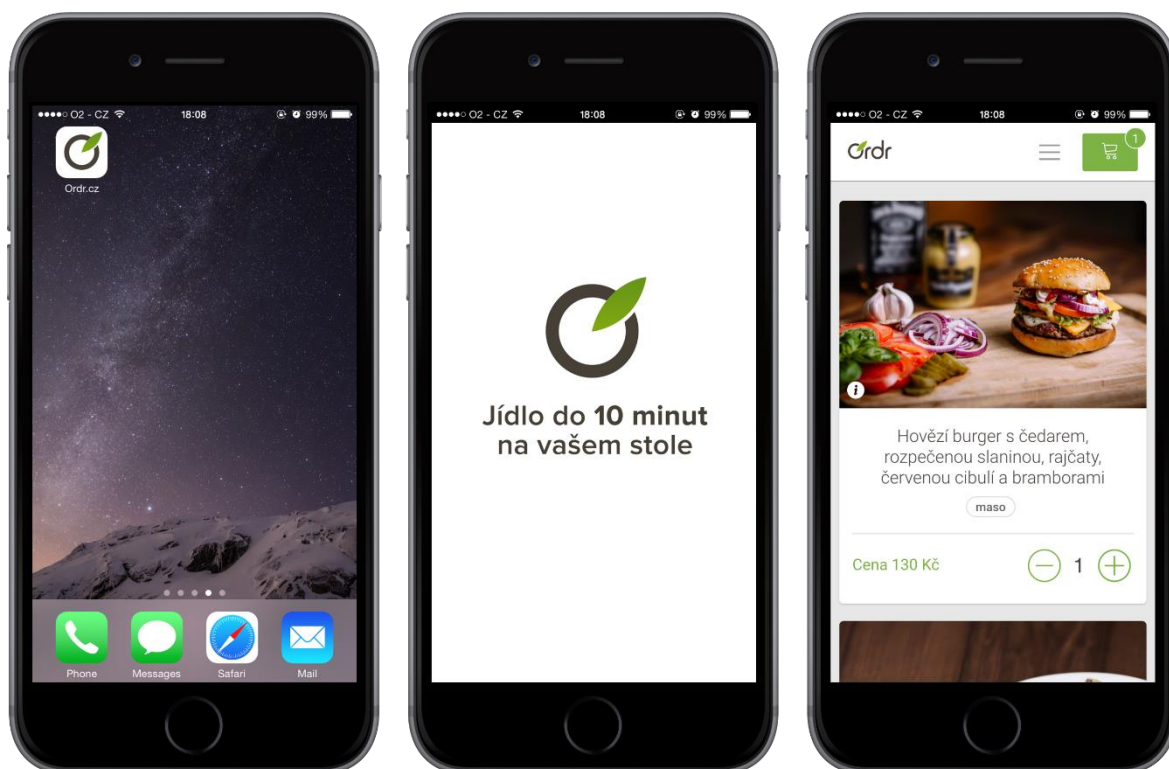
Případ bez zobrazení ovládacích prvků prohlížeče ilustruje jako předchozí dvě funkce Obrázek 9 (třetí telefon zleva).

Zamezení otevírání odkazů mimo standalone režim

S předchozí zmiňovanou funkcí skrytí nativních ovládacích prvků prohlížeče, čímž je webová aplikace dostána do tzv. standalone režimu, souvisí zamezení otevírání vnitřních odkazů celého webu v novém okně prohlížeče. To samotný tag totiž nevyřeší a Apple dokumentace se tím vůbec nezaobírá, ani to nijak nenaznačuje. Řešení tohoto problému publikoval autor Irae Carvalho na svém GitHub profilu pod volně šiřitelnou MIT a BSD licencí. Celý jeho kód, který je nutné umístit do hlavičky HTML každé stránky, je dostupný z <https://gist.github.com/irae/1042167>.

Výsledek po uložení na plochu

Jsou-li v hlavičce HTML kódu vloženy všechny tagy zmíněné v přechodících odstavcích, celá webová aplikace bude snadno spustitelná z pracovní plochy smartphone zařízení a co je hlavním klíčovým přínosem - bude se se svým vzhledem a chováním velmi blížit nativním mobilním aplikacím, viz Obrázek 9.



Obrázek 9: Web Ordr.cz po uložení na plochu iPhone 6

3.5 Optimalizace touch událostí

Primárním účelem každého responsivního webu je jeho korektní zobrazení na mobilním zařízení a proto je při vývoji nutné si také uvědomit, že tato zařízení jsou v drtivé většině ovládána dotyky prstů. V pokračování za poskytnutím co nejpřístupnějšího UX mobilní webové verze aplikace a rovněž přiblížení její podobnosti chování nativním aplikacím, existují problémy vzniklé na stranách mobilních prohlížečů, kterým je nutné věnovat pozornost.

Hover efekt

Prvním problémem mobilních verzí webů je absence kurzoru myši, na které je závislý selektor `:hover`. Klasický přístup desktopových interakcí, kdy po najetí myší se např. zobrazí nějaká nabídka, může na mobilních přístrojích mít jiná chování. Jonathan Fielding [4 str. 73] zmiňuje, že některé mobilní prohlížeče hover událost převádí na jednoduchý přepínač, takže po prvním kliknutí (resp. tapnutí) se provede daný efekt a po druhém stisknutí elementu se opět vypne. Možným vyřešením problému by podle Fieldinga mohla být detekce dotykových displejů pomocí media queries. Tím se mj. zabývá i Scott Jehl [7 str. 61], podle kterého použití např. `@media (hover) {...}` je blížkou budoucností a tomuto novému parametru nasvědčuje už i jeho výskyt v oficiální dokumentaci W3C s názvem *Media Queries Level 4*³⁷ vydanou v květnu 2015.

Nejsou-li k používání aplikace důležitá složitá dotyková gesta (např. tahy konkrétními směry aj.), není nutné importovat knihovny zajišťující takové chování a ani není nutné tlačítkům přidávat explicitně události na `ontouchstart` nebo `ontouchend`. Lze si vystačit i s jednoduchou `.click()` událostí v jQuery. Je ovšem důležité zajistit, aby uživatel dostal nějakou zpětnou vazbu o tom, že tlačítko, kterého se svým prstem dotýká, je aktivní. Toho lze klasicky na desktopových zařízeních docílit změnou barvy po najetí kurzorem atp. Stejný princip je vhodný i pro dotyková mobilní zařízení, ale je nutné ho upravit. Fielding ve své knize navrhuje nastavit tlačítkům stejný efekt změny barvy pro selektory `:hover`, `:focus` a `:active`. Takový přístup byl použit a testován i v aplikaci Ordr.cz v mobilních prohlížečích Safari a Chrome, ale jeho výsledek nesplňoval očekávání. Po každém tapnutí na dané tlačítko zůstala na

³⁷ <https://drafts.csswg.org/mediaqueries-4/>

elementu změněná barva, i přestože se ho prst již nedotýkal. Přebarvení zmizelo až po dotknutí se jiného elementu. To znamená, že na daném elementu se `:hover` nebo `:focus` selektor vždy zasekl. Je zajímavé, že podobným tématem se nezabývají frameworky Bootstrap ani Foundation. Na mobilním webu Bootstrap se zpětná vazba při tapnutí na tlačítka vůbec nevyskytuje, dá se tedy předpokládat, že je vypnutá. V případě Foundation se již tlačítka obarvují, avšak jejich obarvení zůstává i po opuštění prstu dotykové obrazovky.

Docílení změny barvy pouze při aktivním dotyku prstu daného elementu se zabýval i autor Michael Vartan, který na svém webu představil jednoduchý volně dostupný JavaScriptový kód³⁸, který očekávané chování umožňuje. Kód vymění selektor `:hover` za `:active`.

Časová prodleva události po dotyku na element

Druhým problémem při ovládání mobilního webu na dotykové obrazovce je časová prodleva 300ms, kdy některé prohlížeče čekají, jestli nebude proveden dvojitý klik, nebo jestli uživatel nebude stránku zvětšovat pomocí zoomování. Konkrétní pořadí sledu je:

1. Touchstart
2. Touchend
3. Čekání 300ms pro případ vyskytnutí dalšího dotyku
4. Click

Takové chování nativní aplikace nemají a každá událost, jenž se má po dotknutí daného tlačítka nebo i odkazu provést, se projeví neprodleně. I když se jedná o zdržení pouhé 0,3 sekundy, ve srovnání s nativní aplikací je rozdíl znatelný a pro uživatele se neokamžitá reakce může rozporovat s jejich přirozeným očekáváním. [12]

V článku *300ms tap delay, gone away* napsaném autorem Jakem Archibaldem na serveru Google Developers je představené řešení problému pro prohlížeče Chrome 32 pro Android. Vložením následujícího kódu je časová prodleva odstraněna.

```
<meta name="viewport" content="width=device-width, user-scalable=no">
```

³⁸ <http://mvartan.com/2014/12/20/fixing-sticky-hover-on-mobile-devices/>

V takovém případě je zamezeno možné přiblížení stránky a prodleva je odstraněna i pro prohlížeč Firefox. V článku je dále uvedeno, že tento kód, umístěný v hlavičce HTML dokumentu, neřeší prodlevu v prohlížečích Safari pro iOS. [12] Vzhledem k výsledkům analýz, provedených v následující kapitole a dokazujících značnou návštěvnost právě z mobilních zařízení Apple, je nutné problém vyřešit i právě pro tyto prohlížeče. (Pozn. autora: 1. verze webové aplikace Ordr.cz tento problém neřešila.)

Scott Jehl ve své knize zmiňuje knihovnu FastClick.js³⁹ vytvořenou vývojářským týmem FT Labs pro Financial Times, ale přímo nevysvětluje její benefity. Podobným tématem, nebo přímo rozebráním dané knihovny, se nezabývá žádný z dalších autorů až na Archibalda, zmíněného v předchozích odstavcích, který její použití doporučuje.

Nasazení FastClicku je velmi jednoduché. Stačí ho inicializovat tak, jak mj. uvádí Zdrojový kód 13. Při testování se ovšem vyskytl konflikt zapříčiněný právě jejím aplikováním do webu spolu s použitím dynamicky generovaných elementů, které se nachází v našeptávači u přidávání zákaznickových adres pomocí knihovny Google Autocomplete⁴⁰. Zdrojový kód 13 obsahuje řešení problému, jehož autorem je vývojář s přezdívkou „mzaslavsky“ a byl publikován v návrhu úpravy⁴¹ knihovny FastClick na jejím GitHub profilu.

```
// Init Fastclick.js on Document Body
FastClick.attach(document.body);

// Fix Fastclick.js for GoogleMaps Autocomplete
$(document).on({
  'DOMNodeInserted': function () {
    $('.pac-item, .pac-item span', this).addClass('needslick');
  }
}, '.pac-container');
```

Zdrojový kód 13: Inicializace FastClicku a jeho opravení pro Google Autocomplete

Implementací uvedeného kódu je konfliktu nadále předejito, protože všem vnořeným elementům, které se v našeptávači vyskytují, je přidána třída `.needslick` a ta podle dokumentace knihovny ignoruje zrušení 300ms prodlevy.

³⁹ <https://github.com/ftlabs/fastclick>

⁴⁰ <https://developers.google.com/maps/documentation/javascript/places-autocomplete>

⁴¹ <https://github.com/ftlabs/fastclick/pull/347>

4 Analýza využívanosti

Služba Ordr.cz startovala 1. července 2014 v úzkém centru Prahy 1 a postupně expandovala i do dalších městských částí, viz Tabulka 2. Zkoumané časové období v této kapitole zahrnuje první rok fungování. Tedy období od prvního dne zahájení rozvozu až do 30. června 2015. Uvedená tabulka rovněž umožňuje větší představu nárůstu množství zákazníků reálných nebo i těch potenciálních, kteří službu mohli v období sledování využít.

