

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Bakalářská práce

**Webová Analytika - Tvorba webové aplikace na
správu účtů v Google Analytics**

Marek Aldorf

© 2016 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Marek Aldorf

Informatika

Název práce

Webová Analytika – Tvorba webové aplikace na správu účtů v Google Analytics

Název anglicky

Web App for user management in Google Analytics

Cíle práce

Cílem bakalářské práce je charakterizovat problematiku webové analytiky a vývoje nástrojů pro webovou analytiku. Ten se skládá z následujících dílčích cílů: popis webové analytiky a jejích nástrojů, tvorba aplikace v programovacím jazyce JavaScript, distribuce výsledné aplikace.

Metodika

Metodika řešené problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů, ale také na praktických zkušenostech. Na základě těchto zdrojů a praktických zkušeností bude vytvořena jednoduchá aplikace. Na základě syntézy teoretických a praktických poznatků bude formulován závěr bakalářské práce.

Doporučený rozsah práce

30-40 stran

Klíčová slova

webová analytika, metrika, dimenze, analytika, data, javascript

Doporučené zdroje informací

CLIFTON, B. *Advanced Web metrics with Google Analytics*. 3rd ed. Indianapolis, IN: Wiley. 2012. ISBN 978-1-118-16844-8.

CLIFTON, B. *Google Analytics: podrobný průvodce webovými statistikami*. Vyd. 1. Brno: Computer Press. 2009. 334 s. ISBN 978-80-251-2231-0.

CROLL, M., YOSKOVITZ, B. *Lean Analytics: Use Data to Build a Better Startup Faster*. Boston: O'Reilly. 2013. ISBN 978-1449335670.

KAUSHIK, A. *Webová analytika 2.0 : kompletní průvodce analýzami návštěvnosti*. Brno: Computer Press, 2011. ISBN 978-80-251-2964-7.

Předběžný termín obhajoby

2015/16 LS – PEF

Vedoucí práce

Ing. Čestmír Halbich, CSc.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 2. 11. 2015

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 10. 11. 2015

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 11. 03. 2016

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Webová analytika – Tvorba webové aplikace na správu účtů v Google Analytics" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne _____

Poděkování

Rád bych touto cestou poděkoval vedoucímu své závěrečné práce panu Ing. Čestmíru Halbichovi, za vedení práce, opravy a pomoc při výběru tématu. Dále bych rád poděkoval kolegům z firmy Medio Interactive s.r.o. jmenovitě Bc. Tomáši Kumstovi a Andrému Hellerovi, kteří mi pomohli s úvodem do problematiky vývoje webových aplikací a s teoretickými znalostmi nástroje Google Analytics a Analytics API.

Webová Analytika - Tvorba webové aplikace na správu účtů v Google Analytics

Souhrn

V první části se práce věnuje především poznatkům o webové analytice a tématům pojících se s problematikou webové analytiky, jako jsou segmentace, cookies, metriky, dimenze, KPI a v poslední řadě se teoretická část práce věnuje analytickému nástroj Google Analytics. Tento nástroj je v práci popsán od implementace měřicího kódu, po základní nastavení a strukturu nástroje z hlediska účtu i aplikačního rozhraní.

Praktická část práce je zaměřena na tvorbu webové aplikace na správu účtů v Google Analytics. Tvorba je popsána od grafického a drátěného modelu po spuštění aplikace v prohlížeči.

Přínosem této práce je zejména pro nové uživatele Google Analytics, ušetření času i několika dnů. Jedná se o automatizaci nahrávání filtrů pomocí vytvořené aplikace, ta během pár vteřin zvládne nahrát do vybraného Google Analytics účtu, předpřipravené filtry, které řeší duplicity, SPAM boty a přehlednost dat.

Klíčová slova: analytika, data, metriky, dimenze, Google Analytics, segmentace, typescript, javascript, gulp, cookies

Web Analytics - Development of web applications for managing accounts in Google Analytics

Summary

The first part deals with particular knowledge about web analytics and issues connected with web analytics such as segmentation, cookies, metrics, dimensions, KPIs, and lastly, gives an analytical tool Google Analytics. This bachelor thesis describes implementation of the tracking code to setting a basic structure of the instrument in terms of the account and the application interface.

The practical part is focused on creating a web application to manage accounts in Google Analytics. Creation is described from graphic and wireframe design to launching the application in a browser.

The contribution of this work is especially for new users of Google Analytics, saving time by several days. This is the automation of setting up filters by created application that can handle a few seconds to upload filters to the selected Google Analytics account, preformed filters, which resolves duplicates, SPAM bots and transparency of data.

Keywords: analysis, data, metrics, dimensions, Google Analytics, segmentation, typescript, javascript, gulp, cookies

Obsah

Úvod	1
Cíl práce	2
Metodika	2
1. Teoretická východiska	3
1.1 Rešerše Literatury	3
1.2 Webová analytika.....	4
1.2.1 Historie webové analytiky	4
1.2.2 Click stream analýza	6
1.2.3 Paradox dat.....	7
1.2.4 Ukazatelé webové analytiky	8
1.3 Metriky.....	8
1.3.1 Obecné vlastnosti platné pro všechny metriky	8
1.3.2 Konverzní poměr.....	9
1.4 Dimenze	9
1.5 KPI	10
1.6 Segmentace	11
1.7 Cookies.....	12
1.7.1 First-party.....	13
1.7.2 Third-party	14
1.7.3 Bezpečnost a ochrana soukromí.....	15
1.8 Google Analytics.....	16
1.9 Implementace	16
1.9.1 Implementace pomocí Snippetu Analytics.js.....	17
1.9.2 Ecommerce.....	18
1.9.3 Enhanced Ecommerce.....	20

1.10	Struktura nástroje.....	21
1.10.1	Struktura nástroje z hlediska nastavení a práv	21
1.10.2	Struktura nástroje z hlediska uživatele.....	21
1.11	Možnosti a nastavení Google Analytics	22
1.11.1	Uživatelská oprávnění.....	22
1.11.2	Možnosti Google Analytics a základní nastavení	23
1.11.3	Filtry.....	23
1.11.4	Cíle.....	24
1.11.5	Vyhledávače - Neplacené vyhledávání	25
1.11.6	Sledování vyhledávání na webu tzv. site-search.....	25
2.	Praktická východiska	26
2.1	Drátěný model.....	26
2.2	Tvorba drátového modelu	27
2.3	Tvorba grafického návrhu	31
2.4	Výběr programovacího jazyka	34
2.4.1	JavaScript.....	34
2.4.2	TypeScript.....	35
2.4.3	HTML5 a Less	35
2.5	Výběr nástrojů pro tvorbu aplikace.....	35
2.5.1	Sublime Text 3 a Microsoft Visual Studio Code.....	35
2.6	Tvorba aplikace	36
2.6.1	Html	36
2.6.2	Less a CSS	36
2.6.3	Infrastruktura.....	37
2.6.4	Gulp.....	39
2.6.5	Authorizace do Google API.....	40
2.6.6	FilterManager.....	44

2.6.7	ConnectionManager	47
2.6.8	Testování a spuštění aplikace.....	48
Závěr	49
Seznam obrázků	51
Seznam použitých zdrojů	53
Seznam příloh	55

Úvod

V dnešní době není snad nikdo, kdo by nějakým způsobem nevyužíval internet a hlavně webové prostředí. Na webové stránky chodí lidé po celém světě všech věkových kategorií, aby našli informace a data, které hledají. Bývá pravidlem, že se na středních školách, gymnáziích a většině vysokých škol tvorba webových stránek dokonce vyučuje. Webových stránek je tedy obrovské množství, prakticky ze všech odvětví a ve všech jazycích, a proto každého majitele stránky zajisté zajímá, zda si uživatelé vybírají právě jeho stránku a co a jak tam dělají. Největší zájem o zjištění informací o uživatelích internetu mají samozřejmě tzv. e-shopy, na kterých si uživatelé mohou nakupovat online. Známe spoustu možností, jak se webem pohybovat a co na webu je možné udělat a co není, a aby to nebylo tak snadné, každý z nás se ve webovém prostředí chová trochu jinak. Tak jako neexistují dvě totožné osoby, stejně tak neexistují ani dva stejní uživatelé webu. V tu chvíli přichází ke slovu webová analytika, která se nám tyto nejasnosti v chování lidí na internetu snaží objasnit nebo se k objasnění alespoň přiblížit. Nejdůležitější skupinou na internetu jsou rozhodně e-shopy. Internetové podnikání totiž přináší spoustu výhod, ať už je to neplacení nájmu pro kamennou pobočku nebo nutnost zaměstnání prodavačky či ochranky a další tomu podobné záležitosti, které se týkají fyzického podnikání. Online podnikání za posledních deset let sklízí obrovský úspěch, a to právě kvůli poměrně snadné realizaci. Velká obliba a výhody e-shopů přivádí samozřejmě konkurenci a právě ke schopnosti čelit své konkurenci se využívají metody webové analytiky a optimalizace.

Cíl práce

Cílem bakalářské práce je charakterizovat problematiku webové analytiky a vývoje nástrojů pro webovou analytiku. Ten se skládá z následujících dílčích cílů: popis webové analytiky a jejich nástrojů, tvorba aplikace v programovacím jazyce JavaScript, distribuce výsledné aplikace.

Metodika

Metodika řešené problematiky bakalářské práce je založena na studiu a analýze odborných informačních zdrojů, ale také na praktických zkušenostech. Na základě těchto zdrojů a praktických zkušeností bude vytvořena jednoduchá aplikace. Na základě syntézy teoretických a praktických poznatků bude formulován závěr bakalářské práce.

1. Teoretická východiska

1.1 Rešerše Literatury

Teoretická část této práce vychází z různých zdrojů, týkajících se internetového podnikání a webové analytiky, především Webová Analytika 2.0[18] od Avaniše Kaushika a Web Analytics: An Hour a Day[17]. Konkrétněji se webovou analytikou i digitální marketingem zabývali v češtině ve svých pracích i studenti, například práce z roku 2013 od Jiřího Leinvebera s názvem Nástroj Google Analytics a jeho užití[20], kde se velmi dobře rozebírá problematika titulního nástroje od zdrojového kódu po nastavení a reporty. Avšak tato práce kvůli změnám týkajících se právě tohoto nástroje už není aktuální. Praktická část vychází především z článků a příruček týkajících se API rozhraní od společnosti Google a návodů a článků týkajících se TypeScriptu, HTML5 a CSS3, jazyků, ve kterém je psaná aplikace. V této době není napsáno moc článků a knih o problematice webové analytiky v češtině, protože na území České republiky se tímto oborem zabývá do hloubky pouze pár jedinců, jako je např. Jan Tichý, Roman Apeltauer a Pavel Jašek. Velkou část této práce proto čerpám z přednášek a školení Jana Tichého a z přednášky Pavla Jaška na České zemědělské univerzitě v předmětu web design. Co se týká rozhraní, API základy čerpám z přednášky Romana Apeltauera z odborného setkání Measure Camp Prague.

Pro rozšíření mohu doporučit následující díla, která mohou pomoci v oblastech spojených s touto prací, kterým se přímo nevěnuje. Pro pomoc pro správné nastavení Google Analytics vyšla v roce 2015 kniha Google Analytics Integrations od Daniela Waisberga[31], která se zabývá základními i pokročilými možnostmi nastavení a využití Google Analytics.

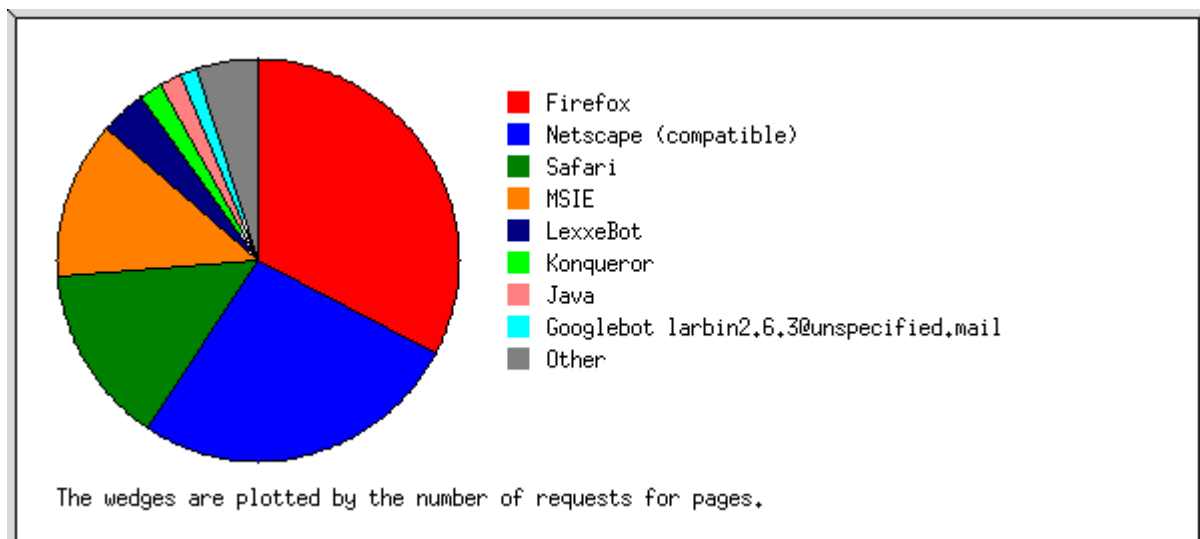
Pro komunikaci s aplikačním rozhraním od firmy Google je nutné využít některý z programovacích jazyků. Zvolil jsem TypeScript, který je nadstavbou JavaScriptu. Pro orientaci v tomto jazyku může pomoci kniha TypeScript Revealed od Dana Maharryho[21].