	Oblast	Měsíc expanze
Testovací období	okolí Václavského náměstí	červenec 2014
První rozšíření	Náměstí Republiky, Florenc	červenec 2014
Druhé rozšíření	Staré Město	srpen 2014
Třetí rozšíření	okolí I.P.Pavlova a Náměstí Míru	srpen 2014
Čtvrté rozšíření	Karlín	srpen 2014
Páté rozšíření	Anděl, Smíchov	říjen 2014
Šesté rozšíření	Invalidovna, oblast kolem Smíchovského nádraží	leden 2015
Sedmé rozšíření	Žižkov, Vinohrady, Albertov, Výtoň	únor 2015

Tabulka 2: Chronologie expanze Ordr.cz do dalších oblastí

Již od svého prvního dne provozu má služba nasazené 3 platformy, na kterých jsou dostupné její aplikace. První platformou je responsivní web, tedy každý zákazník si může službu spustit v prohlížeči buď na svém počítači, nebo mobilním zařízení. Další platformy, mobilní, jsou pro operační systémy iOS a Android nabízené ve svých obchodech s aplikacemi a jsou ke stáhnutí zdarma. Každý uživatel si může aplikaci nainstalovat do svého mobilního zařízení, avšak zákazníkem se stává až ve chvíli, je-li v oblasti rozvozu, protože v jiném případě si jídlo objednat nemůže.

Každá platforma je napojená na Google Analytics, z kterého jsou čerpána data o využívanosti aplikací a vytvářeny následné analýzy. V případě platform iOS a Android jsou ještě navíc využity jednoduché reporty a informace např. o počtu stažení atp. z administračních prostředí jejich obchodů. Zároveň všechny platformy

mají jeden backend, z kterého jsou získávána data o objednávkách z konkrétní platformy. Pro shrnutí jsou tedy hlavní zdroje sběru dat pro analýzu využívání platformy:

- Google Analytics
- iTunes Connect⁴² a Google Play Developer Console⁴³ - administrační prostředí obchodů s aplikacemi
- Vlastní administrační prostředí Ordr.cz

4.1 Webová platforma

V následujících částech této kapitoly bude detailně rozebrána použitelnost webové platformy a přístupy cílového publika do ní mj. i podle technických hledisek jako velikosti displejů, typy zařízení, prohlížeče, operační systémy atp., které je velmi výhodné při vývoji responsivní webové aplikace znát.

Analýza zkoumá 1. verzi této platformy, nikoliv verzi novou, jejíž plánované spuštění je v září 2015, a na které byly implementovány responsivní techniky uvedené v předchozí kapitole. Ty ovšem z velké části vycházely právě z jejího základu a mj. tak umožnily větší soustředěnost na konkrétní problémy při vývoji.

Ve sledovaném období je zaznamenáno celkem 128 871 návštěv webu Ordr.cz. Google Analytics ve svém administračním prostředí vysvětluje, že návštěva je doba, po kterou se uživatel aktivně věnuje webovým stránkám, aplikaci atd. Je důležité neplést zobrazení stránek nebo unikátní zobrazení s touto metrikou, každá z nich má jiné hodnoty:

- Zobrazení stránek: 333 210
- Unikátní zobrazení stránek: 245 888

Ve všech statistikách provedených při analýze webové platformy je pracováno s návštěvami.

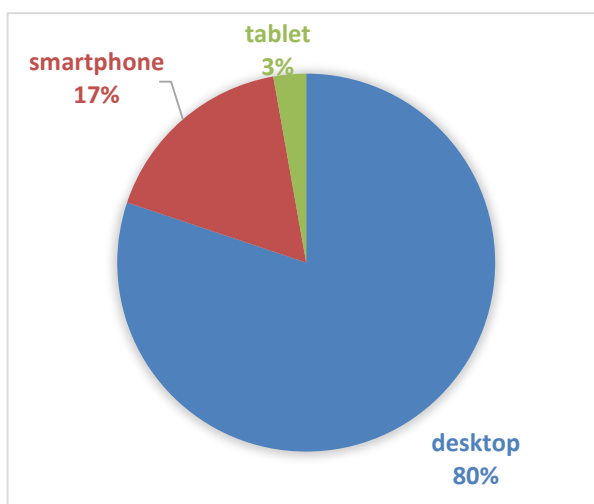
4.1.1 Zařízení

Jako první je nutné zjistit a rozdělit celkový počet návštěv dle platform (kategorií zařízení) - tablet, mobilní telefon (smartphone) a desktop, tedy osobní počítač. Graf 1 ukazuje, že přibližně 103 tisíc, tedy 82 %, uživatelů navštěvuje aplikaci ze svých

⁴² <https://itunesconnect.apple.com>

⁴³ <https://play.google.com/apps/publish/>.

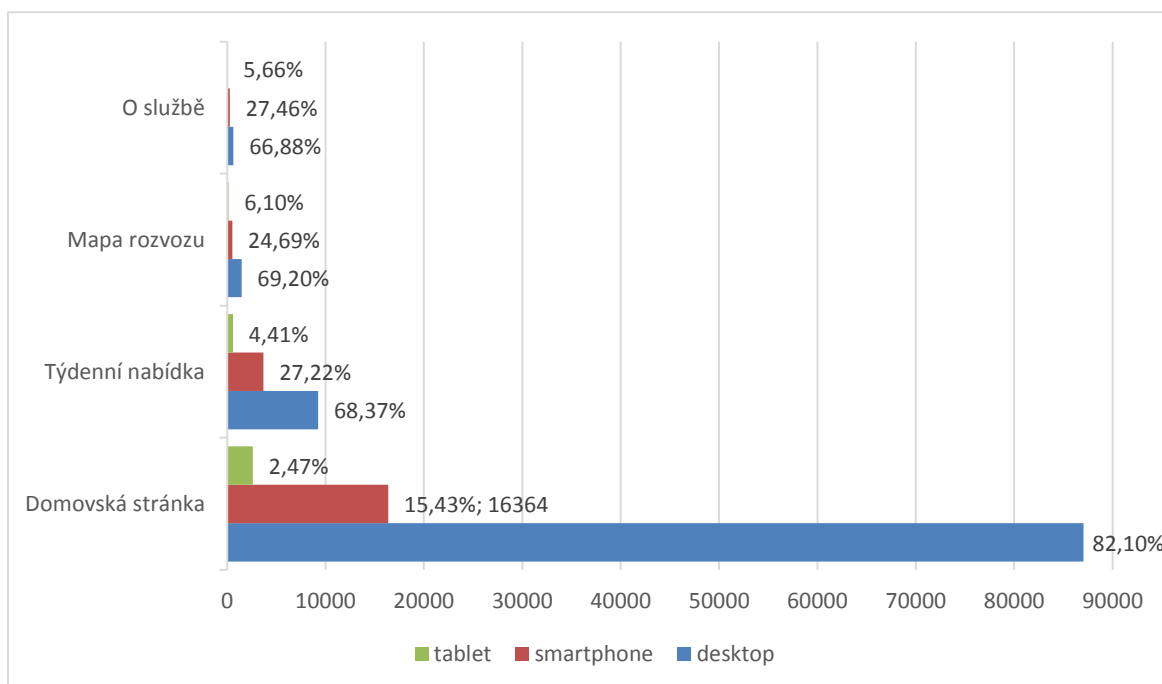
desktopů, což lze do určité míry předpokládat vzhledem k umístění služby v centru Prahy, kde je mnoho kancelářských zástaveb. 17 % tvoří návštěvy z mobilních zařízení. To je rovno přibližně 22 tisícům přístupů, proto je nasnadě se při vývoji podporou designu stránek pro takto malá zařízení zabývat. Minoritu ve zkoumaném rozdělení tvoří návštěvy ze zařízení označených jako tablet, jsou to 3 %, přibližně 3,5 tisíc návštěv.



Graf 1: Návštěvnost webu dle kategorie zařízení

Návštěvnost hlavních stránek

Návštěvnost čtyř hlavních stránek – domovská stránka s objednávkou, týdenní nabídka, mapa rozvozu a prezentační stránka o tom, jak služba funguje - podle kategorií zařízení ukazuje Graf 2.



Graf 2: Návštěvnost konkrétních stránek webu dle kategorie zařízení

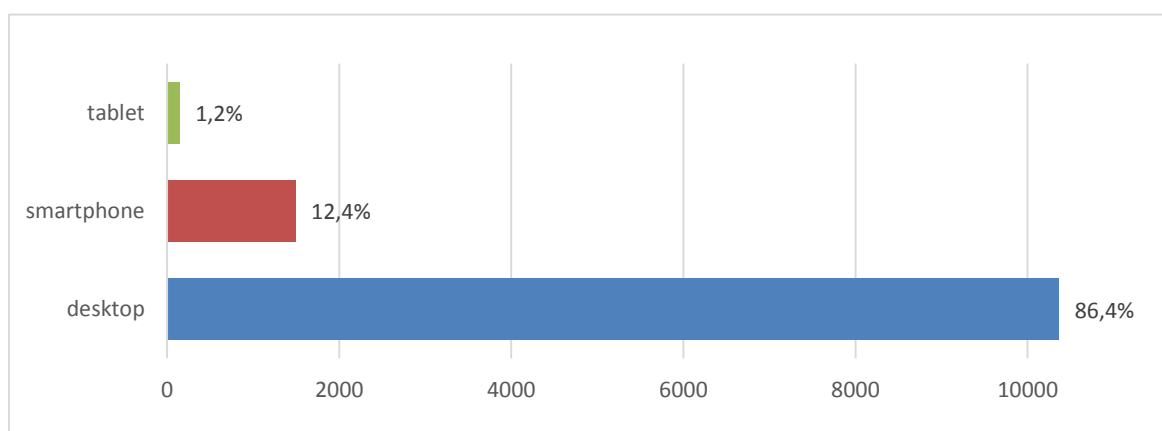
Statistika vypovídá o tom, že nejvíce uživatelů využívalo domovskou stránku, což je předvídatelné, protože je hlavní vstupní branou do celého webu a rovněž proto, že

přímo na ní se odehrával celý objednávkový proces. Je ovšem důležité brát v úvahu i fakt míry okamžitého opuštění s ohledem na doby rozvážení. Z grafu lze určit dominantu desktopu náležící této stránce, což se projeví i v dalších analýzách.

Je překvapivé, že stránka O službě, která obsahuje kroky, jak celý proces od objednání po doručení jídla funguje, má nejmenší návštěvnost. Jistá neznalost služby, kterou lze potvrdit tento trend, se při provozu projevila a působila komplikace, kdy např. zákazníci často nečekali před svým domem na kurýra, což je jednou z podmínek doručení. Proto bylo při návrhu nové verze webu metodou mobile-first rovněž nutné přímo na domovskou stránku umístit tyto informace, aby podobným komplikacím bylo předejito. Porovnání vzhledu domovské stránky verze staré a nové je uvedeno v Příloze 2 a 3 pro desktop a v Příloze 4 pro mobilní velikost.

Množství objednávek podle kategorie zařízení

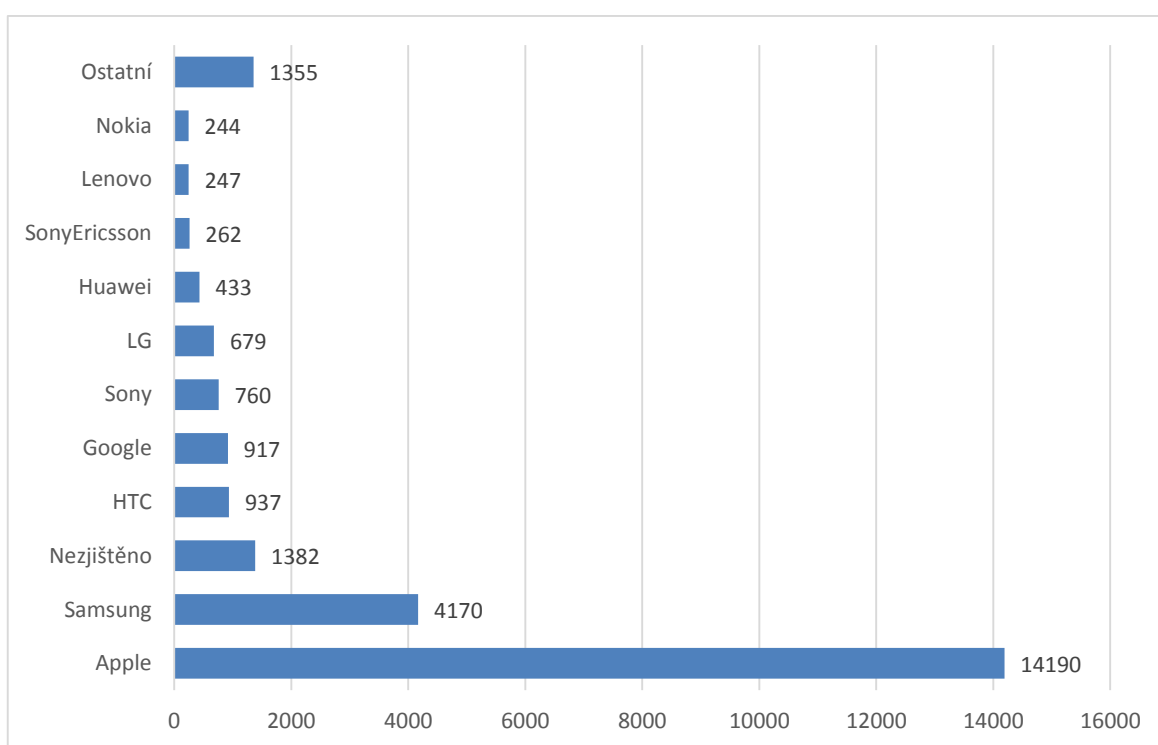
Z webové platformy bylo v průběhu sledovaného období vytvořeno přibližně 12 tisíc objednávek. Graf 3 potvrzuje trend značné převahy desktopu. 86,4 % objednávek bylo vytvořeno právě z této webové verze. Z mobilního webu na smartphonech služba vyřídila 12,4 % objednávek, což představuje přibližně 1,5 tisíců obslužených. Nejméně si uživatelé pro objednávání volí tablet. Tato skupina tvoří pouhých 1,2 %. Sloučením obou mobilních kategorií (tablet a smartphone) do jedné, tvoří mobilní objednávky přibližně 1,6 tisíc a to už lze označit jako závažný důvod, proč mobilním verzím a technikám implementovaných v předchozích kapitolách věnovat při vývoji pozornost.



Graf 3: Počty objednávek z webu dle kategorií zařízení

Mobilní zařízení podle výrobců

Mobilní zařízení (smartphony a tablety) zaznamenala celkem přibližně 25,5 tisíc návštěv. Lze je dále rozdělit podle výrobců viz Graf 4. Nejvíce návštěv z této skupiny přísluší uživatelům vlastníci zařízení od výrobce Apple – tvoří přibližně 60 %. Rozdělíme-li tohoto výrobce dále, zjistíme, že na iPhone připadá 11,5 tisíc návštěv, což tvoří 52,5 % všech návštěv smartphonu zařízení. V případě iPadu je to 2,6 tisíc a to značí 75 % všech návštěv zaznamenaných z platformy tablet. Apple a jeho mobilní zařízení, jak lze ze statistiky usoudit, jsou nejoblíbenější mobilní platformou pro přístup k webové aplikaci.



Graf 4: Návštěvnost webu dle výrobců mobilních zařízení

Na pomyslné druhé příčce v kategorii návštěv dle výrobců mobilních zařízení stojí Samsung, kde v zapříčinění široké modelové nabídky je zaznamenáno hned několik desítek přístrojů. Nejvíce jich však zastupuje model Galaxy S IV se svými 650 návštěvami a Galaxy S4 mini s 500. U celkem 1400 návštěv Google Analytics nezjistil výrobce zařízení. To může být například zapříčiněné zobrazením webu nikoliv v prohlížeči OS přístroje, ale pouze ve WebView aplikace jako je např. Facebook atp. Další dva výrobci s ještě poměrně zasahujícím přínosem do statistiky jsou HTC s modely M7 One, Desire 500 aj. a Google s Nexus 4,5 a 7 aj. Zbývající výrobci jako

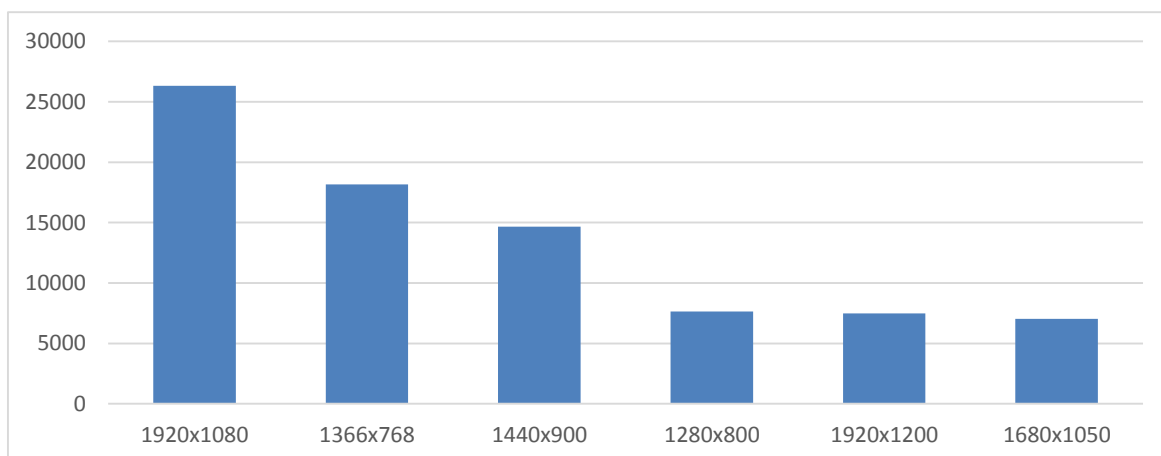
Sony, LG, Huawei atd. tvoří již pouhých 15 %, kde rovněž díky více modelovým řadám je detekováno mnoho přístrojů, které už v porovnání s celou skupinou návštěv z mobilních zařízení tvoří jednoznačnou minoritu.

4.1.2 Velikost displeje

Dalším aspektem, kterým je velmi důležité se při vývoji responsivní webové aplikace zabývat a analyzovat ho, je rozlišení displejů. Těch je pochopitelně veliká řada, proto jsou při analýze rozděleny na velikosti rozlišení desktopových počítačů a mobilních přístrojů a je vždy vybráno prvních šest v každé kategorii.

Rozlišení na desktopových zařízeních

Graf 5 ukazuje celkem 63% návštěvnost dle v něm obsažených velikostí rozlišení desktopových displejů nebo notebooků. Nejčastějším výskytem je velikost 1920x1080px při cca. 26 tisících návštěvách. Následuje rozlišení 1366x768px, které ve většině případů zastupuje 15" velikost notebooků. Bylo detekováno necelých 15 tisíc návštěv z velikosti rozlišení 1440x900px, což jsou již například stolní 17-19" monitory nebo notebook MacBook Air. Ostatní zjištěná rozlišení viz Graf 5.



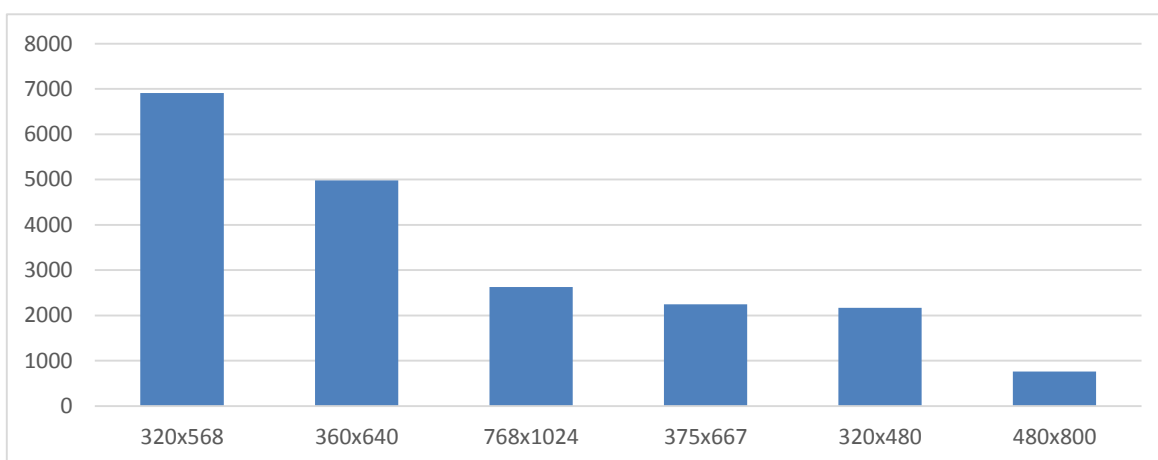
Graf 5: Návštěvnost webu dle rozlišení desktopů

Na základě této analýzy byly v druhé kapitole v tématu zabývajícím se nastavením breakpointů vybrány body 1250px a 1920px provádějící změny velikostí některých elementů pro takto velké zobrazovací plochy. Bod 1250px je aplikován např. větších fontů titulků v úvodních obrazech hlavních stránek a stejně tak i v případě bodu 1920px, kde jsou posléze ještě větší. Při tomto stanovování je důležité brát

v potaz možnosti zmenšení či zvětšení oken (to Google Analytics neměří), protože na velikých displejích větších než např. 1920px je již nepravděpodobné, že uživatelé mají prohlížeče přes celou obrazovku.

Rozlišení na mobilních zařízeních

Mobilní velikosti ukazuje Graf 6, jehož šest rozlišení tvoří 77 % všech mobilních návštěv webu Ordr.cz. Na pomyslném prvním místě při téměř 7 tisících návštěvách je rozlišení 320x568px, jenž patří Apple přístrojům iPhone 5 nebo iPhone 6 (žádné jiné smartphony ho nepoužívají). Z toho tedy lze vypočítat, že z celkem 11,5 tisíce návštěv rozpoznaných ze zařízení iPhone patří cca. 60 % právě těmito dvěma modelům. Zmíněnému rozlišení náleží 30 % skupiny návštěv z přístrojů smartphone. Rozlišení 360x640px má přibližně 22% podíl v segmentu smartphonů a spadá pod něj většina Android zařízení. Třetí nejčastější rozlišení 768x1024px představuje platformy tabletu, např. iPad. Další rozlišení náleží modelu iPhone 6 (rovněž ho má jako jediný model na trhu). 18% zastoupení drží 320x480px, což je velmi často např. iPhone 4. Zbylá rozlišení jsou již minoritní.

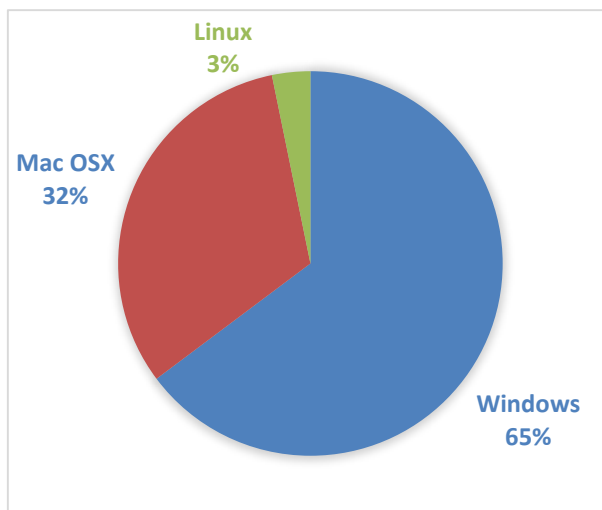


Graf 6: Návštěvnost webu dle mobilních rozlišení

Ve druhé kapitole byla zmíněna krátká analýza provedená Lukem Wroblewskim [3], kde již v roce 2011 byla jako velká první vlna mobilních zařízení označena ta zařízení, jenž měla rozlišení 320x480px. Graf 6 nasvědčuje tomu, že šířka 320px je stále poměrně oblíbená, alespoň při návštěvnosti webu Ordr.cz. Proto je v něm s touto šířkou počítáno jako s tou nejužší.