1.2 Webová analytika

Definice pojmu webová analytika podle Digital Analytics Association (dříve Web Analytics Association): „Webová analytika je měření, sběr, analýza a reportování internetových dat pro účely porozumění a optimalizace webové použitelnosti“ [35]. Pro pohled, jak se vyvíjela v průběhu času, je použita kniha Web Analytics: An Hour a Day od Avinash Kaushika[17] a také kniha Web Analytics 2.0[18] od stejného autora Avinash Kaushika.

1.2.1 Historie webové analytiky

Historie webové analytiky sahá do dob prvních veřejně dostupných WWW stránek na počátku devadesátých let dvacátého století. Tehdy šlo především o analýzu serverových logů. Ty si uchovávaly informaci o tom, že uživatel vstoupil na stránku, ale také o chybách na straně uživatele nebo na straně serveru, jako je například chybně zadaná URL adresa nebo nedostupnost souborů, čas, kdy uživatel vstoupil na stránku, název souboru, IP adresu a také typ prohlížeče. Okolo roku 1993 se začali sledovat tzv. hity, které dohromady dávali návštěvu; podobně je tomu i dnes. S rozvojem internetu se tyto logy stávají z informací pro správce serveru, informací i pro marketingová oddělení, a proto vzniká potřeba tyto data upravit a sloučit do podoby, která by byla čitelná nejen správcům serveru. První veřejností používanou aplikací určenou k analýze logu byl Analog vytvořený roku 1995 Stephenem Turnerem. [17][19][32]



Obrázek 1- Analog - Koláčový graf prohlížečů [32]

Obsahoval základní metriky, jako počty úspěšných a neúspěšných žádostí (requestů), počet různých uživatelů podle IP adresy či velikost odeslaných dat. Kromě toho prezentoval tato data pomocí jednoduchých grafů.

Přišly další nástroje na podobném základu jako WebTrends, které už nebyly volně dostupné, ale ještě více zlepšily možnosti a prezentaci dat. Díky těmto nástrojům se zlepšila manipulace s daty, ale stále zde byl problém s technologickou částí tohoto získávání dat pro webovou analýzu. Hlavním problémem je totiž tzv. cachování (využívání vyrovnávací paměti v prohlížeči). Do vyrovnávací paměti si prohlížeč může ukládat načtenou webovou stránku a její části. Při další návštěvě uživatele se obsah stránky načte z vyrovnávací paměti a tím pádem se znovu nemusí stahovat a nezatěžuje se při této metodě síť tak, jak by tomu bylo bez cache. To je samozřejmě problém, protože se tím pádem neodešle serverový log a měření návštěvnosti se tím stává značně nepřesným. [17][18][34]

Okolo roku 2000 se při velké oblíbenosti a nárůstu počtu uživatelů internetu začaly objevovat javascriptové měřicí kódy známé jako snippets. Tímto způsobem měří velká většina nástrojů dodnes. Při používání těchto měřících kódů zaznamenali velkou popularitu počítadla návštěvnosti, která ve své době zažívala opravdu veliký úspěch a která byla prakticky na každé stránce. Také se začalo v té době měřit chování uživatelů na stránce. Metrikou, která se začala



Obrázek 2 - Urchin - Aplikační prostředí [29]

kromě návštěvnosti používat byla Click-density, nebo-li hustota prokliků na stránce. Jedním z historicky důležitých programů, který měřil návštěvnost a chování na stránce pomocí javascriptových snippetů byl Urchin od firmy Urchin Software Corporation. Urchin byl hybridní nástroj, což znamená, že využíval metodu měření serverových logů, kterou mohl kombinovat s měřením pomocí javascriptového snippetu. Díky této technice byly data přesnější, protože měření oběma metodami najednou bylo daleko přesnější. [17][19][29]

V roce dubnu roku 2005 odkoupila Urchin Software Corporation firma Google Inc. a to vedlo ke vzniku dnes nejznámějšího analytického freeware nástroje Google Analytics. Urchin byl po určitou dobu vyvíjen souběžně s Google Analytics, v roce 2012 byl Urchin plně nahrazen a jeho provoz byl ukončen. [17][18][17][25]

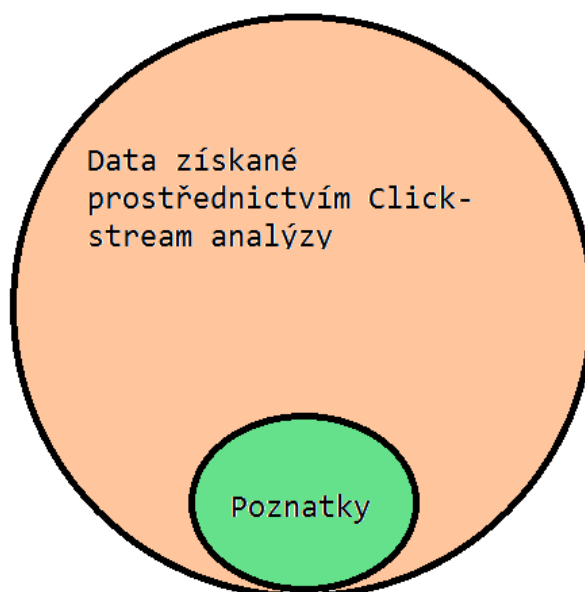
1.2.2 Click stream analýza

Clickstream analýza představuje širokou škálu analytických prostředků, aplikovaných na informace získané provozem obchodních, či jinak zákaznický orientovaných, řešení v

prostředí internetu. K jejich získání dochází, když zákazník přichází na web a prochází se jeho stránkami. Tímto se vytváří spojitý tok kliknutí počítačovou myší, označovaný pojmem clickstream. Technicky jde v podstatě o stopu interakce, kterou po sobě uživatel nesmazatelně zanechává na daném webu. Těchto dat je následně použito při analýze, která má za cíl pochopit chování zákazníka a těchto poznatků pak využít k prospěchu online komerčních aktivit.[18]

1.2.3 Paradox dat

Paradox dat je výraz, který má pro webovou analytiku nesmírný význam a je důležité na tento paradox dbát a brát v potaz vždy při vytváření reportů a analýz ze zkoumaného webu. Jedná se o to, že z webu máme možnost získat obrovské množství dat, ale jen velice málo z nich lze využít k vyhodnocování nebo vylepšování webu. V různých nástrojích pro webovou analytiku máme možnosti prohlížet spousty přednastavených reportů a také možnosti vytváření vlastních reportů z vlastních ukazatelů. Právě z tohoto množství reportů a tabulek uživatelé analytických nástrojů disponují s velkým počtem dat, ze kterých si nezkušený uživatel odnese minimální množství poznatků. [18]



Obrázek 3 - Paradox dat [Vlastní archiv autora]

1.2.4 Ukazatelé webové analytiky

Webová analytika vznikla za účelem shromažďování dat a informací o návštěvnicích a o tom, co se děje na stránce. Právě tato data a informace zahrnují dvě podkategorie, kterými jsou metriky a dimenze. [3][7][18][28]

1.3 Metriky

Metriky obsahují informace kvantitativního měření. Jsou to údaje, které se dopočítávají z fakt získaných měřicím kódem. Fakta jsou např. číselné údaje. [7][17][28]

1.3.1 Obecné vlastnosti platné pro všechny metriky

Pro každou firmu jsou webové metriky naprosto jedinečné. Každá dobrá metrika by měla mít tyto vlastnosti:

1. **jednoduchost** – metrika musí být pochopitelná všem zúčastněným,
2. **relevantnost** – musí mít vypovídající schopnost, tzv. musí být použitelná pro měření úspěšnosti při dosahování cílů,
3. **aktuálnost** – souvisí se sběrem dat a časovým úsekem nutným pro kvalitní vyhodnocení da tak, aby výsledné podklady pro rozhodování byly poskytnuty včas,
4. **okamžitá užitečnost** – metrika musí být pochopitelná tak, aby její výsledky mohly být okamžitě použity pro kvalitní rozhodování a vedly k novým poznatkům.

Využívané metriky je nutné pravidelně ověřovat minimálně z hlediska vlastností dobré metriky. Webová analýza neznamená rutinní provádění stejných činností. Smyslem analýzy je neustálé vylepšování. Proto je třeba metriky analyzovat, vylepšovat, testovat a ověřovat.[18]

S velkou a důležitou skupinou metrik se pojí pojem konverze. Konverze se může vykládat více způsoby. Ve webové analytice to znamená moment, kdy návštěvník provede nějakou akci. Nejčastější konverzí a nepoužívanější je úspěšné splnění objednávkového procesu, tedy moment, kdy se z návštěvníka webu stává zákazník. Dalšími konverzemi mohou být například odběry newsletterů, rozkliknutí informací u produktů, sledování na sociálních sítích atp. Konverze může být tedy jakákoli událost vyvolaná uživatelem, která nějakým způsobem souvisí s cílem měřených webových stránek nebo aplikace. [7][17][18]

1.3.2 Konverzní poměr

Konverzní poměr se počítá jako počet splněných konverzí dělený počtem návštěv. Tato metrika bývá hlavně v oblasti e-commerce hodně zmiňována a spousta majitelů e-shopů to bere jako hlavní ukazatel úspěšnosti. To je velmi špatný pohled, protože pro zisk společnosti nebo e-shopu není nejdůležitější konverze, ale prodaný produkt a hlavně zisk, který prodaný produkt firmě přinese. To znamená, že firmy s nižším konverzním poměrem, ale větší maržovostí mohou mít daleko větší zisk a celkový úspěch e-shopu. [17][18]

1.4 Dimenze

Dimenze popisují určité charakteristiky uživatelů, jejich návštěv a akcí. Dimenze *Město* určitým způsobem charakterizuje návštěvy a udává město (např. Praha nebo Brno), odkud jednotlivé návštěvy pocházejí. Dimenze *Stránka* charakterizuje akce zobrazení stránek a uvádí adresy URL jednotlivých stránek, které byly zobrazeny.

Mezi důležité dimenze, bez kterých by bylo vyhodnocování veškeré webové analytiky prakticky nemožné, patří: [7]

1. Zdroj / Source

- a. Tzv. původce neboli zdroj nabývá názvů stránek, ze kterých se uživatel na měřený web dostal.

2. Medium

Dimenze Medium nabývá hodnot způsobu, který uživatel zvolil pro návštěvu na měřený web. Může nabývat čtyřech různých hodnot, podle toho jaký typ návštěvy webu návštěvník zvolí například: [6]

- a. Přímá návštěvnost / Direct – vzniká zadáním URL adresy stránky do příkazové řádky prohlížeče. [7]
- b. Odkaz / Refferal – vzniká z pro kliků přes odkaz z jiného webu, který není vyhledávačem. [7]
- c. Neplacené vyhledávání / organic – vzniká při pro kliků z neplaceného vyhledávání. [7]
- d. Placené vyhledávání / ppc nebo cpc – vzniká pro klikem přes reklamu ve vyhledávání. [7]
- e. Kampaň / Campaign – také vzniká pro klikem skrz reklamu, ale označenou utm parametry. [7]

1.5 KPI

Pro měření efektivity webu se běžně používají metriky, které se nazývají klíčové ukazatele výkonnosti (anglicky *Key Performance Indicators*, dále jen KPI). Někdy se setkáváme s pojmy klíčové ukazatele úspěchu (*Key Succes Indicators*, KSI). Všechny tyto pojmy vyjadřují jedno a totéž, snaží se postihnout ukazatele, které vypovídají o výkonnosti zkoumaného webu. Pro každý web, se mohou tyto klíčové ukazatele lišit. Z pravidla jedním z KPI, bývají pro většinu e-shopů tržby a návštěvnost.