4.1.3 Operační systém

Při vývoji webové aplikace není operační systém uživatelů v případě desktopů příliš důležitý aspekt. Vzhledem k podpoře prohlížečů napříč platformami Windows, Mac OSX a Linux ovšem někdy vznikají rozdíly v jádrech samotných webových klientů a může se stát, že např. webová aplikace se v prohlížeči Chrome chová jinak v systému Windows než v Mac OSX.



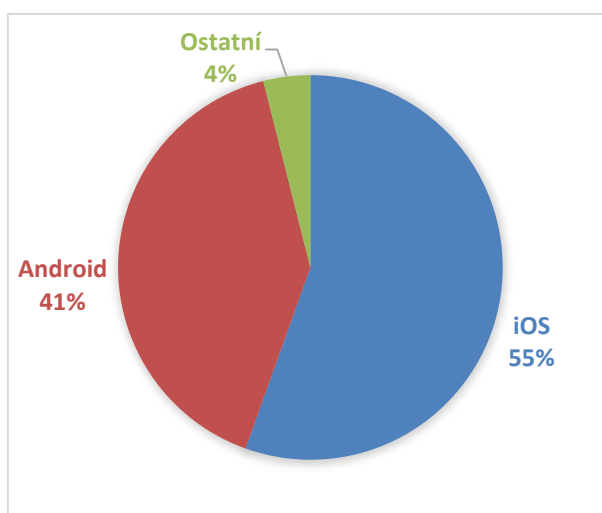
Graf 7: Návštěvnost webu dle OS pro PC

Návštěvnost webu dle operačních systémů pro PC uvádí Graf 7. Jak je z něho zřejmé, většinový podíl z celkových 103 tisíc návštěv drží při 65 % Windows. Na druhém místě je s 32 % operační systém vyskytující se pouze na počítačích Apple. Značnější minoritu, 3,2 tisíc návštěv, tvoří volně dostupný operační systém Linux.

Mobilní operační systémy z pohledu návštěvnosti

V případě operačního systému pro mobilní zařízení je poruchovost a rozdílnost v chování aplikace větší oproti systémům pro PC. Je to zapříčiněné zejména velkým počtem Android zařízení, které se liší svým hardwarem a z toho důvodu vznikají chyby v chování webů, protože systém může mít na každém zařízení jiný výkon.

Jak již analýza a Graf 4 v předešlé části napovídá, nejvíce návštěv bylo v průběhu sledovaného období na mobilních zařízeních se systémem iOS, což potvrzuje i Graf 8. Tuto vedoucí skupinu, z celkových 25,5 tisíc, tvoří



Graf 8: Návštěvnost webu dle mobilních OS

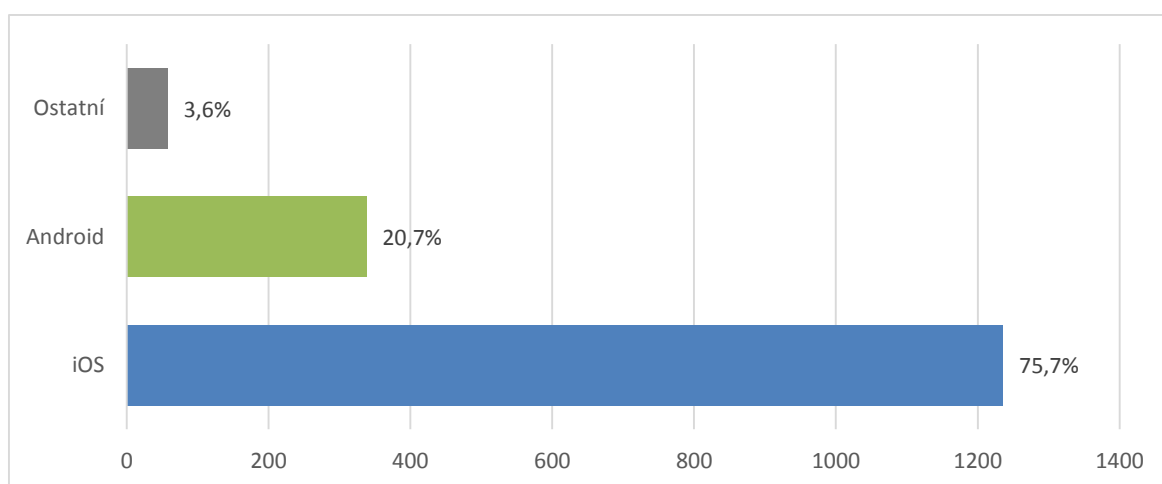
55 %. Z toho 82 % připadá platformě smartphone a 18 % platformě tablet. Nečastějším výskytem tohoto systému byly verze 7 a 8.

Operační systém Android byl detekován ve 41 % návštěv, přičemž 91 % tvoří smartphone a zbylých 8 % tablet. Nejčastějšími verzemi tohoto systému jsou 4+ a 5+. Verze 2.3 a nižší, které často způsobují problémy v kompatibilitě, např. s technikou importováním fontů jako ikon uvedené ve třetí kapitole, se vyskytly v drtivé menšině – přibližně pouhých 100 návštěv, z kterých nebyla provedena žádná objednávka. Scházející 4 % ve sledovaném segmentu tvoří ostatní systémy jako Windows Phone a BlackBerry.

Množství objednávek podle mobilního OS

Aby mohla být provedena komparace využívanosti webu s využívaností nativních mobilních aplikací pro iOS a Android, je nutné zanalyzovat rovněž množství objednávek z webové platformy podle operačních systémů zobrazovacích zařízení.

Analýza dle Graf 9 potvrdila oblíbenost platformy Apple. Z jejího operačního systému bylo registrováno téměř 76 % všech objednávek provedených na mobilním webu, 20,7 % objednávek bylo zaznamenáno z platformy Android a zbývající rozpoznané systémy jako Windows Phone nebo BlackBerry čítá přibližně pouhých 60 objednávek.



Graf 9: Množství objednávek z webu podle mobilních OS

Převaha a oblíbenost platformy iOS pro provádění objednávek dokazuje nutnost soustředit se mj. i na tuto platformu a hledat možnosti optimalizace výkonu aplikace a jejího UX tak, aby byla i pro tyto uživatele co nejvíce přístupná, např. možnostmi

uložení webu na plochu, nebo vyřešení problému s prodlevou po každé touch události, jak je uvedeno ve třetí kapitole.

4.1.4 Prohlížeč

S postupnou evolucí moderních webových technologií vzniká frontendovým vývojářům neustále větší riziko, že jejich aplikace nebudou správně fungovat ve starších webových prohlížečích. Právě veliké množství samotných prohlížečů (především velká škála jejich verzí) je často, dá se říci, noční můrou této komunity. Před spuštěním aplikace široké veřejnosti lze pouze odhadovat využívanost daných prohlížečů její cílové skupiny. Po prvním roce fungování webové aplikace Ordr.cz lze tuto skupinu s ohledem na zmiňovaný aspekt více analyzovat a problematiku podpory se při vývoji zabývat konkrétními problémy daných prohlížečů nebo jejich verzí. Web již od svého spuštění nepodporoval Internet Explorer ve verzích nižších a včetně 9.

Statistika používání prohlížečů celosvětově

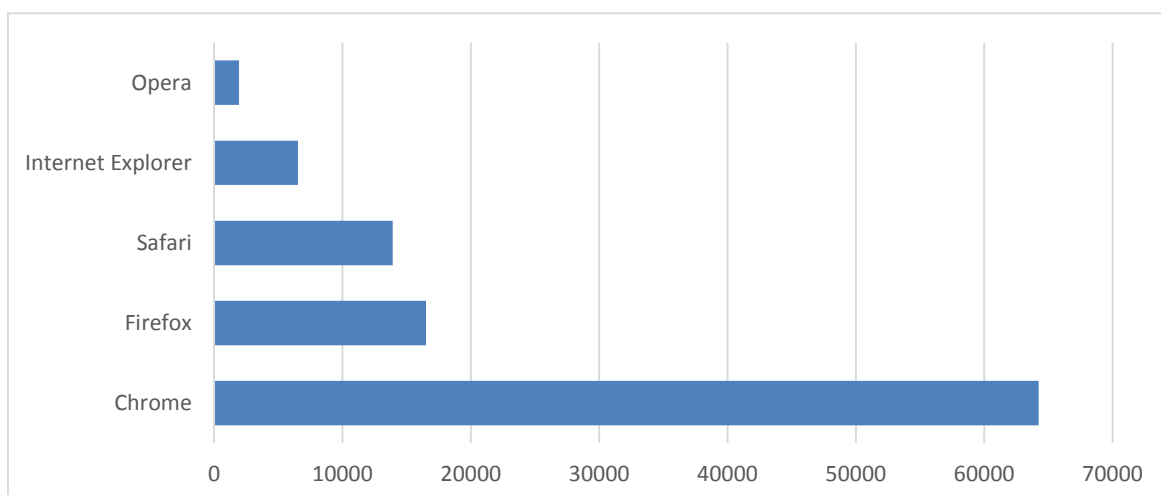
Pro porovnání návštěvnosti webu podle prohlížečů byla z <http://gs.statcounter.com/> provedena krátká analýza o používání prohlížečů ve zkoumaném období v globálním měřítku. Zdroj statistiky sestává ze sítě StatCounter, která obsahuje více než 3 miliony webových stránek a je neustále každé 4 hodiny aktualizován.

Od 1. července 2014 do 30. června 2015 vedl na desktopu dle daného zdroje s 51,2 % prohlížeč Chrome. Druhé místo s 21,5 % obsadil prohlížeč Internet Explorer (s většinovým podílem verze 11). Poté následovaly Firefox s 18,6 %, Safari s 5,2 %, Opera s 1,6 % a ostatní prohlížeče 1,9 %.

Na mobilní platformě statistiku vedl s 29 % nativní systémový prohlížeč Safari. Chrome tvořil 26,4 % celkového použití. Třetím nejpoužívanějším mobilním prohlížečem byl Android Browser s 18,8 %. Následoval ho UC Browser s 9,7 % a Opera s 8,2 %. Zbylé prohlížeče jako BlackBerry, Nokia atp. již představovaly velmi malé podíly.

Desktopové prohlížeče při návštěvnosti webu

Z jakých desktopových prohlížečů byla aplikace Ordr.cz navštěvována ukazuje Graf 10. Stejně jako analýza o používání prohlížečů v globálním měřítku je nejvíce návštěv detekováno z Chrome - přibližně 64 tisíc, což tvoří 62,3 %, všech návštěv z desktopových zařízení. Druhou nejpoužívanější cestou pro zobrazení webu na desktopu je prohlížeč Firefox s 16% podílem. Na pomyslném třetím místě je umístěno Safari s 13,5 %. Až jako čtvrtý zjištěný prohlížeč je Internet Explorer s 6 %. Oproti statistice v globálním měřítku je to značná minorita, tvoří přibližně 6,5 tisíce návštěv. Tento, často problematický, prohlížeč čítá přibližně 4,3 tisíce návštěv z verze 11, z verze 10 je to 1 tisíc, z verze 9 přibližně 8 set a 4 sta návštěv přišlo z verze 8 a nižších. Zbylé jedno procento je zastoupeno prohlížečem Opera.