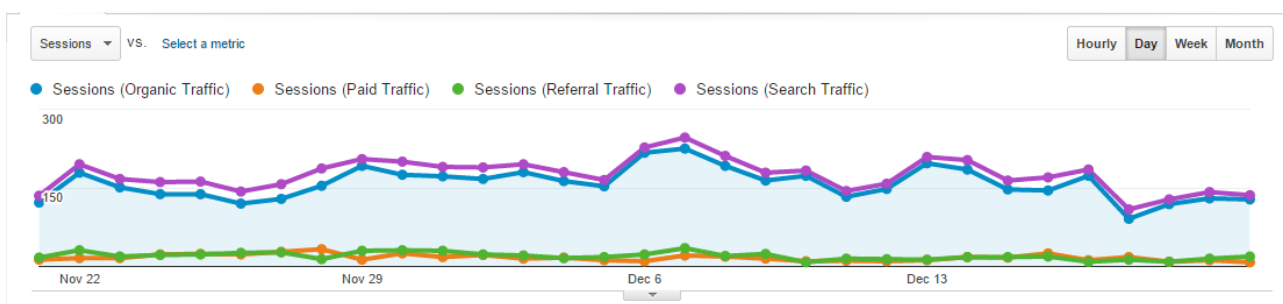
Základním stavebním kamenem pro porozumění úspěchu webu je správné definování KPI. [3][18]

1.6 Segmentace

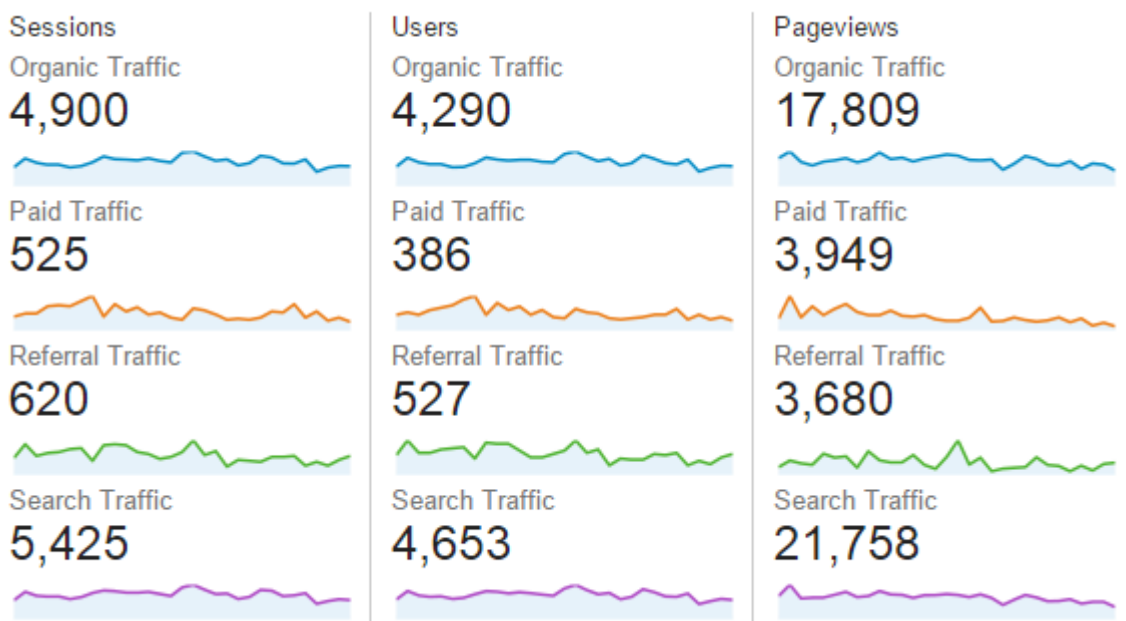
Segmentace je způsob členění uživatelů do menších podskupin, čímž se získá relevantnější pohled na návštěvníky nebo na jejich určitou akci či chování. Je to jeden ze základních metod clickstream analýzy, kde právě bez segmentace prakticky není možnost dostat co nejsprávnější řešení a výstupy. Většina nástrojů webové analytiky segmentaci obsahuje a ty, co jí neobsahují, se používají k jiným typům analýz, než je clickstream. [18]

Ve webové analytice se tedy segmentace používá k odlišení uživatelů, kteří jsou nějakým způsobem jedineční, ať už je to nějaká interakce, kterou na stránce provedou např. přihlášení do newsletteru, dokončení objednávky na e-shopu, registrace a tomu podobné konverze. Dále se uživatelé mohou členit podle jejich chování na webu. Například segmentace vracejících se návštěvníků, nových návštěvníků nebo návštěvníků, kteří nakupují nebo nakoupili na sledovaném webu. Na obrázku je vidět segmentace z nástroje Google Analytics na návštěvnost podle jejich kanálu. Výstup obsahuje segmenty: [17][18]

1. Organické (neplacené) vyhledávání[7]
2. Placené vyhledávání[7]
3. Návštěvnost z odkazujících stránek[7]
4. Návštěvnost z vyhledávačů celkem[7]



Obrázek 4 - Analytics - Segmentace - Časová osa [Vlastní archiv autora]



Obrázek 5 - Analytics - Segmentace – Porovnání [Vlastní archiv autora]

1.7 Cookies

Cookies (česky *sušenky* nebo *koláčky*) jsou malé textové soubory ukládané na straně klienta v prohlížeči (v počítači, smartphonu či tabletu).

Jedná se o jeden z nejpoužívanějších způsobů ukládání dat u návštěvníka webu. Data se odesílají a ukládají při komunikaci se serverem, což z nich dělá výbornou metodu, jak identifikovat návštěvníka. Vzhledem k tomu může mít každý uživatel cookies s různými hodnotami a nastavením. To je jeden z důvodů, proč se v dnešní době bez cookies většina webů neobejde. Pro pohodlnost návštěvníků si totiž ukládají do jejich prohlížečů cookies, které zaručí automatické přihlášení anebo zapamatování kontaktních údajů.

Kromě přihlašování uživatelů se cookies hojně používají pro jejich **sledování**. Cookie tak používají měřicí skripty jako je Google Analytics nebo různé reklamní systémy pro lepší cílení reklam. [5] [6]

Co mohou cookies obsahovat? [5] [6] [12] [24]

1. jméno,

2. hodnotu,
3. doménu,
4. URL cestu / Adresář,
5. expiraci,
6. velikost,
7. ochranu / zabezpečení.

Maximální velikost cookies, kterou podporuje většina prohlížečů, je 4 KB. Maximální počet cookies na prohlížeč se hodně liší, bývá to okolo 300 a víc. Počet cookies na jednu doménu bývá 20. Novější prohlížeče mají limity velkorysejší. Do velikosti se nepočítá jen hodnota cookies, ale všechny její parametry od jména až po zabezpečení.[5] [6] [24]

1.7.1 **First-party**

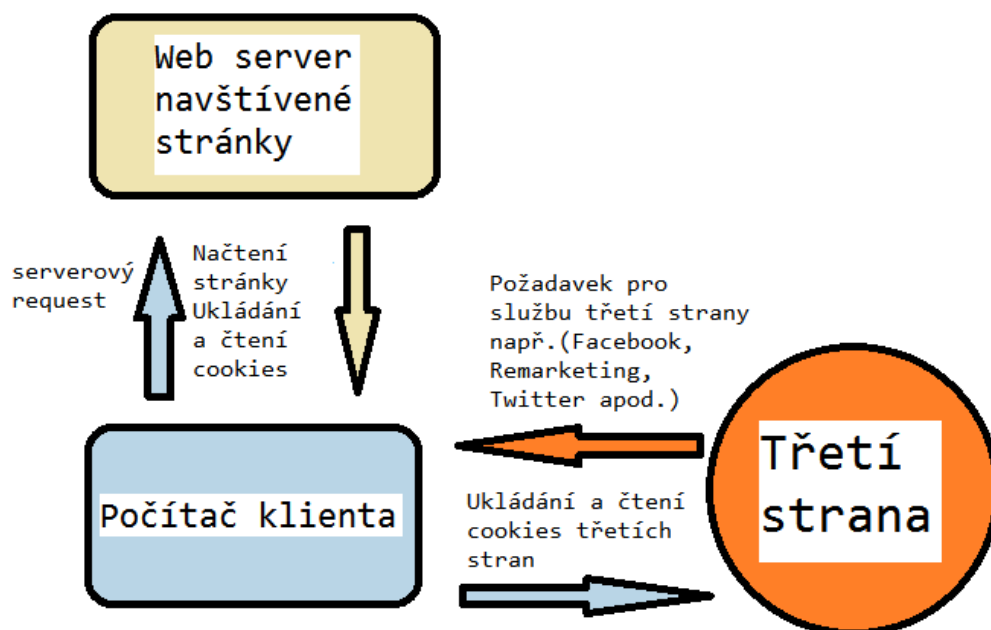
Jsou to ty Cookie soubory, jejichž doména se shoduje s doménou navštívené stránky, neboli s doménou napsanou v Adress baru prohlížeče. Cookie soubory první strany si server navštívené webové stránky uloží u návštěvníka na HDD. Mají k nim přístup jen server navštívené domény a návštěvník. Nikdo jiný je číst, přepisovat anebo ukládat nemůže. Jsou bezpečnější než Cookies třetích stran a využívají se například pro zapamatování přihlašovacích údajů, zapamatování hesla, změnu jazyka nebo pro měření chování uživatelů na stránce. Tento druh cookies využívá např. Google Analytics.[5] [12]



Obrázek 6 - Cookie první strany [Vlastní archiv autora]

1.7.2 Third-party

Jsou to soubory Cookies, které může využívat i další doména. Soubor třetích stran, podobně jako u cookies první strany, si server uchovává u návštěvníka, avšak s tím rozdílem, že tuto cookies může číst a využívat i jiná doména než jen ta navštívená. To vystihuje přesně název třetí strana. Využívají se pro lepší cílení reklamy tzv. remarketing a dynamický remarketing, dále se využívají pro sociální sítě jako je například Facebook, Twitter, Pinterest a Google+. Využívají je prakticky všechny reklamní systémy Adwords, Sklik, RTB House apod. [5] [12]



Obrázek 7 - Cookie třetí strany [Vlastní archiv autora]

1.7.3 Bezpečnost a ochrana soukromí

Cookies samy o sobě žádné nebezpečí nepředstavují, ale servery si do cookies můžou ukládat jakékoli informace, které o uživateli zjistí. Když například uživatel vyplní formulář na odeslání objednávky, tak si server může u uživatele uložit do cookies jeho osobní údaje. Proto se v poslední době o této možnosti narušení soukromí hodně hovoří a v roce 2015 byl schválen Evropskou Unií tzv. “Sušenkový” zákon, který sbírání informací o uživateli pomocí cookies dost omezuje a přikazuje webovým stránkám, které využívají osobní údaje a především cookies třetích stran, že se musí nacházet na stránce tzv. cookie lišta. Kde uživatelé mají možnost se prokliknout na informace o sběru cookies na webu a k čemu ten konkrétní web cookies používá a hlavně možnost souhlasit či nesouhlasit s odesláním cookies. To pro provozovatele webu řídicí se tímto zákonem znamená, že cookies může použít až po souhlasu uživatele s použitím. Do té doby se s cookies hlavně třetích stran pracovat nesmí. Pokud ale používáte cookies první strany a nepoužíváte je pro sbírání osobních údajů, cookie lišta být nemusí. [6][22][24]

1.8 Google Analytics

Google Analytics je nástroj zaměřený na webovou analytiku od společnosti Google, tento nástroj umožňuje vlastníkům webových aplikací a stránek získávat data o uživateli, kteří jejich službu využívají. Díky nástroji Google Analytics je možné sledovat současnou návštěvnost tzv. real-time a historickou návštěvnost, chování uživatelů a jejich události a konverze, prodeje a další. Statistiky dokazují, že přibližně polovina webových stránek využívá k měření právě tento nástroj a tím se Google Analytics stávají nejpoužívanějším nástrojem pro webovou analytiku. Důvodem je kvalita zpracování, jednoduchost a fakt, že pokud uživatel nepožaduje premium verzi, je to zcela zdarma. GA jsou totiž službou typu freemium, to znamená, že základní verze této aplikace je zcela zdarma, ale pro uživatele, kteří požadují lepší funkčnost a prakticky žádná omezení, které se vyskytují u standardní verze, je tu ještě Google Analytics Premium. [17][18][20][16][31][33]

Google Analytics se vytvořil po koupi společnosti Urchin společností Google v roce 2005. V tomtéž roce vyšla už pod záštitou společností Google první verze Google Analytics, která byla zpřístupněna uživatelům na vyžádání, avšak poptávka byla tak velká a počtu přihlášení přibývalo, že se nové přihlašování do této služby zavřelo po týdnu. V roce 2006 se Google rozhodl dodat další přístupy tentokrát formou losování a rozeslal jen určitý počet klíčů vybraným uživatelům přesně tak, aby to vyhovovalo kapacitě serverů. [15][16][31]

1.9 Implementace

Implementace Google Analytics se v praxi provádí více způsoby. Pokud WWW stránka, kterou měříme, běží na redakčním systému Wordpress, Prestashop atp., dají se stáhnout zdarma nebo za poplatek rozšíření, tzv. pluginy, které implementaci a správné měření provedou za vás. Stačí si jen založit na stránce <https://analytics.google.com/> svůj účet a založit službu (v angličtině Property) a pomocí jedinečného UA ID, které systém Google Analytics při založení Property vygeneruje, jej propojit se svým pluginem. [1][19][20]

Další možností je implementovat klasický GA snippet. To je základní implementace doporučená od Googlu. Pokud se jedná o e-shop, který využívá pro systém různé snippety třetích stran, např. Adwords, Google Analytics, Sklik, Facebook konverzní pixel atp, je tento

základní typ implementace časově náročnější a stránka bude mít v sobě obrovské množství Javascriptových snippetů a bude velmi obtížné je všechny spravovat a kontrolovat. Proto je ještě třetí možnost, a to využít správce značek, jako je například Google Tag Manager (dále jen GTM). Tato varianta implementace Google Analytics začíná být velice populární z hlediska jednoduchosti a vlivnosti nástroje GTM, který je také zdarma, jako je tomu u Google Analytics.[1][19][31]

1.9.1 Implementace pomocí Snippetu Analytics.js

Snippet Analytics.js je nový javascriptový měřicí kód pro měření na webu prostřednictvím Google Analytics. Implementuje se hned za počáteční značku body, tak aby se odeslal co nejdříve při načítání webu. [1][19]

```
<!-- Google Analytics -->
<script>
(function(i,s,o,g,r,a,m){i['GoogleAnalyticsObject']=r;i[r]=i[r]||function(){
  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new
  Date();a=s.createElement(o),
  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)
})(window,document,'script','//www.google-
analytics.com/analytics.js','ga');

ga('create', 'UA-XXXXX-Y', 'auto');
ga('send', 'pageview');
</script>
<!-- End Google Analytics -->
```

[1]

Tento kód plní čtyři hlavní funkce:

1. vytváří script, který asynchronně stáhne knihovnu analytics.js JavaScript library, ze stránky <https://www.google-analytics.com/analytics.js>

2. inicializuje hlavní funkci (nazývanou `the ga() command queue`) která umožňuje naplánovat jednotlivé příkazy, které se po úspěšném načtení knihovny provedou,
 3. přidá příkaz do `ga() command queue` pro vytvoření trackeru s parametrem služby(property), do které má být měřeno, tedy založí tracker s `UA-XXXXX-Y`,
 4. přidá další příkaz do `ga() command queue` pro odeslání pageview pro stávající stránku.
- [1]

Vlastní implementací můžeme přidávat změny do měřicího kódu. Změny se provádějí v posledních dvou řádcích javascriptového snippetu. Můžeme přidávat další příkazy. [1]

1.9.2 Ecommerce

Ecommerce je pojem, který zahrnuje jakékoli obchodní transakce provedené na internetu, tedy je to výraz pro obchodování na internetu. Hlavní částí ecommerce je především internetový obchod, tedy e-shopy a s nimi spojená problematika. Patří sem i další činnosti týkající se obchodu na internetu, jako je například online reklama, webová analytika, emaily, affiliate programy, A/B testování a další činnosti spojené s internetovým obchodem. [9][10]

V Google Analytics, zapnutím elektronického obchodu, v nastavení Google Analytics na úrovni Výběru dat nástroji dáte vědět o tom, že web nebo aplikace, na kterém je měřicí kód, obsahuje možnost nákupu. Aktivní ecommerce neboli elektronický obchod zpřístupní několik reportů týkající se transakcí, produktů a prodeje na webu. Pro správnou implementaci ecommerce je potřeba na děkovací stránku přidat snippet pro odeslání transakce. Pro odesílání ecommerce údajů je nutné stáhnout knihovnu ecommerce. Do `ga()` snippetu vložíme tento kód: [9][10]

```
ga('require', 'ecommerce'); [9]
```

Po načtení knihovny se automaticky vytvoří objekt nákupního procesu, kterému můžeme přiřadit transakční údaje, tzn. transakci a produkty. Těmito daty se nákupní objekt naplní pomocí dalších `ga()` funkcí s různými parametry. Příkaz pro vytvoření transakce `ecommerce:addTransaction`. Název příkazu se posílá jako první parametr funkce `ga()`. Jako druhý parametr se posílá obsah příkazu ve formě JSON objektu. JSON objekt musí obsahovat určité povinné prvky a může obsahovat další nepovinné prvky. Vždy je povinné ID produktu

nebo transakce a také proměnná name u produktů. Pro vytvoření a naplnění transakce je to následující kód: [9]

```
    ga('ecommerce:addTransaction', {
'id': '1234', // ID transakce. Povinný údaj.
'affiliation': 'Acme Clothing', // Název obchodu nebo affiliate.
'revenue': '11.99', // Cena transakce bez DPH
'shipping': '5', // Poštovné, Dopravné
'tax': '1.29' // Daň
});
```

[9]

Při úspěšném naplnění transakce je potřeba transakci provázat s jednotlivými produkty, které si zákazník nakoupí.

```
    ga('ecommerce:addItem', {
'id': '1234', // ID transakce. Povinný údaj.
'name': 'Fluffy Pink Bunnies', // Jméno produktu. Required.
'sku': 'DD23444', // SKU/skladovací číslo.
'category': 'Party Toys', // Kategorie produktu.
'price': '11.99', // Cena jednoho produktu bez DPH
'quantity': '1' // Množství.
});
```

[9]

Celý kód bude vypadat ve výsledku následovně:

```
ga('create', 'UA-XXXXX-Y');

ga('require', 'ecommerce');

    ga('ecommerce:addTransaction', {
'id': '1234', // ID transakce. Povinný údaj.
'affiliation': 'Acme Clothing', // Název obchodu nebo affiliate.
'revenue': '11.99', // Cena transakce bez DPH
'shipping': '5', // Poštovné, Dopravné
'tax': '1.29' // Daň
});

    ga('ecommerce:addItem', {
'id': '1234', // ID transakce. Povinný údaj.
'name': 'Fluffy Pink Bunnies', // Jméno produktu. Required.
```

```

'sku': 'DD23444', // SKU/skladovací číslo.
'category': 'Party Toys', // Kategorie produktu.
'price': '11.99', // Cena jednoho produktu bez DPH
'quantity': '1' // Množství.
});

ga('ecommerce:send');

```

[9]

1.9.3 Enhanced Ecommerce

Enhanced Ecommerce umožňuje kromě doposud známého měření transakcí měřit výkonnost celé prodejní cesty, a to od tzv. impresí produktů (zobrazení produktu uživateli), přes jejich klikání, zobrazování detailů až po přidávání a odebrání z košíku, detailní popis objednávkového procesu až po samotnou transakci. V rámci tohoto je dále možné sledovat například výkonnost jednotlivých prodejních seznamů nebo celých kategorií produktů. Implementace pomocí snippetu analytics.js je obdobná jako u základní ecommerce s tím rozdílem, že se vyžaduje knihovna 'ec'. Opět tedy musíme zavolat následující funkci: [11]

```
ga('require', 'ecommerce'); [11]
```

Dále postupujeme jako v předchozí části bakalářské práce, tedy zavoláme kód pro měření impresí produktů a doplňujeme jednotlivé parametry s tím, že opět je nutné správně implementovat hlavně povinné údaje, a to kvůli správnému provázání transakcí a produktů: [11]

```

ga('ec:addImpression', { //Pokud je produktů více, vkládáme
je jako pole do hranatých závorek
['id': 'P12345', // ID produktu (string)Povinný údaj.
'name': 'Android Warhol T-Shirt', // Jméno produktu (string). Povinný
údaj
'category': 'Apparel/T-Shirts', // Kategorie produktu (string).
'brand': 'Google', // Značka produktu (string).
'variant': 'Black', // Varianta produktu (string).
'list': 'Search Results', // Produktový výpis (string).
'position': 1] // Pozice produktu z leva do prava a od
zhora dolů (number).
});[11]

```

Stejným způsobem můžeme měřit přidání nebo odebrání produktu z košíku, kliknutí na produkt atp. [11]

1.10 Struktura nástroje

1.10.1 Struktura nástroje z hlediska nastavení a práv

1. Google účet
 - a. e-mail pro přihlášení,
 - b. s jedním Google účtem je možné mít přístup do více Analytics účtů.
2. Analytics účet
 - a. oficiálně jeden účet pro celou firmu,
 - b. best practice – co jeden web, to jeden samostatný účet.
3. Služba / Property / Web
 - a. odpovídá jednomu měřicímu kódu,
 - b. nejčastěji má tedy každý web svou službu, svůj měřicí kód.
4. Výběr dat / Zobrazení / View / Profil
 - a. jedna úplná sada všech statistik,
 - b. jeden měřicí kód měří zároveň do všech podřízených profilů

Příklad struktury účtů viz Příloha 2 [30]

1.10.2 Struktura nástroje z hlediska uživatele

Co se týká struktury nástroje Google Analytics, ten se dělí na čtyři hlavní záložky:

Obrázek 8 - Struktura nástroje [Vlastní archiv autora]

1. Domovská stránka
 - a. Zobrazení všech Účtů, Služeb a Výběrů dat, ke kterým má uživatel přístup.
2. Přehledy
 - a. Reporty, Panely, Zkratky, Informační události vybraného výběru dat.
3. Přizpůsobení
 - a. Možnost tvorby, úpravy a vytváření vlastních přehledů.
4. Správce
 - a. Zde se provádí veškerá nastavení ke všem úrovním struktury Google Analytics.
[20]

1.11 Možnosti a nastavení Google Analytics

1.11.1 Uživatelská oprávnění

Každá úroveň struktury Google Analytics má vlastní přístupová práva. To platí s výjimkou Google účtu, kam má přístup pouze jeho majitel (nebo by alespoň měl mít). Co se týče přístupových práv pro uživatele, máme 4 úrovně: [30]

1. správa uživatelů (umožňuje přidávat a odebírat uživatele a nastavovat práva),
2. upravit (umožňuje veškeré nastavení a úpravy až na správu uživatelů, zahrnuje oprávnění spolupráce a číst a analyzovat),

3. spolupráce (umožňuje vytvářet veškeré osobní podklady, tzn. custom reporty, segmenty atp., zahrnuje oprávnění číst a analyzovat),
4. číst a analyzovat (umožňuje práci s reporty a filtrací tabule, přidávání sekundárních dimenzí a také vytváření podkladů avšak na podkladech nemůže spolupracovat). [30]

1.11.2 Možnosti Google Analytics a základní nastavení

Pro správné měření webových stránek a aplikací je důležité mít analytický nástroj správně nastavený tak, aby jeho výstupy co nejvíce odpovídaly skutečnosti a nebyly zde duplicity a všechna data měla správný tvar. [17]

1.11.3 Filtry

Filtry jsou pravidla nebo podmínky, které určují, co se s nasbíranými daty, které se pošlou na server Analytics, pro zpracování bude dít. U filtrů je nepsané pravidlo, že se dělí do 5 kategorií. [20]

1. Vylučovací
 - a. Vylučovací filtry slouží k upřesnění skupiny lidí, kterou měříme. Tyto filtry nastavujeme jako *první*, abychom nemuseli zbytečně pracovat s daty, která ve výsledku vůbec nevyužijeme. V praxi se používá vyloučení vlastní IP adresy pro přesnější měření. Vylučují se zde také SPAM boti a Google a Seznam boti, kteří narušují návštěvnost. [20]
2. Nahrazovací
 - a. Nahrazovací filtry slouží jako standardizace názvů. Díky nim nedochází k duplicitám, a vše padá do správných kolonek. Základními nahrazovacími filtry jsou Lowercase. Poté následuje oprava známých rozdílových médií a zdrojů. Tyto filtry nastavujeme jako druhé. Mezi tyto filtry patří např. změny velikosti písmen, nahrazování zdrojů a médií a nastavování nových médií. [20]
3. Pokročilé a Ostatní
 - a. Přídavné a složitější filtry pro sjednocování názvů zdrojů a médií. Mezi tyto filtry patří sjednocování zdrojů a médií například pro facebook, twitter, youtube atp. [20]
4. Přídavné speciální
 - a. Filtry přidávané jen v ojedinělých případech. Například sjednocování jazykových nastavení nebo sjednocení partnerských srovnávačů pod jeden hlavní. [20]

Výše uvedené pořadí má svůj význam, protože GA filtry používají od prvního k poslednímu. Je tedy důležité, aby vylučovací byly jako první a aby všechny ostatní filtry, které se navzájem ovlivňují, byly ve správném pořadí. [20]

1.11.4 Cíle

Pomocí cílů můžete měřit, jak vaše webové stránky nebo aplikace plní plánované záměry. Cíl představuje dokončenou činnost, nazývanou konverze, která je pro vaši firmu přínosná. Příklady cílů: uskutečnění nákupu (u webu s elektronickým obchodem), dokončení jedné úrovně hry (u mobilních herních aplikací) nebo odeslání formuláře s kontaktními údaji (u marketingového webu nebo webu zaměřeného na oslovování potenciálních zákazníků). [4]

Základní součástí každého plánu měření a analýzy dat je definování cíle. Díky správně nakonfigurovaným cílům může služba Google Analytics poskytovat zásadní informace, například počet konverzí a míru konverze pro váš web nebo aplikaci. Bez těchto informací je téměř nemožné vyhodnotit účinnost internetových obchodních a marketingových kampaní. [4]

Cíle můžeme nastavit pěti způsoby:

1. Cíl / Destination

- a. Toto nastavení je používané pro děkovací stránky např. po objednávce, přihlášení k newsletteru nebo registraci.