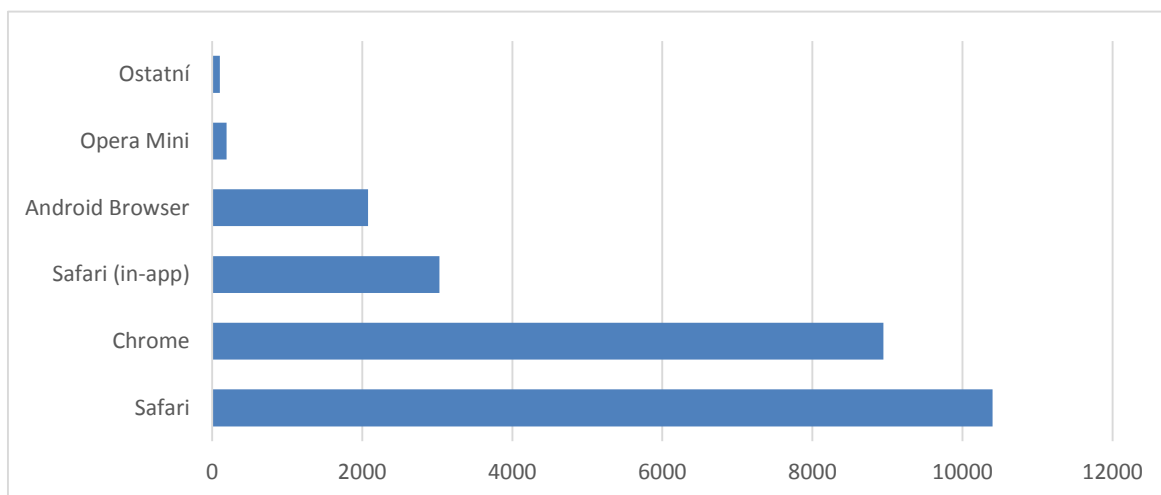


Graf 10: Návštěvnost webu dle desktopových prohlížečů

Mobilní prohlížeče při návštěvnosti webu

Cesty pro zobrazení mobilního webu (smartphone i tablet) Ordr.cz ukazuje Graf 11. Zde je, z celkových cca. 25,5 tisíc, nejpoužívanějším prohlížečem Safari – nativní systémový prohlížeč všech iOS zařízení. Vzhledem k trendu převahy návštěv z mobilních zařízení Apple není tento výsledek nikterak překvapivý. Má 42% podíl. Druhým prohlížečem je mobilní Chrome s 36 %, přibližně 9 tisíci, jenž je v 10 % použit na mobilním zařízení iOS. Zbytek tvoří použití na Androidu. V grafu uvedené Safari (in-app) představuje WebView na zařízeních iOS otevřené v jiné aplikaci než je Safari. To může být situace, kdy je např. uživatel přesměrován v aplikaci Facebook na

web, který se neotevře v Safari, nýbrž v prohlížené aplikaci, nebo i situace, kdy je mobilní web otevírán z pracovní plochy mobilního přístroje Apple, čemuž je věnován prostor v druhé polovině třetí kapitoly. Systémový prohlížeč Androidu tvoří 8 % návštěv. (Pozn. autora: Valná většina návštěv z prohlížečů v této kategorii mobilních zařízení je tvořena smartphony.)



Graf 11: Návštěvnost webu dle mobilních prohlížečů

Analýza stejně jako v případě množství přijatých objednávek z mobilních operačních systémů dokazuje, proč je nutné např. redukovat čekací 300ms prodlevu po každé touch události, kterou řeší rozebíraná knihovna FastClick.js ve třetí kapitole. Implementací knihovny je optimalizováno až přes 12 tisíc mobilních návštěv.

4.2 Platformy iOS a Android

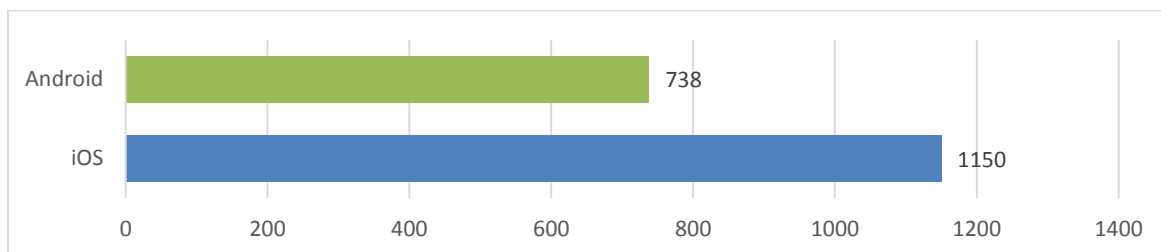
Mobilní aplikace měly v průběhu sledovaného období omezenější nabídku funkcí, než kterou bylo možné využít na webu. Nebylo v nich možné zjistit týdenní nabídku, ani zobrazit mapu rozvozu a neumožňovali platební metodu Twisto⁴⁴. Při vývoji těchto aplikací se kladl vysoký důraz na maximální jednoduchost a minimální počet kroků nutných k provedení objednávky.

Počet stažení mobilních aplikací

Analýza provedená v administračních prostředích obchodů App Store a Google Play ukázala počty stažení, které zobrazuje Graf 12. S celkovými 1150 instalacemi aplikace

⁴⁴ <https://www.twisto.cz>

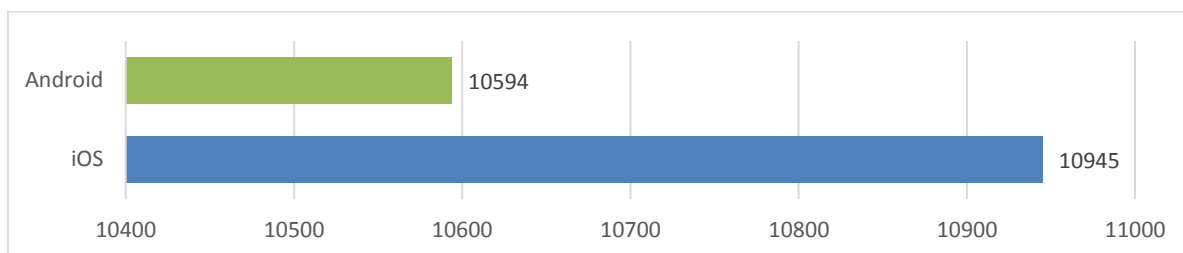
Ordr.cz vede platforma iOS před její sesterskou aplikací pro operační systém Android. Pro toto prostředí si aplikaci stáhlo 738 uživatelů.



Graf 12: Počty stažení mobilních aplikací

Návštěvnost mobilních aplikací

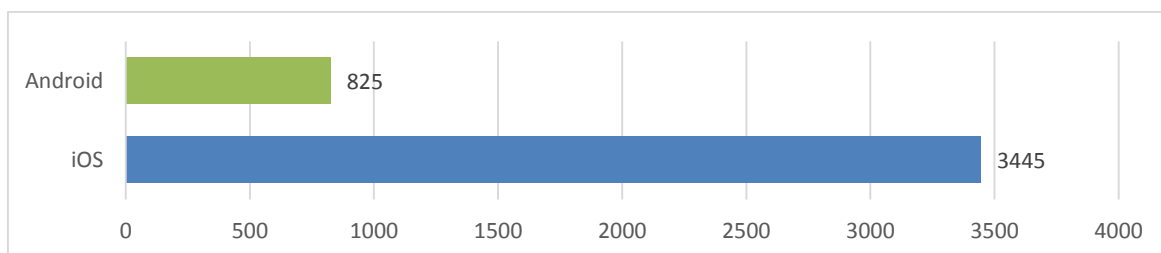
Analýza ukázala, že ve sledovaném období se vyskytlo přibližně dohromady 22 tisíc návštěv v obou mobilních aplikacích. Výsledek porovnání, viz Graf 13, ukazuje téměř rovný poměr mezi systémy Android a iOS.



Graf 13: Počty návštěv mobilních aplikací dle platform iOS a Android

Počet objednávek z mobilních aplikací

Jak se ukázalo při analýzách webové platformy, i v případě mobilních aplikací přicházelo podstatně více objednávek ze systému iOS. Graf 14 ukazuje, že z celkové sumy objednávek odeslaných z mobilních aplikací, tento systém představuje 80,7% podíl a 4 krát méně objednávek služba přijímá z Androidu.



Graf 14: Počty objednávek z mobilních aplikací dle platform iOS a Android

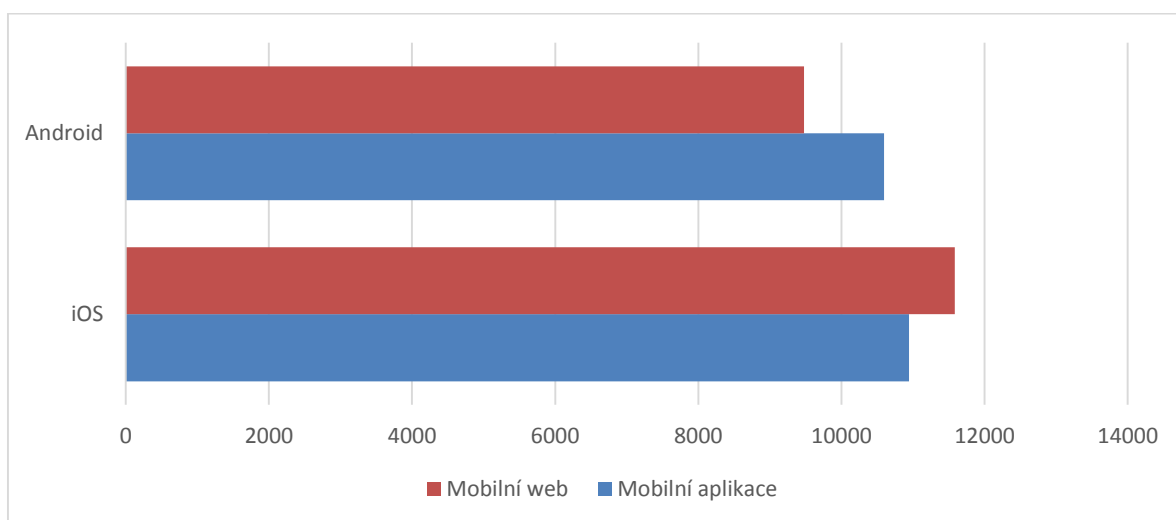
4.3 Komparace všech platforem

Získáním dat provedených na základě předchozích analýz je možné porovnat využívanost aplikací služby napříč všemi jejími platformami. Budou následovat tři komparace podle:

- Mobilní návštěvnosti webu a aplikací
- Počtu objednávek z mobilních aplikací a webu
- Počtu objednávek z mobilních aplikací a mobilního webu

Návštěvnost – mobilní web vs. mobilní aplikace

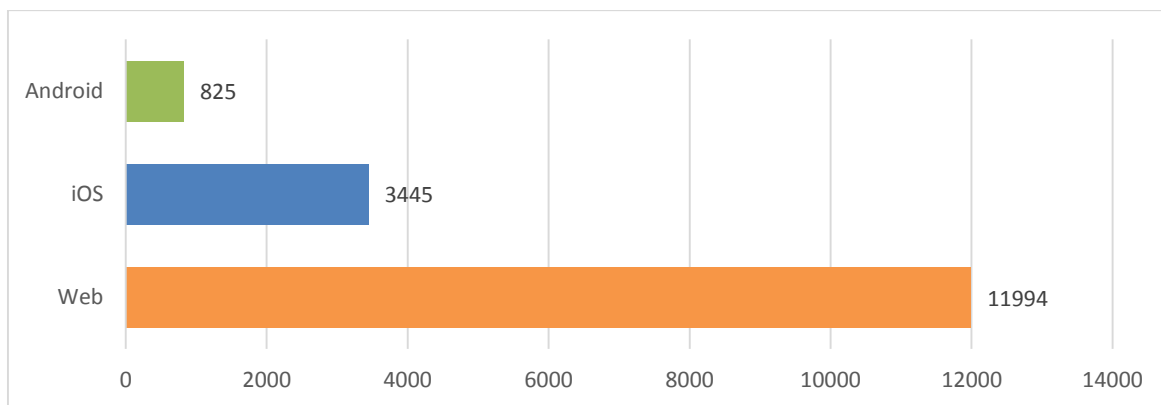
Srovnání návštěvnosti webu z operačních systému iOS a Android s nativními aplikacemi pro tyto dva systémy ukázalo, že v nich není žádný veliký rozdíl. Jak ukazuje Graf 15, mobilní web je pouze o pár stovek navštěvován více ze systému iOS oproti jeho sesterské aplikaci pro daný systém. V případě Androidu je tomu naopak a o něco více uživatelů tohoto systému navštívilo nativní aplikaci, než mobilní web.