2. Doba trvání / Duration

- a. Pro tento cíl se používá požadovaná doba trvání návštěvy, v praxi tento typ cíle nemá moc využití.

3. Počet stránek nebo obrazovek na návštěvu / Pages/Screens per session

- a. Tento cíl se využívá při měření zainteresovanosti zákazníka.

4. Událost / Event

- a. Tento cíl je často používaný jako reakce na důležité události, jako je například potvrzení cookie lišty, kliknutí na tlačítko přidat produkt atp.

5. Chytrý cíl / Smart Goal

- a. Pro chytrý cíl je nutností mít propojený účet Google Analytics a účet Google Adwords. [4]

Chytré cíle využívají strojové učení k vyhodnocování desítek signálů o návštěvách vašeho webu, aby určily, které z těchto návštěv s největší pravděpodobností vyústí v konverzi. Každé návštěvě je přiřazeno bodové hodnocení a „nejlepší“ návštěvy se označí za Chytré cíle. Některé příklady signálů, které jsou zahrnuty do modelu Chytrých cílů, jsou *doba trvání návštěvy, počet stránek na návštěvu, lokalita, zařízení a prohlížeč*. [4]

K určení nejlepších návštěv vytvoří funkce Chytré cíle prahovou hodnotu tak, že vybere přibližně nejlepších 5 % návštěv vašeho webu pocházejících z kampaní AdWords. Když je určena prahová hodnota, Chytré cíle ji uplatní na všechny návštěvy webu, včetně návštěv z jiných kanálů než AdWords. Když aktivujete funkci Chytré cíle ve službě Google Analytics, je možné tyto cíle importovat do služby AdWords. [4]

1.11.5 Vyhledávače - Neplacené vyhledávání

Google Analytics mají na stránce <https://support.google.com/analytics/answer/2795821?hl=cs#searchEngine> seznam, už v jejich systému zahrnutých vyhledávačů. Jsou zde všechny známé větší vyhledávače, avšak existuje pár vyhledávačů, například atlas.cz, centrum.cz, zboží.cz atp., které v seznamu zahrnuté nejsou. Je tedy potřeba v nastavení Google Analytics na úrovni služby tyto vyhledávače nastavit. Pokud se tomu tak nestane, v reportech Google Analytics nebudou mít tyto vyhledávače medium Organic (neplacené vyhledávání), ale budou mít medium Referral (odkaz). [17][27]

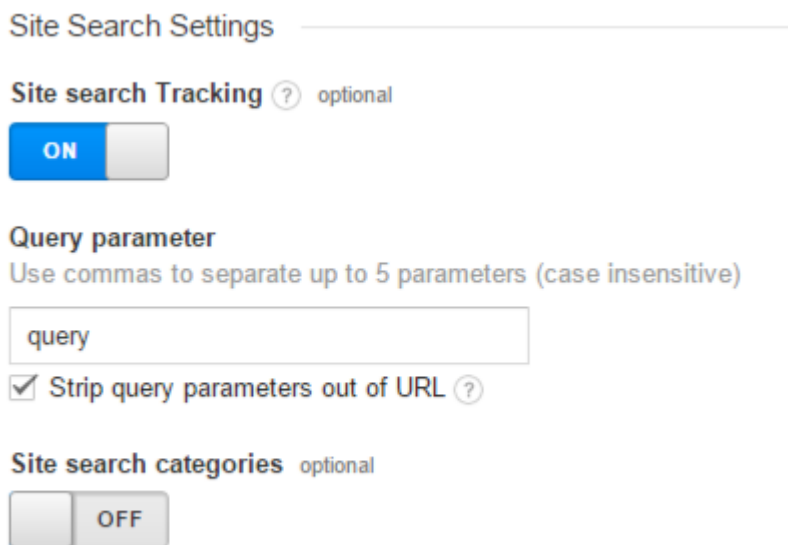
1.11.6 Sledování vyhledávání na webu tzv. site-search

Nastavení site-search umožní sledovat používání vyhledávání na měřené stránce a zároveň sběr vyhledávacích výrazů, které návštěvníci používají. K nastavení je potřeba najít v url adrese při vyhledávání na stránce tzv. query parametr, který má za sebou právě

vyhledávací výraz. [23]



Obrázek 9 - Vyhledávání na webu - query parametr [Vlastní archiv autora]



Obrázek 10 - Vyhledávání na webu - nastavení [Vlastní archiv autora]

2. Praktická východiska

V další části bakalářské práce popíší tvorbu webové aplikace na zpravu filtrů pro Google Analytics. Tato aplikace urychluje a automatizuje prvotní nastavení filtrů pro nově založený nebo nefiltrovaný profil.

2.1 Drátěný model

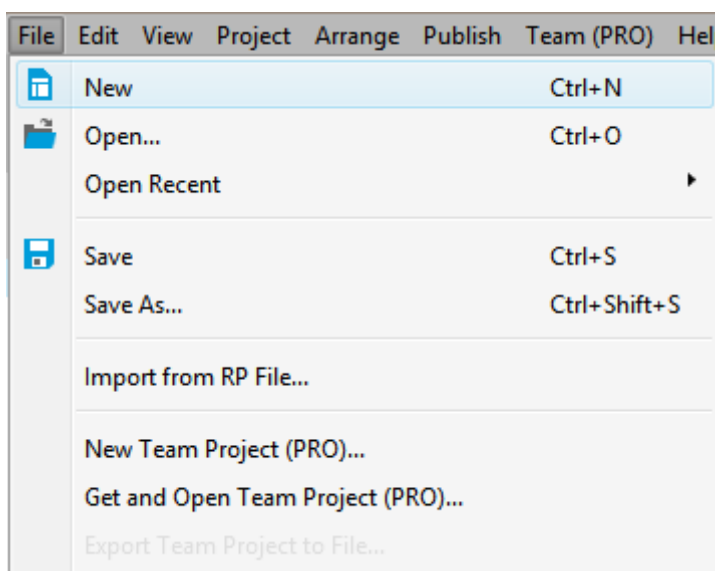
Při návrhu webu nebo webové aplikace je velmi důležitá vstupní přípravná část. V té se určuje strategie stránek, definují způsoby, jak naplnit očekávané cíle a výsledky.

V návaznosti na to je určeno i členění a podoba webu. Vhodně navržená charakteristika stránek, jejich podoba v hrubých rysech (kostra), je základem budoucí úspěšnosti webu. S návrhem podoby, obsahových dispozic a funkcí pomáhá tzv. drátěný model, neboli wireframe. [8]

2.2 Tvorba drátového modelu

Pro tvorbu drátového modelu byl vybrán pro tvořenou aplikaci program Axure. Tento program je ve zkušební verzi na 30 dní zdarma a má uživatelsky přívětivé prostředí a snadné ovládání.

Při tvorbě si nejprve musíme vytvořit nový projekt. Na hlavní aplikační liště zvolíme záložku File a zvolíme Nový projekt.



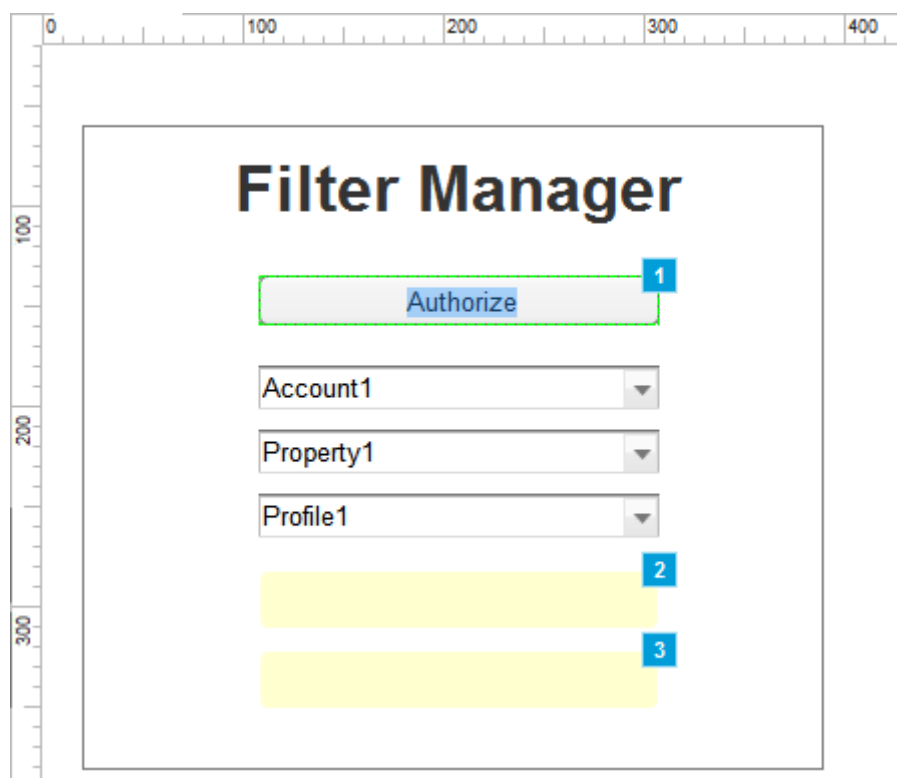
Obrázek 11- Axure - Nový projekt [Vlastní archiv autora]

V levé části aplikace v poli Widgets volíme jednotlivé prvky, které v aplikaci jsou použity. Prvky přetahujeme pomocí levého tlačítka na myši, tažením na zvolenou pozici. Pro aplikaci tvořenou v této práci jsou zvolené prvky následující

Nadpis H1 pro název aplikace

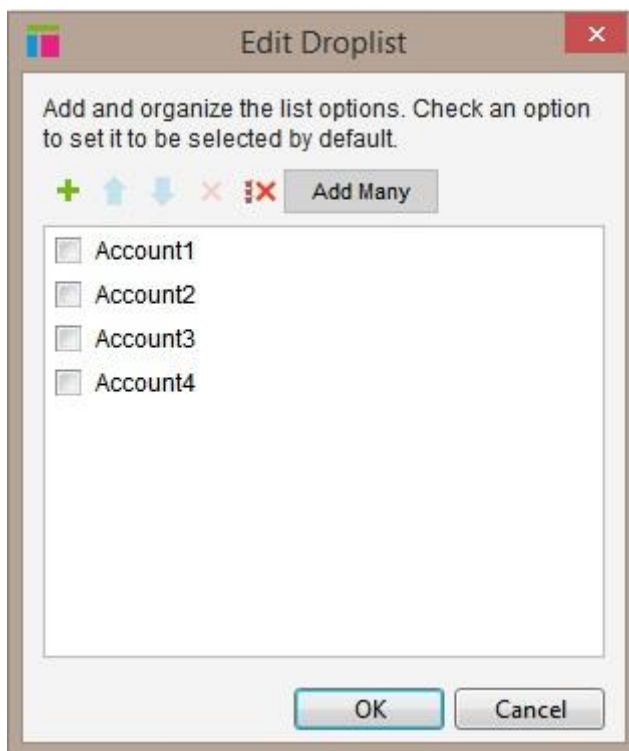
Pro každou úroveň účtu Google Analytics jeden Droplist.

Tři tlačítka neboli buttony, pro základní funkce, která aplikace Filter Manager využívá.



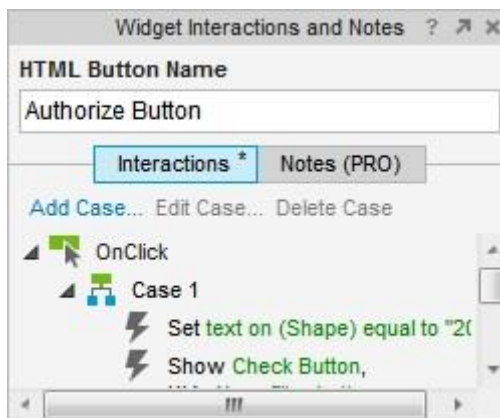
Obrázek 12 - Axure - Drátový model aplikace [Vlastní archiv autora]

Prvkům přiřadíme textové hodnoty pomocí dvojkliku nebo pravého tlačítka. U droplistů můžeme dvojklikem také přidat hodnoty pro rolovací menu, kliknutím na prázdný checkbox vybereme hodnotu, která se bude zobrazovat uživateli jako první. Button Check a New Filter schováme pravým tlačítkem a volbou Set Hidden. V dalších krocích pomocí interakcí tyto tlačítka budeme postupně zobrazovat.



Obrázek 13 - Axure - Nastavení droplistu [Vlastní archiv autora]

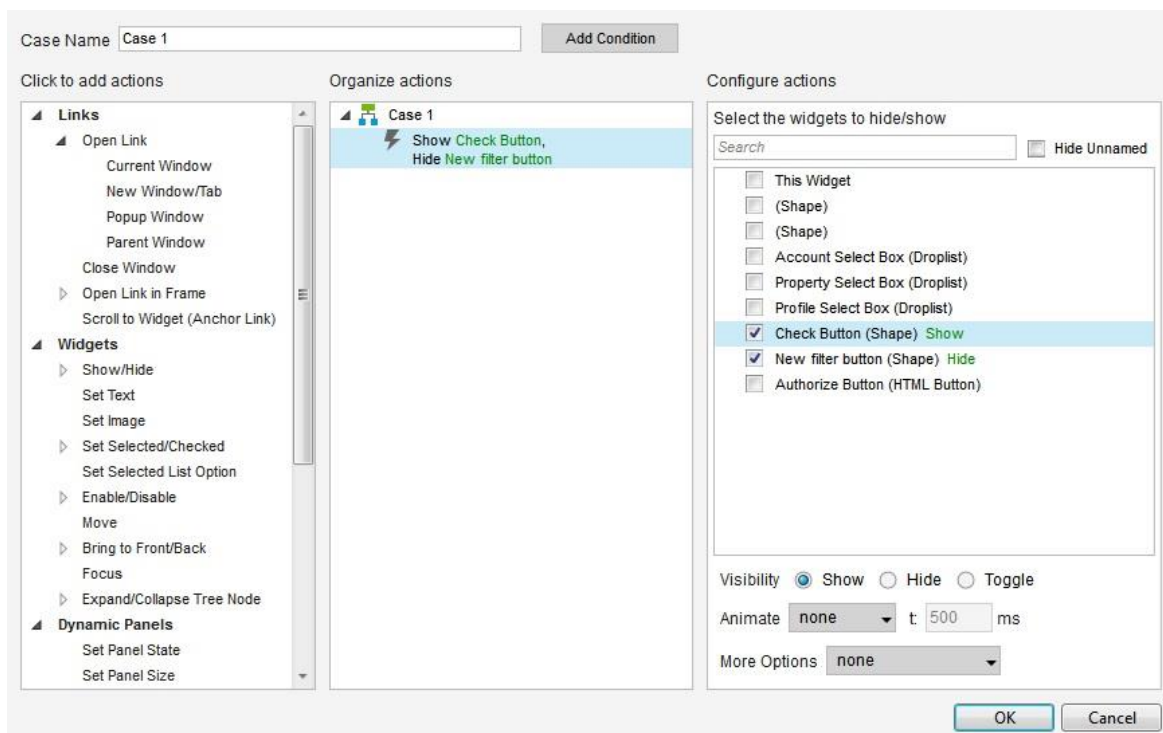
V pravé části programu v poli Widget Interactions and Notes přidáme každému prvku název pro přehlednost při přidělování interakcí.



Obrázek 14 - Axure - Interakce [Vlastní archiv autora]

Dalším krokem bude přidělování interakcí prvkům. Interakce můžeme zvolit ve stejném okně Widget Interactions and Notes. V případě této aplikace jsem použil pouze tři interakce, a to u každého tlačítka. Všechny interakce jsou typu OnClick. To znamená, že se zvolená akce

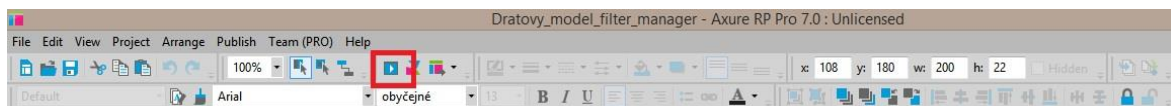
spustí po kliknutí na vybraný prvek. Událost OnClick vybereme kliknutím a v zobrazeném menu zvolíme akci, která se má provést. V případě vybrané aplikace pro tlačítko Authorize je to akce Show/Hide, která zobrazí schovaný button Check, který vybereme v pravé části zobrazeného okna.



Obrázek 15 - Axure - Události a Interakce [Vlastní archiv autora]

Stejný postup provedeme i pro tlačítko Check s tím rozdílem, že bude odhalovat poslední tlačítko New Filter. Každý tento button vyjadřuje určitou funkci. Authorize napojí aplikaci na API a stáhne data pro uživatele a jeho přístupy k jednotlivým účtům v GA se strukturou Account, Property a Profile. To vyjadřují naše Droplisty, které se dynamicky mění podle výběru vždy v nadřazeném Droplistu. V Axure se mi tato funkčnost nepodařila nasimulovat, a tak jsem select boxům dal jen pár hodnot a dynamicky se nemění. Po Authorize se zobrazí tlačítko check, které spustí funkci pro kontrolu nastavených filtrů pro vybraný profil v select boxech. Pokud proběhne správně stažení a kontrola filtrů pro vybraný profil, zobrazí se tlačítko New Filter, který nahraje nové filtry do účtu.

Pro zobrazení drátového modelu a spuštění preview slouží klávesa F5 nebo modrá ikona s play šipkou ve prostřed.



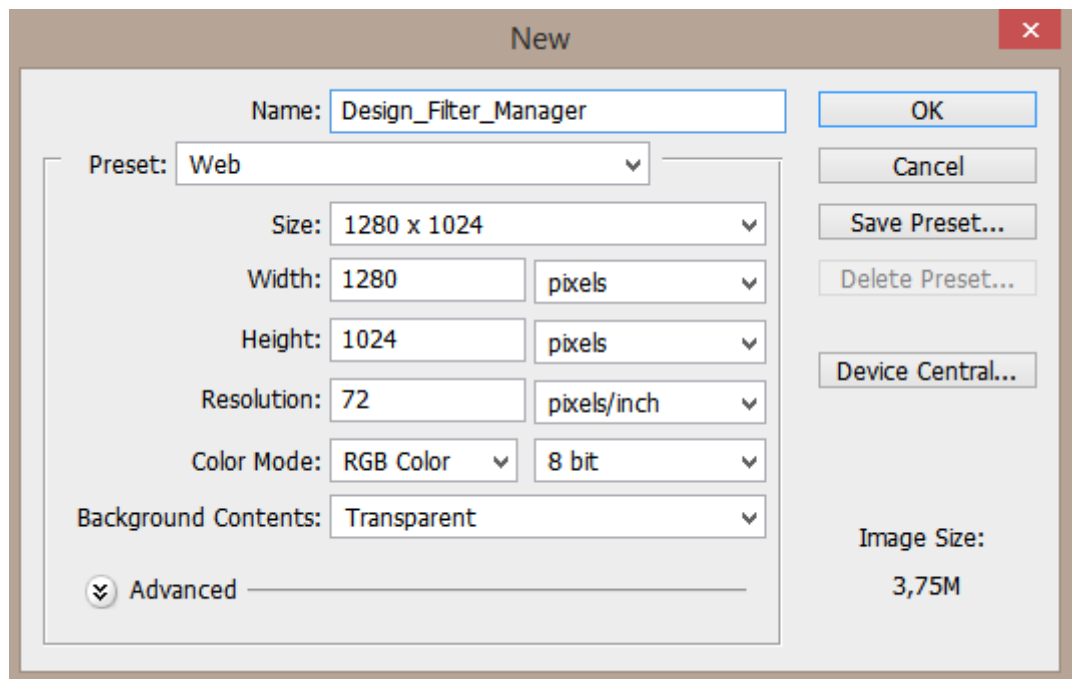
Obrázek 16 - Axure - Spuštění preview [Vlastní archiv autora]

2.3 Tvorba grafického návrhu

Pro vytvoření grafického návrhu, bylo rozhodováno mezi dvěma programy. Adobe Photoshop CC a Gimp 2. Gimp 2 je pro nekomerční použití zdarma, ale nepřehledné uživatelské prostředí a předchozí zkušenosti s programem Gimp rozhodly o výběru Adobe Photoshop CC, který je ve zkušební verzi na 30 dní také zdarma a má přehlednější a uživatelsky přívětivé rozhraní.

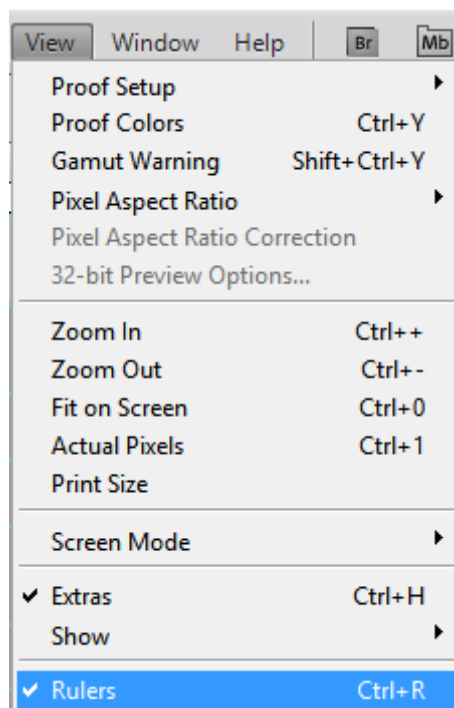
Po zvolení programu na tvorbu grafického návrhu s pomocí drátového modelu, který byl zhotoven, byl vytvořen grafický design.

Pro design této aplikace bylo zvoleno plátno s přednastavenými rozměry pro web 1280 x 1024.



Obrázek 17 - Adobe Photoshop - Nové plátno [Vlastní archiv autora]

Pro rozvržení stránky pomocí výběrů byla použita pravítka, ze kterých je možné vést uchycovací přímky, na které je možné uchytit kurzor při výběrech, vybarvování atp. Pravítka se zobrazují na hlavní programové liště v horní části aplikace, pod záložkou Zobrazení (v anglické verzi pod názvem View). Po zobrazení pravítek tažením myši směrem dolů z horního pravítka a směrem doprava z pravítka na levé straně se nastavují uchycovací přímky (Guide Lines), podle kterých byl tvořen celkový design.



Obrázek 18 - Adobe Photoshop - Pravitka [Vlastní archiv autora]

Při jednotlivých krocích tvorby designu byly vytvořené nové vrstvy, které byly seřazeny do jednotlivých skupin pro lepší přehlednost. Pro design byly nahrány tři nové styly písma z Google Fonts, které jsou zdarma.

1. Pro nadpis byl zvolený styl Racing Sans One styl od Impallari Type
2. Pro tlačítka byl zvolen styl Ubuntu od Daltona Maaga
3. Pro rozbalovací menu byl použit styl Lato od autora Łukasze Dziedzice
4. Pro patičku byl použit už zabudovaný styl Arial

Design byl zvolený dvoubarevný v odstínech šedi. Layout obsahuje patičku a hlavní sekci, které jsou od sebe oddělené. Patička je uchycena na spodní části okna a je v plné šířce. Hlavní sekce je v šířce 60 procent celkového okna.



Obrázek 19 - Adobe Photoshop - Finální Design [Vlastní archiv autora]

2.4 Výběr programovacího jazyka

Další fází při tvorbě aplikace je výběr správného programovacího jazyka. Z hlediska přenosnosti, kompatibility a předchozích zkušeností jsem zvolil skriptovací jazyk JavaScript a jeho nadstavbu TypeScript, který rozšiřuje JavaScriptovou funkčnost.

2.4.1 JavaScript

JavaScript je objektově orientovaný jazyk. Tento jazyk jsem pro aplikaci Filter Manager zvolil pro jeho jednoduchost a poměrně přehlednou syntaxi. JavaScript je interpretovaný jazyk, to znamená, že nevyžaduje další kompilaci. A může se vkládat přímo do HTML stránky. JavaScript je programovací jazyk primárně určený pro použití na klientské straně, ale čím dál tím více se stává populární nadstavba Node.JS, která je server-side tedy program a jeho výpočet probíhá na straně serveru a na uživatelském klientu se zobrazí jen výsledky. Pro účely námi

vybrané aplikace budeme potřebovat JavaScriptovou znalost, hlavně z důvodu testování a kontroly zkompilevaného TypeScriptu.

Jednou z hlavních nevýhod JavaScriptu je možnost jeho vypnutí uživatelem v prohlížeči a také problémy s různými prohlížeči a jeho verzemi, jako jsou starší verze Internet Exploreru.