Graf 15: Srovnání návštěvnosti mobilního webu a mobilních aplikací

Počet objednávek – web vs. mobilní aplikace

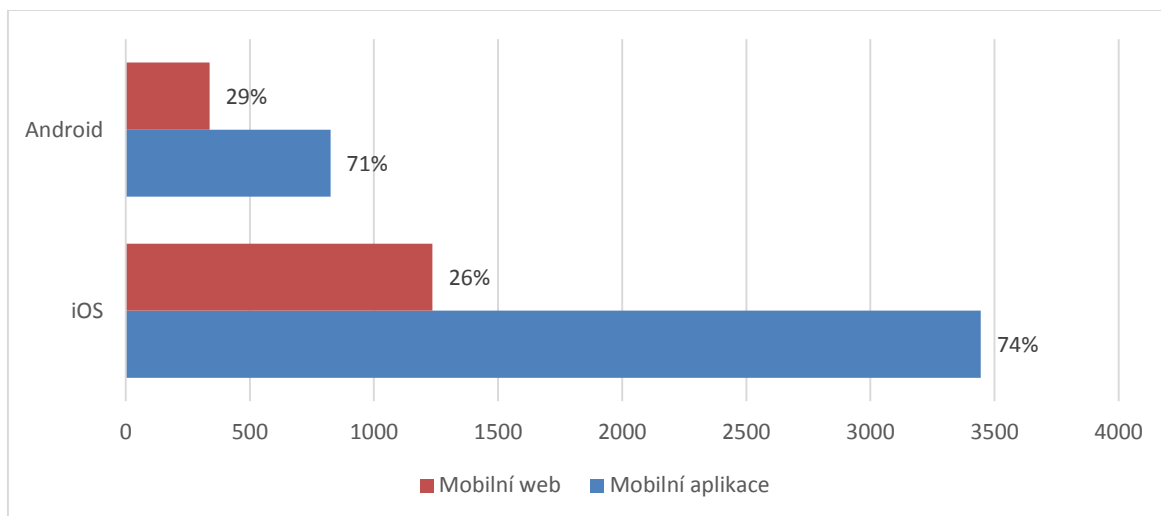
Z celkového počtu objednávek ve sledovaném období, které činí přibližně 16,2 tisíc, přišla valná většina z webové platformy, konkrétně 74 % s 12 tisíci objednávkami. Mobilní objednávky drží zbylý 26% podíl, který představuje přibližně 4,2 tisíce. Z nich bylo přibližně 3,4 tisíce registrováno administracním systémem služby z platformy iOS a dalších zhruba 8 set objednávek z operačního systému Android, viz Graf 16.



Graf 16: Srovnání počtu objednávek z webu a mobilních aplikací

Počet objednávek – mobilní web vs. mobilní aplikace

Posledním srovnáním je množství objednávek provedených pouze z mobilních zařízení se systémy iOS a Android a to v platformě webové a platformách nativních aplikací. Graf 17 ukazuje, že mobilní aplikace jsou v případě samotných objednávek užívanější více než jejich alternativa v podobě mobilního webu.

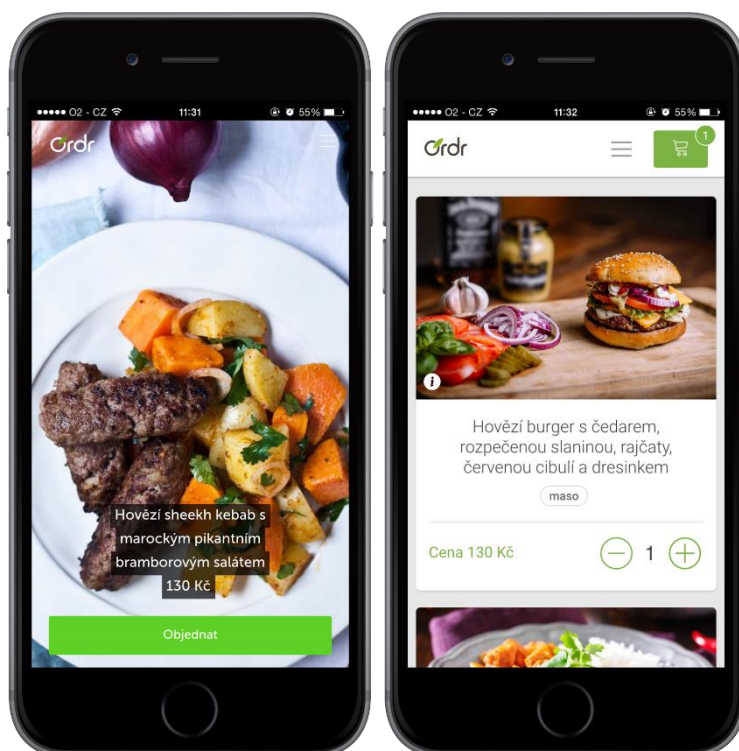


Graf 17: Srovnání počtu objednávek z mobilního webu a mobilních aplikací

Rozdělíme-li si uživatele Androidu na ty, kteří objednávají z mobilního webu, pak tvoří 29 % sledovaného segmentu a zbylých 71 % objednává přes nativní aplikaci. V případě uživatelů systému iOS je poměr velmi podobný, avšak reálná čísla objednávek jsou o řády tisíců větší, jak je vidět na ose x v Graf 17.

5 Shrnutí výsledků

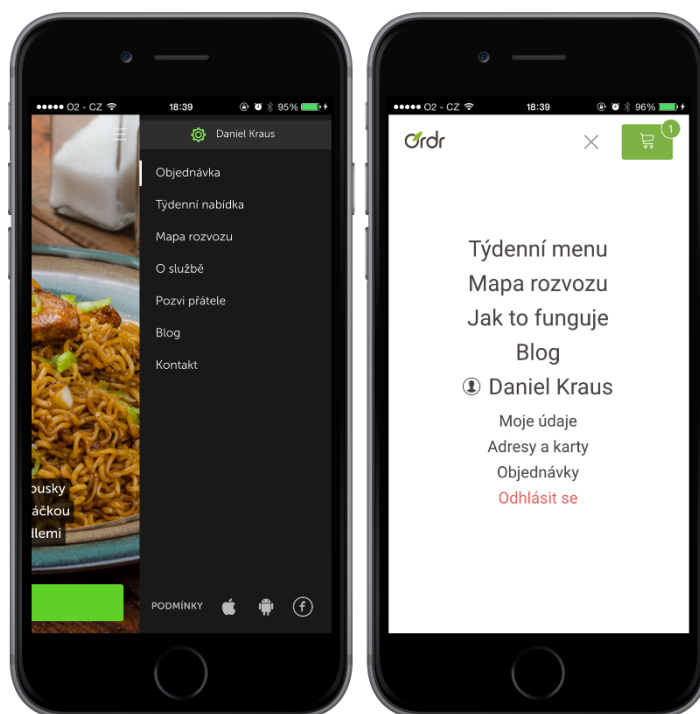
Bakalářská práce se ve své první části zabývala technikami, které bylo nutné vybrat při prvotních návrzích nové verze webové aplikace Ordr.cz. První metodou byl přístup mobile-first. Jak bylo zjištěno, použití metody při grafickém i kódovém návrhu od mobilních verzí napomůže identifikaci právě klíčových vlastností a prvků služby, jenž musejí být umístěny (prezentovány) v omezeném menším zobrazovacím prostoru mobilních displejů tak, aby nedocházelo ke snížení pochopení hlavní podstaty aplikace. Aplikování přístupu na nový web Ordr.cz přineslo soustředěnost na co nejjednodušší proces objednání a rychlé zjištění, jak služba funguje (pro nové zákazníky). Proto nabídka dvou hlavních jídel byla umístěna přímo na první obraz, který uživatel při spuštění webu vidí jak na mobilní, tak desktopové verzi. Pod nabídkou byla umístěná krátká vysvětlení procesu objednávky. Obrázek 10 představuje porovnání staré a nové verze mobilního webu, na kterých lze vidět vylepšení v podobě lepší prezentace obrázků jídel. V předchozí verzi byly obrázky navíc upravovány pro konkrétní velikosti rozlišení displejů, což znamenalo zdlouhavější práci při správě jídel.



Obrázek 10: Porovnání staré a nové mobilní verze webu Ordr.cz

Důležitost soustředěnosti na domovskou stránku rovněž potvrdila analýza využitelnosti, která mj. přinesla informaci o tom, že stránka O službě byla velmi málo navštěvovaná. Docházelo tak k častým nedorozuměním při pochopení objednávkového procesu služby Ordr.cz. Proto byla domovská stránka v nové verzi více propracována a byly na ni všechny důležité informace o službě jako takové umístěny. Kompletní grafické porovnání staré verze a všech částí domovské stránky v nové verzi je umístěno v Příloze 2, 3 a 4.

Při vytváření responsivního layoutu byl na základě frameworků doporučených autory uvedených v literatuře vybrán Grid Systém jedné z nejpoužívanějších frontendových knihoven Bootstrap. Byl zvážěn import celé knihovny, ale protože grafický design webu Ordr.cz obsahuje více specifických prvků (i z důvodu zachování větší unikátnosti), byla importována pouze tato část knihovny umožňující snadnou a rychlou práci se stavbou responsivního layoutu pomocí kontejnerů, řádků a sloupců. Na základě analýzy využitelnosti staré verze webu, kde byly zjištěny nejčastější velikosti rozlišení desktopů i mobilních přístrojů uživatelů, bylo vybráno a definováno 5 breakpointů nutných ke korektnímu responsivnímu chování. Pro snadnější práci při definování media queries byly vytvořeny vlastní proměnné umožňující efektivnější proces psaní stylů, případně rychlou změnu těchto bodů.



Obrázek 11: Porovnání navigací starého a nového mobilního webu

Ve třetí kapitole práce věnovala prostor frontend implementaci dalších technik zajišťujících responsivní chování webové aplikace Ordr.cz. Bylo vytvořeno velmi snadně přístupné navigační menu, které je v porovnání se starší verzí webu lépe ovladatelné i díky zjištění minimálních velikostí rovinných útvarů, které se snadno ovládají dotykem prstů. Výsledek implementace v porovnání se starou verzí webu představuje Obrázek 11.

Důležitým prvkem webu jsou obrázky. Ve třetí kapitole jim byl věnován veliký prostor. Pro práci s obrázky, které se budou přizpůsobivě zobrazovat rozlišením s vysokou hustotou pixelů (např. Retina displeje), byly zkoumány dvě techniky za účelem dosažení zachování správné sémantiky obsahu. V případě obrázků implementovaných v LESS jako pozadí, byl vytvořen vlastní mixin (`.image2x („nameOfImage.png“)`) urychlující implementaci podpory právě takových rozlišení. Pro vkládání obrázků v HTML byl testován nový atribut `srcset` pro známý element `img`. Jak bylo zjištěno, autor Scott Jehl vyřešil problém kompatibility tohoto atributu ve starších prohlížečích vytvořením knihovny `Picturefill.js`, a tak je možné responsivní podporu vyšších rozlišení implementovat i v samotném obsahu HTML dokumentu. Rovněž byly zkoumány a testovány optimalizační metody obrázkových souborů pomocí bezztrátových kompresí datových objemů. Nejvýhodnějším nástrojem provedené testování označilo online nástroj `Kraken.io`. Další optimalizační metoda, která ušetřila na nové stránce Naše jídla se seznamem všech nabízených produktů službou Ordr.cz (v předchozí verzi nebyla) až 4MB datového objemu stránky při jejím načítání ze serveru, se nazývá `LazyLoad`. Jejím přínosem je načítání obrázků jídel až ve chvíli, kdy je uživatelé potřebují, resp. kdy se mají zobrazit ve viditelném prostoru prohlížeče.

Velmi vhodnou metodou pro dosažení podobného chování a vzhledu aplikace běžící na webové platformě aplikacím nativním, což bylo jedním z cílů práce, je implementace meta tagů, které umožňují uložení webu na plochu. Testování ukázalo, že přestože tyto tagy byly vytvořené ve společnosti Apple primárně pro zařízení se systémy iOS, jsou postupně adaptovány i konkurenčním systémem Android.