2.4.2 TypeScript

Pro tvorbu aplikace byla zvolena nadstavba TypeScript. Na rozdíl od JavaScriptu nabízí TypeScript typovou kontrolu, čemuž odpovídá i část jeho názvu „type“ (v překladu „typ“). Kromě typové kontroly nabízí TypeScript tvorbu tříd, moduly rozhraní, dědičnost, IntelliSense a refactoring kódu. Mezi další jazyky podobné TypeScriptu je například Coffee Script, Script# a Dart. Oproti těmto jazykům je TypeScript nadstavbou JavaScriptu a tak je možné JavaScriptovým souborům jen změnit typ a jsou automaticky validní, například Dart disponuje jiným prostředím, než má JavaScript, musí se tedy na úkor výkonu uměle vytvářet.

2.4.3 HTML5 a Less

Pro prezentaci dat získaných JavaScriptem a grafické znázornění aplikace byl zvolen značkovací jazyk HTML verze 5. Pro stylování aplikace byl zvolen Less, který je nadstavbou klasických kaskádových stylů.

2.5 Výběr nástrojů pro tvorbu aplikace

2.5.1 Sublime Text 3 a Microsoft Visual Studio Code

Pro psaní HTML dokumentů byla vybrána open source varianta, program Sublime Text 3 a Microsoft Visual Studio Code, oba tyto editory obsahují podporu HTML, CSS a JavaScriptu. Visual Studio Code je vytvořený a doporučovaný přímo tvůrci TypeScriptu pro vytvoření webových aplikací v tomto jazyce.

2.6 Tvorba aplikace

2.6.1 Html

Pro zobrazení webové aplikace byl vytvořen HTML soubor, který obsahuje základní HTML strukturu a několik dynamických prvků, které se budou pomocí JavaScriptu dynamicky měnit. Tyto prvky byly přidány do značky `<body>`, `<body>` je rozdělena do dalších tří částí `<header>`, `<section>` a `<footer>` viz Příloha 1 [autorský archiv].

Veškerá dynamičnost a komunikace s JavaScriptem probíhá v části `<section>`. Aby mohl HTML dokument využívat JavaScript, je zde odkaz na JavaScriptový soubor v oblasti `<head>`. Hlavička dokumentu kromě odkazu na JavaScript obsahuje také odkazy na kaskádové styly, písma Google Fonts, meta tagy a titulek stránky viz *Příloha 1 [Vlastní archiv autora]*.

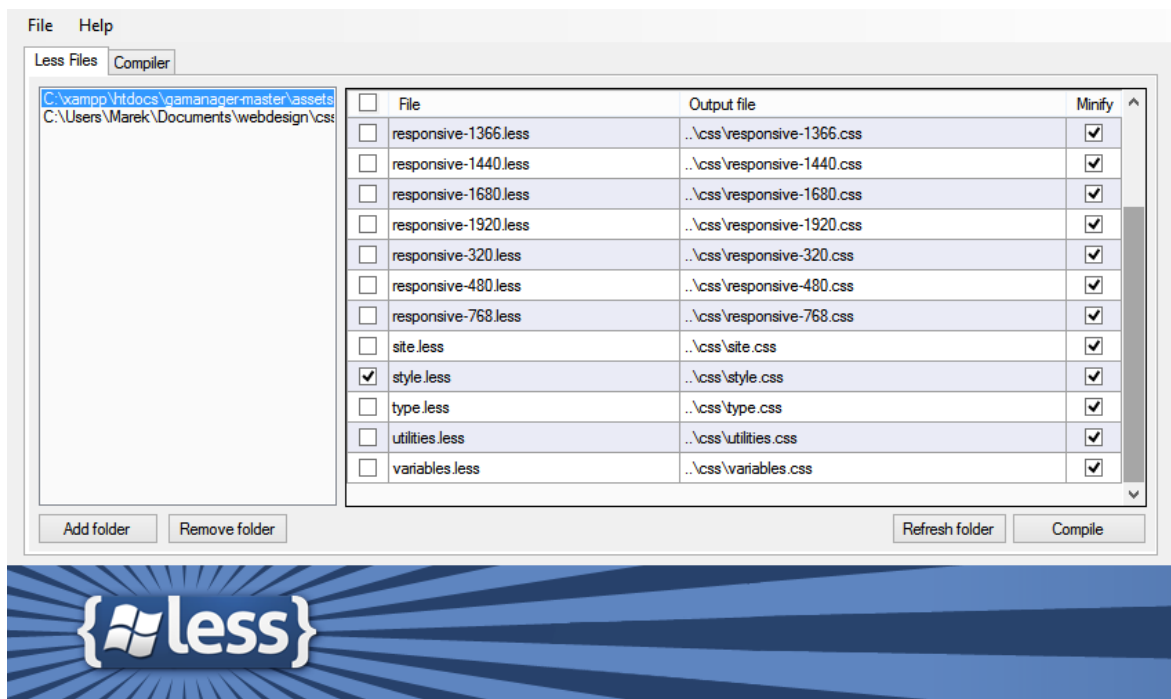
Dalším krokem při tvorbě webové aplikace bylo takzvané stylování dokumentu neboli přidání kaskádových stylů. Stylování vychází z grafického návrhu. Pro správné nastylování každého prvku byly přidány atributy. Atribut `id` a atribut `class`, také bylo použito vnořování, které funguje díky funkcionalitě formátu `less`.

2.6.2 Less a CSS

K úpravě designu stránky HTML byl zvolen formát `Less`. `Less` byl zvolen z důvodu funkcí, které `CSS3` neobsahuje. `Less` je nadstavbou kaskádových stylů a obsahuje lepší funkčnost a agilnost

`WinLess` byl zvolen pro kompilaci `Less` souborů na soubory formátu `CSS`. Aplikace spravuje kompilaci souborů a správnou interpretaci `CSS` vlastností do konečného souboru s kaskádovými styly.

Po přidání projektu do aplikace byla automaticky sledována změna souborů, kde po jejich uložení došlo k opětovné kompilaci a vytvoření finálního `style.css` souboru.



Obrázek 20 - Winless - Aplikační prostředí [Vlastní archiv autora]

2.6.3 Infrastruktura

Po stylování aplikace nastává další fáze tvorby, kterou je navržení infrastruktury aplikace. Aplikace je rozdělená do tří vrstev. Každou vrstvu reprezentuje TypeScriptový soubor, který si musíme založit.

1. App.ts
 - a. Tato vrstva komunikuje přímo s HTML souborem a zajišťuje dynamičnost prvků.
2. FilterManager.ts
 - a. Tato vrstva provádí veškeré výpočty a synchronní funkce, výsledky předává vrstvě App.ts, která je prezentuje uživateli skrz HTML kód.
3. ConnectionManager.ts
 - a. ConnectionManager je vrstva, která zpracovává a vytváří veškerou komunikaci se serverem Google Analytics. Jejím hlavní funkcí je autorizace a odesílání

požadavků přes Google API. Po stažení dat ze serveru předá výstupy vrstvě GManager.ts

Pro provázanost těchto souborů a vlastně pro provázanost jakýchkoli souborů v typescriptu se používá referenční soubor s názvem reference.ts. Tento soubor odkazuje navzájem na soubory, které spolu mají spolupracovat. Obsahuje jen referenční cesty, například pro tři výše uvedené vrstvy aplikace referenční cesty představuje tento kód:

```
    ///<reference path="./filtermanager/FilterManager.ts"
/>

    ///<reference
path="./filtermanager/services/ConnectionManager.ts" />

    ///<reference path="./app.ts" />
```

[Vlastní archiv autora]

Po připojení vrstev pomocí souboru reference.ts, na první řádek každého souboru, který chce využívat reference, referenční odkaz.

```
    ///<reference path="./reference.ts" />
```

[Vlastní archiv autora]

Vrstvy FilterManager a Connection aplikace obsahuje module s názvem filtermanager a také deklaraci třídy se stejným názvem, jako má TypeScriptový soubor. V souboru ConnectionManager.ts je deklarovaná třída ConnectionManager. Tyto dvě třídy FilterManager a ConnectionManager definujeme s prefixem export. Možnost export třídu zpřístupní napříč celou aplikací a ne jen ve svém souboru.

```
module filtermanager {

    export class ConnectionManager{

        constructor(private gam: FilterManager){}

    }

}
```



```
}
```

[Vlastní archiv autora]

Tímto způsobem založíme třídu i ve `FilterManager.ts`. Soubor `app.ts` obsahuje dvě třídy. První funkce nese název `Launcher`. Tato třída obsahuje funkce, které se spustí při načtení okna, tedy při události `window.onload`. Všem prvkům přidá určité `eventListenery`. `EventListener` vyčkává na zadanou událost a při spuštění události, například kliknutí na prvek, spustí v sobě deklarované funkce.

2.6.4 Gulp

Pro kompilaci TypeScriptových kódů do JavaScriptové podoby byl použit nástroj `gulp`. `Gulp` je buildovací nástroj, jehož cílem je sestavit aplikaci do konečné podoby. Tento nástroj využívá `nodeJS` streamy a příkazovou řádku.

`Gulp` nainstalujeme do automaticky generovaného souboru `package.JSON`, kam se ukládají moduly a nástroje nainstalované za pomoci příkazu `npm install` v příkazovém řádku. Pro založení tohoto souboru byl pomocí příkazové řádky zavolán tento příkaz.

```
npm install
```

Po připraveném `package.JSON` byly spuštěny další příkazy pro nainstalování dalších modulů např. `Gulp` a `TypeScript`.

```
npm install --save gulp  
npm install --save typescript -g  
npm install --save gulp-concat
```

`Gulp` využívá `tasky`, neboli úkoly, které se volají za pomoci konzole. `Gulp` má velmi široké použití, ale pro tuto aplikaci byly vybrány jen dvě funkce na kompilaci `typescriptu`.

Ve složce, kde je uložený `package.JSON`, je vytvořený i `gulpfile.js`. `Gulpfile.js` obsahuje tři `tasky` - `Build`, `Dev` a `Default`. `Task default` je základní `task`, který se provede při zavolání

v konzoli. Příkaz gulp nebo gulp default obsahuje console.log, který vypíše hlášku. Task build je na zavolání v konzoli pomocí příkazu

```
gulp build
```

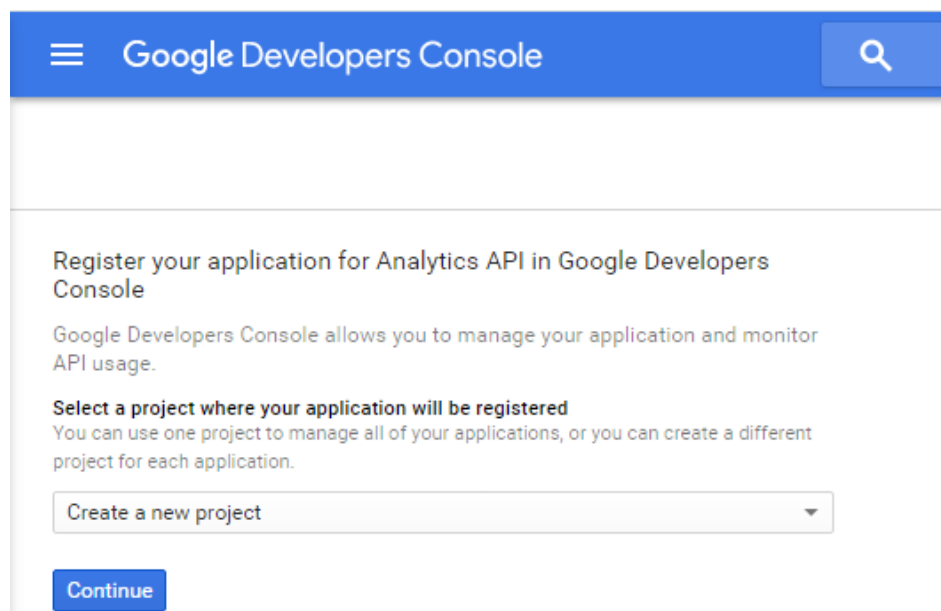
Zkompiluje veškeré typescriptové soubory a vloží do common.js.

Task dev sleduje v reálném čase ukládané změny v typescriptových souborech a automaticky při stisknutí Ctrl+S nebo přes ikonu uložení v aplikační liště Visual Studia soubory kompiluje a nahrává do common.js. Tento task se volá příkazem v konzoli

```
gulp dev
```

2.6.5 Autorizace do Google API

Pro napojení do Google API byl vytvořen vlastní projekt v Google Developers Consoli na stránce <https://console.developers.google.com/projectselector/apis/credentials>. Po založení profilu si na dashboardu svého projektu, v levé části stránky, na záložce credentials, ve které zpřístupníme Analytics API.



Obrázek 21 - Google API - Založení projektu [Vlastní archiv autora]

V záložce credentials byla zvolena v prvním poli volba Analytics API a ve druhém poli pod dotazem, odkud budeme API volat, zvolíme možnost Web browser (JavaScript).

Add credentials to your project

1 Find out what kind of credentials you need

We'll help you set up the correct credentials
If you wish you can skip this step and create an [API key](#), [client ID](#), or [service account](#)

Which API are you using?
Determines what kind of credentials you need.