Při optimalizaci ovládání dotykových mobilních přístrojů byl zkoumán a testován problém časové prodlevy 300ms mezi dotknutím se elementu a vyvoláním patřičné události. V případě prohlížečů Chrome pro systém Android toto chování po

implementaci konkrétního meta tagu není, ale protože analýza prováděná ve 4. kapitole ukázala, že v průběhu prvního roku fungování služby bylo na mobilním webu detekováno až 12 tisíc návštěv z platformy iOS, bylo nutné nalézt řešení i pro tento systém, resp. jeho prohlížeč Safari. Prodleva byla odstraněna implementací knihovny FastClick.js a je poměrně zajímavé, že žádný z frontendových frameworků jako je Bootstrap nebo Foundation toto chování sám neřeší.

Pomocí všech zkoumaných technik se podařilo naplnit jeden z cílů práce, kterým bylo dosažení podobného vzhledu a chování webové aplikace Ordr.cz těm nativním. Dostání cíle ilustruje Obrázek 12. Na prvním přístroji ze tří je mobilní web, na druhém nativní aplikace pro iOS a na třetím nativní aplikace pro Android.



Obrázek 12: Porovnání mobilního webu, nativní aplikace pro iOS a pro Android

Pro zodpovězení otázky stanové v úvodu, je-li responsivní webová aplikace schopna nahradit ty nativní, byla dále provedena analýza návštěvnosti a počtu objednávek registrovaných ze všech platforem. V případě webu výsledky ukázaly, že ve sledovaném období 1 roku z celkových přibližných 103 tisíc tvoří 20 % mobilní návštěvy (smartphone i tablet) a zbylá většina patří desktopům nebo notebookům.

Nativní mobilní aplikace zaznamenaly přibližně 22 tisíc návštěv s přibližně rovným podílem a to se téměř rovná i návštěvnosti mobilního webu.

Analýza webové platformy rovněž věnovala pozornost množství návštěv z konkrétních zařízení dle výrobců, nebo podle velikostí displejů, operačních mobilních i desktopových systémů a podle prohlížečů, kde mj. byly zjištěny přibližně 1,2 tisíce návštěv z prohlížeče Internet Explorer ve verzích 9 a 8, které nebyly starou verzí webu podporovány. Vzhledem k zjištěnému počtu návštěv by bylo v budoucnu vhodné věnovat pozornost kompatibilitě těchto verzí s webovou aplikací Ordr.cz. Tím se práce nezabývala a nový web v současné chvíli tyto verze rovněž nepodporuje, ale v průběhu zkoumání responsivních technik byla představena některá řešení, která připravují základ pro řešení problému. Např. zmíněná knihovna Respond.js, která umožní rozeznání media queries definic, nebo Picturefill.js zajišťující korektní načtení obrázků.

Z pohledu množství objednávek dominuje platforma webová. Z přibližných 16 tisíců objednávek je 12 tisíc zaznamenáno z webu. Zbylé 4 tisíce patří nativním mobilním aplikacím. V porovnání mobilního webu a mobilních nativních aplikací výsledky ukázaly, že 74 % (3,4 tisíce) objednávek ze systému iOS bylo odesláno z nativní aplikace a zbylých 26 % (1,2 tisíce) z mobilního webu. Podobně tomu bylo v případě systému Android, kde 71% podíl (8 set) drží objednávky z nativní aplikace a zbylých 29 % (3 sta) bylo odesláno z mobilního webu otevřeného na zařízeních s tímto systémem.

6 Závěr

V úvodu bakalářské práce byl vytyčen hlavní cíl, za kterým celá práce směřovala. Tím bylo dosažení co nejpodobnějšího chování mobilní verze responsivního webu nativním aplikacím pro systémy iOS a Android a rovněž korektní responsivní zobrazení co největšímu počtu zařízení dle jejich parametrů. Na základě výsledků zkoumání a testování některých technik autorů, z které práce čerpala, byl cíl naplněn.

V průběhu práce byly techniky přímo implementovány do nové verze webové aplikace Ordr.cz a bylo pozorováno jejich chování. Existuje celá řada technik a metod, které lze při vývoji za dosažením stanoveného cíle použít a které byly autory v jejich knihách předkládány. Bakalářská práce všechny techniky zdaleka neshrnuje, ale tento fakt nasvědčuje atraktivitě zkoumaného tématu a je to další důvod, proč problematiku responsivních webů nadále sledovat. Zejména nové možnosti, jenž přináší HTML5 a nové moduly pro CSS (jenž jsou neustále vyvíjeny a rozšiřovány W3C) značně ulehčují vývoj responsivních webů. Problémem v budoucnu ovšem stále bude podpora starších prohlížečů. Vývojáři budou muset neustále vytvářet knihovny, které budou (v nejlepších případech) zpětnou kompatibilitu zajišťovat. Dá se předpokládat, že současné nejnovější prohlížeče si budou nové trendy a metody rychle adaptovat, ale například staré verze prohlížeče Internet Explorer (8 a 9) bude v horizontu nejbližších let velmi těžké podporovat. Stejně tomu bylo v případě první verze webové aplikace Ordr.cz, kde tyto prohlížeče podporovány už nebyly, protože se v nich vyskytovaly závažné chyby, které během roku nebyly vyřešeny. Analýza provedená v práci ukázala, že během prvního roku fungování služby bylo z těchto dvou prohlížečů zaznamenáno 1,2 tisíce návštěv. To nasvědčuje ne příliš závažnému problému, ale o několik objednávek služba určitě přišla. Nová verze webu v současné chvíli tyto starší verze taktéž nepodporuje, ale příprava pomocí zjištěných knihoven bakalářskou prací již na řešení problému začala a bude vhodné se dané problematice nadále věnovat, je-li v plánu expanze služby do dalších měst.

Dalším zajímavým zjištěným faktem spojeným s optimalizačními technikami dotykových ovládaní webu na mobilních zařízeních je neřešení tohoto problému valnou většinou autorů, nebo i známých frontendových frameworků Bootstrap a Foundation. Je pravděpodobné, že blízká budoucnost žádnou změnu v ovládaní zařízení nepřinese

a dotykový způsob ovládnání bude spolu s rostoucím počtem vyrobených mobilních zařízení vzrůstat na reálném využití. Proto je zde prostor pro hlubší zkoumání a možná řešení tohoto problému.

Vedlejším cílem bakalářské práce bylo zodpovědět položenou otázku o možné nahraditelnosti nativních aplikací pro mobilní operační systémy webovou platformou. Na to lze pohlížet dvojím způsobem. Z technického hlediska je možné mobilní web velmi přiblížit chování nativní aplikace na základě uplatněných technik v této práci. Je důležité si uvědomit, že se jedná stále o responsivní web. Existuje řada přístupů, které umožňují vyvíjet nativní aplikace i na webových technologiích. Jedním ze současných řešení, kterému je předpovídána zajímavá budoucnost, je multiplatformní knihovna React⁴⁵, vyvíjená společností Facebook. Nasazení knihovny na platformy aplikací Ordr.cz může být dalším cílem, jemuž lze v budoucnosti věnovat pozornost. Druhým hlediskem, kterým lze k otázce přistupovat, je zjištění míry nutnosti vývoje nativních aplikací umožňujících nabízení produktu či služby a věnování tomuto vývoji další finanční prostředky začínající společnosti jako je Ordr.cz. Provedená analýza ukázala, že ¼ všech objednávek (přibližně 4,2 tisíce) během prvního roku fungování připadá nativním aplikacím a ze zbylé částí patřící webové platformě bylo registrováno přibližně 13,6 % (1,5 tisíce) objednávek z mobilního webu. Z toho lze usoudit, že během prvního roku by pravděpodobně nebylo až tolik nutné takové prostředky vynakládat a responsivní web by mohl být vhodným substituentem. Rovněž ale nelze na základě zjištěných čísel ani říci, že investice do nativních aplikací je chybná či úplně zbytečná.

⁴⁵ <http://facebook.github.io/react/>

7 Seznam použité literatury

- [1] **Marcotte, Ethan.** *Responsive web design*. New York : A Book Apart, 2011. 978-0-9844425-7-7.
- [2] **Makino, Takaki a Phan, Doantam.** Rolling out the mobile-friendly update. *Webmaster Central Blog*. [Online] Google, Inc., 21. Duben 2015. [Citace: 6. červenec 2015.] <http://googlewebmastercentral.blogspot.cz/2015/04/rolling-out-mobile-friendly-update.html>.
- [3] **Wroblewski, Luke.** *Mobile First*. New York : A Book Apart, 2011. 978-1-937557-02-7.
- [4] **Fielding, Jonathan.** *Beginning Responsive Web Design with HTML5 and CSS3*. New York : Apress, 2014. 978-1-4302-6694-5.
- [5] **Sharkie, Craig a Fisher, Andrew.** *Jump Start Responsive Web Design*. Melbourne : SitePoint Pty. Ltd., 2013. 978-0-9873321-6-5.
- [6] **Firdaus, Thoriq.** *Responsive Web Design by Example Beginner's Guide*. Birmingham : Packt Publishing Ltd., 2013. ISBN 978-1-84969-542-8.
- [7] **Jehl, Scott.** *Responsible Responsive Design*. New York : A Book Apart, 2014. 978-1-9375571-6-4.
- [8] **Roberts, Harry.** High-level advice and guidelines for writing sane, manageable, scalable CSS. *CSS Guidelines*. [Online] 2014. [Citace: 16. červenec 2015.] <http://cssguidelin.es>.
- [9] **T, Anthony.** Finger-Friendly Design: Ideal Mobile Touchscreen Target Sizes. *Smashing Magazine*. [Online] 21. únor 2012. [Citace: 14. červenec 2015.] <http://www.smashingmagazine.com/2012/02/finger-friendly-design-ideal-mobile-touchscreen-target-sizes/>.
- [10] **Michálek, Martin.** Lazy Loading. *Vzhůru dolů*. [Online] 6. leden 2014. [Citace: 12. červenec 2015.] <http://www.vzhurudolu.cz/prirucka/lazy-loading>.

[11] **Apple, Inc.** Configuring Web Applications. *iOS Developer Library*. [Online] Apple, Inc., 10. březen 2014. [Citace: 12. červenec 2015.] <https://developer.apple.com/library/ios/documentation/AppleApplications/Reference/SafariWebContent/ConfiguringWebApplications/ConfiguringWebApplications.html>.

[12] **Archibald, Jake.** 300ms tap delay, gone away. *Google Developers*. [Online] Google Inc., 13. prosinec 2013. [Citace: 6. srpen 2015.] <https://developers.google.com/web/updates/2013/12/300ms-tap-delay-gone-away>.

[13] **Refsnes Data.** HTML <i> Tag. *w3schools.com*. [Online] [Citace: 16. červenec 2015.] http://www.w3schools.com/tags/tag_i.asp.

8 Přílohy

Příloha 1: HTML kód navigačního pásu

Příloha 2: Vzhled domovské stránky desktopové verze 1. verze webu

Příloha 3: Vzhled domovské stránky desktopové verze nového webu

Příloha 4: Kompletní vzhled domovské stránky mobilní verze starého vs. nového webu

Příloha 1:
HTML kód navigačního pásu

```
<header>
  <div class="header-bar">
    <a class="header-bar__logo" href="@Url.Action("Index","Home")">
      <span class="d-hidden">Ordr - Nejrychlejší rozvoz</span>
    </a>
  <nav id="header-nav" @if(ViewBag.Homepage==true){ <text> class="" </text> }>
    <ul class="header-nav__items">
      @Html.MenuItem("Týdenní menu", "Menu", "Pages")
      @Html.MenuItem("Mapa rozvozu", "DeliveryMap", "Pages")
      @Html.MenuItem("Jak to funguje", "About", "Pages")
      @Html.MenuItem("Blog", "Index", "Blog")

      @if (Model != null)
      {
        <li id="user-menu">
          <span class="icon-user_male" id="user-
            name">@Model.FullName</span>

          <ul id="user-menu__items">
            <li>
              <a href="@Url.Action("UserProfile","Account")">Moje
                údaje</a>
            </li>
            <li>
              <a
                href="@Url.Action("AddressesAndCards","Account")">Adresy
                a karty</a>
            </li>
            <li>
              <a
                href="@Url.Action("Orders","Account")">Objednávky</a>
            </li>
            <li>
              <a href="@Url.Action("Logout","Account")">Odhlásit
                se</a>
            </li>
          </ul>
        </li>
      }
      else
      {
        @Html.MenuItem("Přihlásit", "SignIn", "Account")
      }
    </ul>
    <div class="header-nav__buttons inline-block">
      <button type="button" id="mobile-menu-btn">
        <span class="lines"></span>
      </button>
      @if (ViewBag.TodayClose != true)
      {
        if (ViewBag.Subpage == true)
        {
          <a href="@Url.Action("Index","Order")" id="header-btn-order"
            class="btn btn-a btn--primary">
            <i class="icon-cart"></i>
            <span class="btn__label">Objednat</span>
          </a>
        }
        else if (ViewBag.Homepage == true)
```



```
        {
            <button type="button" id="header-btn-order"
                class="btn @((ViewBag.UserCount == 0)? "btn--
disabled" : "btn--primary")"
                @((ViewBag.UserCount == 0) ? "disabled" : "")>
                <i class="icon-cart"></i>
                <span class="btn__label">Objednat</span>
                <span id="header-btn-order-count"
class="@((ViewBag.UserCount == 0) ? "" : "active")">@ViewBag.UserCount</span>
                </button>
        }
    }
</div>
</nav>
</div>
</header>
```

Příloha 2:
Vzhled domovské stránky desktopové verze 1. verze webu

Hovězí sheekh kebab s marockým pikantním bramborovým salátem
130 Kč

Objednat >

Grdr
Týdenní nabídka | Mapa rozvozu | O službě | Pozvi přátele | Blog | Kontakt

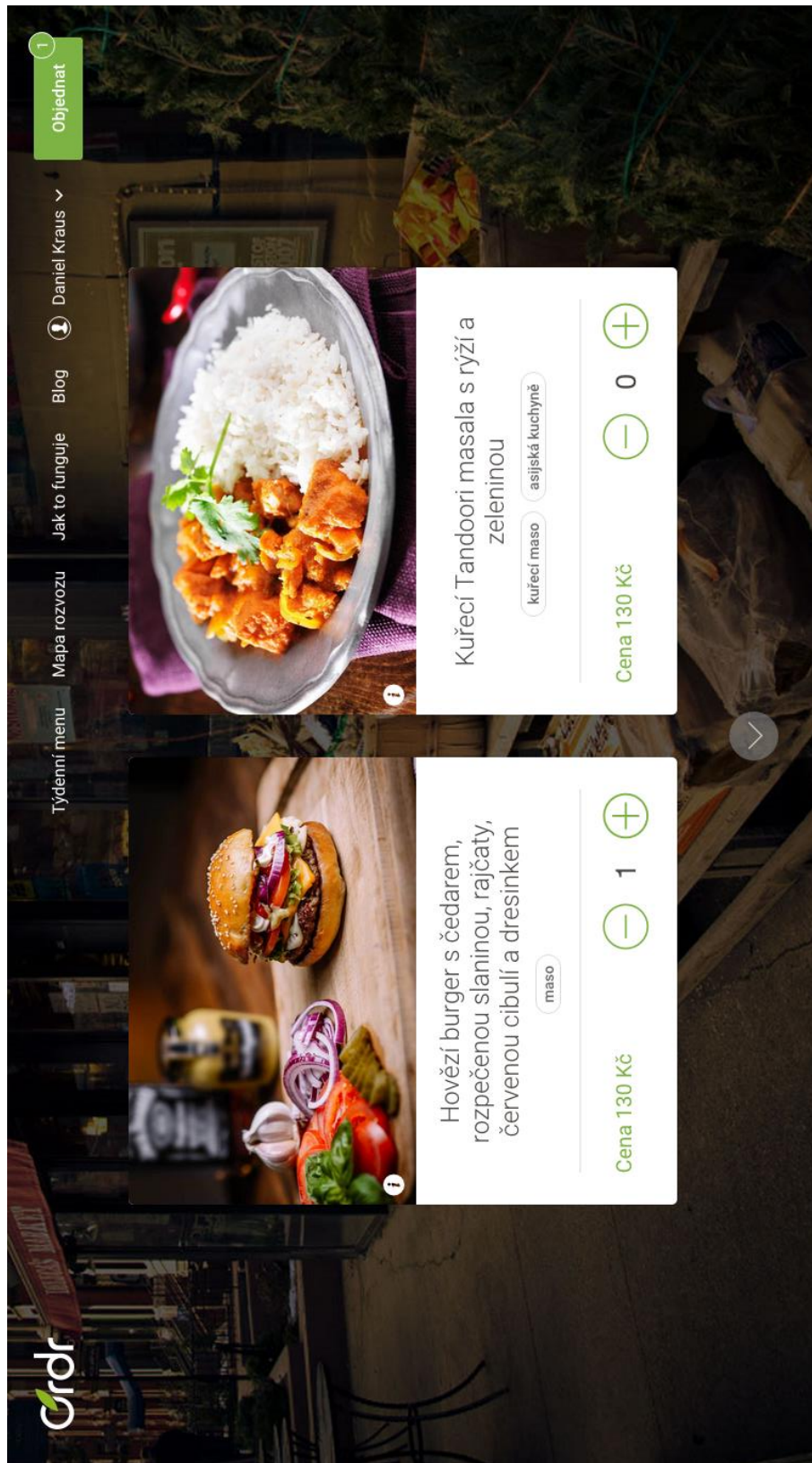
PODMINKY | Online podpora

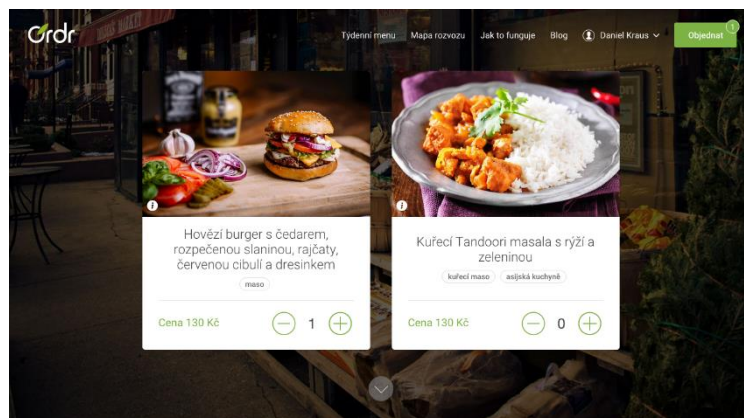
Daniël Kraus

ROZVOZ ZDARMA!

The image shows a desktop view of a website's home page. The main visual is a photograph of a plate of food, specifically a kebab dish with vegetables and potatoes. Overlaid on the bottom left of the image is a dark grey text box containing the name of the dish, 'Hovězí sheekh kebab s marockým pikantním bramborovým salátem', and its price, '130 Kč'. Below this text box is a green button with the text 'Objednat' and a right-pointing chevron. On the right side of the image, there is a dark grey navigation bar with white text and icons. The navigation bar includes 'Týdenní nabídka', 'Mapa rozvozu', 'O službě', 'Pozvi přátele', 'Blog', and 'Kontakt'. Below these items are social media icons for Facebook, Apple, and Android, and a speech bubble icon labeled 'Online podpora'. At the top left of the image, there is a small green logo with the text 'ROZVOZ ZDARMA!' and a profile picture of 'Daniël Kraus'.

Příloha 3:
Vzhled domovské stránky desktopové verze nového webu





Jak Ordr funguje?



Objednejte

Stačí jen pár kliknutí na webu nebo v mobilní aplikaci



Počkejte na SMS

Dostanete zprávu od kurýra, za kolik minut vám vaše jídlo přivezeme



Bězte před dům

Čekajte ve stanovený čas na našeho kurýra před domem

Dodavatelé surovin

Uvědomujeme si, že rychlost je pouze polovina úspěchu, tou druhou je kvalita jídla. Abychom dosahovali co nejvyšší kvality, dodavatele si pečlivě vybíráme. Dbáme na kvalitu a čerstvost potravin, protože jinak se požívá gastronomie dělat nedá.



Ordr pro iOS a Android

Kromě webu si můžete pohodlně objednat i přes mobilní aplikace pro iOS a Android ať jste kdekoliv

- ✓ nejnižší objezdění
- ✓ stačí pár kliknutí
- ✓ upozornění na začátek rozvozu



Buďte s námi na sociálních sítích



"Tak jsem vás dneska večer prvně zkusil a fakt super. Jídlo chutnalo, čas dovozu nějakých šest minut. Ten čas mě zaujal velmi;)! Rozhodně jsem neobjednal naposledy."

Facebook, Pavel Liška



O jídle

Týdenní nabídky
Naše jídla
Dodavatelé a partneři
Pozvat přátele
Velké objednávky

Pro vás

Jak to funguje
Mapa rozvozu
Časté dotazy
Kariéra

O nás

Náš tým
Napiš o nás
Blog
Podmínky užití
Kontakt

Ke stažení

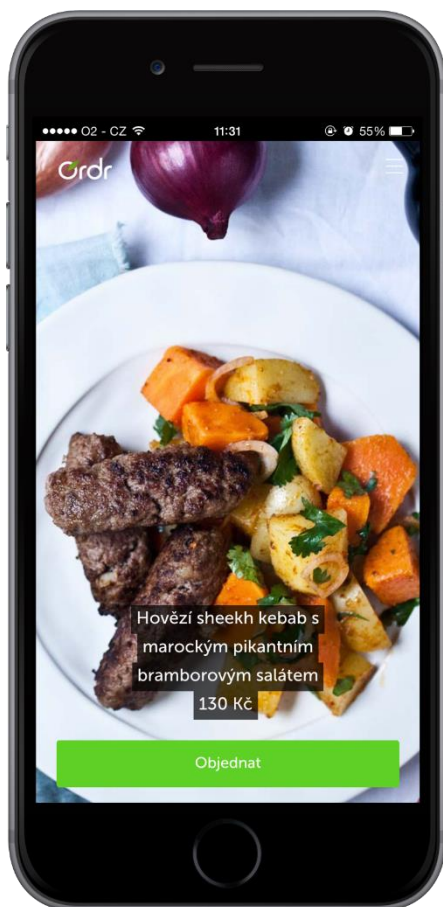
Phone
Android
Media Kit

+420 730 855 296

zakaznik@ordr.cz



Příloha 4:
Kompletní vzhled domovské stránky mobilní verze starého vs. nového webu



Uvědomujeme si, že rychlost je pouze polovina úspěchu, tou druhou je kvalita jídla. Abychom dosahovali co nejvyšší kvality, dodavatele si pečlivě vybíráme. Dbáme na kvalitu a čerstvost potravin, protože jinak se poctivá gastronomie dělat nedá.



Ordr pro iOS a Android

Kromě webu si můžete pohodlně objednat i přes mobilní aplikace pro iOS a Android ať jste kdekoli.



Buďte s námi na sociálních sítích



O jídle

Týdenní nabídka
Naše jídla
Dodavatelé a partneři
Pozvat přátelé
Velké objednávky

Pro vás

Jak to funguje
Mapa rozvozu
Časté dotazy
Kariéra

O nás

Náš tým
Napsali o nás
Blog
Podmínky užití
Kontakt

Ke stažení

iPhone
Android
Media Kit

+420 730 855 296

zakaznik@ordr.cz





Zadání k závěrečné práci

Jméno a příjmení studenta:

Daniel Kraus

Obor studia:

Informační management (3)

Jméno a příjmení vedoucího

Pavel Čech

Název práce:

Vývoj responsivní webové aplikace

Název práce v AJ:

Development of Responsive Web Application

Podtitul práce:

Podtitul práce v AJ:

Cíl práce: Implementace responsivního designu webové aplikace Ordr.cz a provedení analýzy její využitelnosti po dobu 1 roku v komparaci s jejími sesterskými nativními aplikacemi pro platformy iOS a Android.

Osnova práce:

1. Úvod
2. Volba responsivních technik
3. Frontend implementace
4. Analýza využitelnosti
 1. Webová platforma
 2. Mobilní platformy iOS a Android
 3. Komparace
5. Shrnutí výsledků
6. Diskuze a závěry
7. Seznam literatury

Projednáno dne: 29. 10. 2014