Analytics API

Where will you be calling the API from?
Determines which settings you'll need to configure.

Web browser (Javascript)

What data will you be accessing?

User data
Access data belonging to a Google user, with their permission

Application data
Access data belonging to your own application

[What credentials do I need?](#)

Obrázek 22 - Google API - Přidání credentials [Vlastní archiv autora]

Nastavení pro JS origin a redirected URIs bylo nastaveno na localhost, na kterém je aplikace spuštěna. Pro localhost je JS origin nastavený na port 8080 a autorizační URL je nastavená na <http://localhost:8080/oauth2callback>

Name

Web client 1

Restrictions
Enter JavaScript origins, redirect URIs, or both

Authorized JavaScript origins
For use with requests from a browser. This is the origin URI of the client application. Cannot contain a wildcard (http://*.example.com) or a path (http://example.com/subdir).

http://localhost:8080 ×

http://localhost ×

http://www.example.com

Authorized redirect URIs
For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

http://localhost:8080/oauth2callback ×

http://www.example.com/oauth2callback

Obrázek 23 - Google API - Nastavení Origin a URIs [Vlastní archiv autora]

Po nastavení a povolení credentials v Developers Consoli je možné využívat v aplikaci Analytics API. Aby bylo ale možné data stahovat a odesílat požadavky na Google server, musí aplikace vyvolat autorizaci. Autorizace se dá provádět více způsoby, pro aplikaci byl zvolen způsob autorizace O2Auth 2.0. Využívání O2Auth vyžaduje client ID a scope. Oba tyto údaje jsou v aplikaci v části ConnectionManager, přes kterou probíhá veškerá komunikace s API. Pro requesty, které posílá aplikace FilterManager, byl zvolený scope edit a Client ID bylo doplněno z developers console.

```
export class ConnectionManager{
  // Replace with your client ID from the developer console.
  private CLIENT_ID: string = '.....-loc10si1h7ra3d6jvkfkpkpcoihhp96tu.apps.googleusercontent.com';

  // Set authorized scope.
  private SCOPES: Array<string> = ['https://www.googleapis.com/auth/analytics.edit'];
}
```

Obrázek 24 - Aplikace - Client ID a Scope [Vlastní archiv autora]

Pro autorizace využíváme knihovnu gapi a také knihovnu Promise. Promise je zvolené řešení v rámci celé aplikace pro asynchronní volání a asynchronní funkce. JavaScript funguje asynchronně a sám o sobě si nedokáže vyřešit asynchronní volání na server. V případě, kdy požadujeme po aplikaci, aby se chovala synchronně a její funkce se spouštěly jedna po druhé, přesně jak zadáme, využijeme Promise.

V TypeScriptové aplikaci si nahráváme celou knihovnu do aplikace. Pro tuto práci byla zvolena knihovna od tvůrce Mika McMahona z roku 2013, která je uložena v adresáři assets/src/Promises. Funkci, která využívá Promises musí být přetykována na Promises.Promise a musí obsahovat deklarování nové promisy a dále její naplnění nebo zamítnutí a return.

```
public authorize(): Promises.Promise {
    var authData = {
        client_id: this.CLIENT_ID,
        scope: this.SCOPEES,
        immediate: false
    };
    var d = new Promises.Deferred();
    gapi.auth.authorize(authData, (response: GoogleApiOAuth2TokenObject) => {
        if(response.error) { |
            d.reject(response.error);
        }
        else {
            d.fulfill('fulfill auth');
        }
    });
    return d.promise();
}
```

Obrázek 25 - Aplikace - Autorizace [Vlastní archiv autora]

Takovéto funkce pak voláme postupně v řetězci pomocí příkazu .then(), funkce si mezi sebou mohou předávat parametry, které vracejí returnem. Pro autorizaci a načtení údajů o Analytics účtu byl vytvořen tento řetězec:

```
public initializeReadOnly(): void{
    this.cm.authorize()
        .then(() => {return this.cm.loadAnalytics();})
        .then((param) => {return this.getAccountSummaries(param);})
        .then((param) => {return this.saveAccountSummaries(param);})
        .then((param)=>{return this.showingButtonCheck();});
}
```

Obrázek 26 - Aplikace - Řetězec funkcí typu Promises [Vlastní archiv autora]

Tento řetězec načte a uloží data o Analytics účtech, službách a profilech a vypíše je do vytvořených třech select boxů. Po dokončení všech těchto funkcí schová button pro autorizaci a zobrazí button pro kontrolu. Funkce getAccountSummaries() odešle požadavek na server pro získání account summaries, což jsou údaje o účtech, službách a profilech, kam má uživatel

přístup. Další funkce s názvem saveAccountSummaries() data přijaté od Googlu uloží do String Map Account.ts, WebProperty.ts a Profile.ts, které jsou vzájemně provázané nadřazeným prvkem. Account je rodič, ten má své property a ty zase své profily.

2.6.6 FilterManager

FilterManager zpracovává veškerá data napříč aplikací. V horní části jsou deklarovány třídy a proměnné. Je zde provázanost se string mapou Accounts, která uchovává data o účtech, dále také s ConnectionManagerem, který předává FilterManageru data z API.

```
///
```

Obrázek 27 - Aplikace - Filter Manager proměnné a constructor [Vlastní archiv autora]

Kromě už výše uvedeného řetězce funkcí s názvem initializeReadOnly() obsahuje Filter Manager dva řetězce, které se spouštějí při kliknutí na zobrazené tlačítko. Při kliknutí na tlačítko Check se spustí řetězec funkcí s názvem apiDataDownload().

```
public apiDataDownload(ids1,ids2,ids3):void{

  this.cm.makeApiCallProfileFilterLinks(ids1,ids2,ids3)
    .then((apiQuery2)=>{return this.cm.handleCoreReportingResults(apiQuery2)})
    .then((apiQuery2)=>{return this.filterLinkIdIdsGetter(apiQuery2)})
    .then((param)=>{return this.cm.makeApiCallAccountFiltersGet(ids3, param)})
    .then((apiQuery1)=>{return this.filterGetter(apiQuery1)})
    .then((param)=>{return this.downloadFilterJson(param)})
    .then((param)=>{return this.showingButtonNew();});
  console.log('Request na download filter linků odeslán');
}
```

Obrázek 28 - Aplikace – funkce apiDataDownload [Vlastní archiv autora]

Tento řetězec funkcí používá tři parametry ids1,2 a 3, které obsahují id vybraného accountu, property a profilu. Tyto parametry se předávají z vrstvy app.ts pomocí atributu value,

kde je nahraný údaj id v naplněných select boxech. První funkce v tomto řetězci odešle request na Analytics API pro získání veškerých Filter Linků pro vybraný profil. Tyto filtr linky v sobě uchovávají údaje o filtru, profilu, accountu a property. Právě z tohoto důvodu si předáváme z app.ts id vybraného profilu, property i účtu.

```
public makeApiCallProfileFilterLinks(ProfId:number,PropId:number,AccId:number):Promises.Promise {
  debugger;
  var d = new Promises.Deferred()
  debugger;
  var downloadedJSON2:any = {
    'accountId': AccId,
    'webPropertyId': PropId,
    'profileId': ProfId
  };
  if(gapi.client.analytics){
    var apiQuery2= gapi.client.analytics.management.profileFilterLinks.list(downloadedJSON2);
    d.fulfill(apiQuery2);
  }
  else{
    d.reject(apiQuery2.message)
  }
  return d.promise();
}
```

Obrázek 29 - Aplikace - funkce makeApiCallProfileFilterLinks [Vlastní archiv autora]

Po vrácení výsledku ze serveru se provede druhá funkce, která si tyto hodnoty uchová. Třetí funkce s názvem filtrLinkIdsGetter() si ze stáhnutých filter linků vybere jednotlivá filter id a předá je další funkci.

```
public AccountFiltersGetter(param):Promises.Promise{
  var d = new Promises.Deferred();
  var filters=[];
  var filterObjArray=Object.keys(param.result);
  var numberRegex = new RegExp('/^[0-9].+$/');
  if(param){
    for (var y = 0; y < param.result[filterObjArray[0]].result.items.length; y++) {
      filters[y]=param.result[filterObjArray[0]].result.items[y].id;
      console.log(param.result[filterObjArray[0]].result.items[y]);
    }
    d.fulfill(filters);
  }
  else{
    d.reject("prázdný param");
  }
  return d.promise();
}
```

Obrázek 30 - Aplikace - funkce AccountFiltersGetter [Vlastní archiv autora]

Další funkce je podobná první funkci s tím rozdílem, že místo filter linků si s pomocí filter ids stažených v předchozím kroku odešle požadavek GET na jednotlivé filtry. Tímto krokem se stáhnou veškeré údaje, které je možné o filtru získat.

Další funkce jen kontrolují, jestli jsou filtry nastavené aplikací či nikoli. V poslední řadě se skryje Check button a zobrazí se button na vkládání nových filtrů. Po kliknutí na button New Filter se opět provede řetězec funkcí.

```
///Funkce na vkládání nových filtrů
public apiDataDownload2(ids1,ids2,ids3):void{
    debugger;
    this.popupDomainSelector()
    .then((domainName)=>{return this.cm.makeApiCallInsertFilters(ids3,domainName)})
    .then((apiQuery2)=>{return this.cm.handleCoreReportingResults(apiQuery2)})
    .then((apiQuery2)=>{return this.cm.makeApiCallAccountFiltersList(ids3)})
    .then((param)=>{return this.AccountFiltersGetter(param)})
    .then((param)=>{return this.cm.makeApiCallAccountFilterLinksInsert(ids1,ids2,ids3,param)})
    console.log('Request na vložení filtru odeslán');
}
```

Obrázek 31 - Aplikace - řetězec funkcí apiDataDownload2 [Vlastní archiv autora]

První funkce otevře vyskakovací okno, do kterého uživatel zadá doménu, na které běží vybrané Google Analytics.

```
/// Popup okno na zvolení domény a IP adresy
public popupDomainSelector():Promises.Promise{
    var result=[]
    var d = new Promises.Deferred();
        var domain = prompt("Prosím zadejte svou doménu", "peregrin-bral.cz");
        result=domain.split(".");
        if (domain != null) {
            console.log(result)
            d.fulfill(result);
        }
        else{
            d.reject("Nepodařilo se zadat doménu")
        }
    return d.promise();
}
```

Obrázek 32 - Aplikace - funkce popupDomainSelector [Vlastní archiv autora]

Další funkce jsou obdobné jako v předchozím řetězci s tím rozdílem, že se neprovádí funkce get, ale insert na vkládání jednotlivých filtrů. Filtrů se vkládá víc pohromadě, a proto se z důvodu většího počtu požadavků na server jednotlivé příkazy balíčkovují do funkce batch().

2.6.7 ConnectionManager

Obsahuje požadavky na Analytics API a Autorizaci. Všechny požadavky využívají gapi knihovny. Požadavky na API vždy vyžadují poslat JSON objekt obsahující minimálně povinné proměnné. Povinné proměnné jsou pro zvolené funkce:

1. funkce filter get jsou povinné proměnné id účtu a filter id,
2. funkce filter list je jedinou povinnou proměnou id účtu,
3. funkce filter link insert jsou povinné proměnné id účtu, id služby a id profilu dále také filter id,
4. poslední nejdůležitější funkce na vkládání filtrů neboli filter insert, tato funkce musí povinně obsahovat id účtu, ale také hodnoty pro filtry.

```
var downloadedJSON2:any = [{
  "accountId": AccId,
  "resource": {
    "type": "EXCLUDE",
    "name": "Nezahrnout SPAM boty 3",
    "kind": "analytics#filter",
    "excludeDetails": {
      "kind": "analytics#filterExpression",
      "field": "REFERRAL",
      "matchType": "MATCHES",
      "expressionValue": "(^|\\.)econom\\.co|^|^\\.)(anal\\-acrobats|
ok\\.ru",
      "caseSensitive": false
    }
  }
},
```

Obrázek 33 - Aplikace - JSON s filtrem na odstranění SPAM botů [Vlastní archiv autora]

Id služby, účtu a profilu si předáváme z vrstvy app.ts do vrstvy FilterManageru a dále z FilterManageru do příslušných funkcí jako argument. Filter id si dohledáme pomocí funkce filter list, která vrátí výpis filtrů pro vybraný účet.

```

public makeApiCallAccountFiltersList(AccId:number):Promises.Promise {

    var d = new Promises.Deferred()
    var batch = gapi.client.newBatch();
    var result;
    if(gapi.client.analytics){
        var downloadedJSON:any = {
            'accountId': AccId
        };
        var apiQuery1 = gapi.client.analytics.management.filters.list(downloadedJSON);
        batch.add(apiQuery1);
        batch.then((param)=>{d.fulfill(param)}, ()=>{d.reject("nepošel batch s requesty")});
    }

    else{
        d.reject(result)
    }
    return d.promise();
}

```

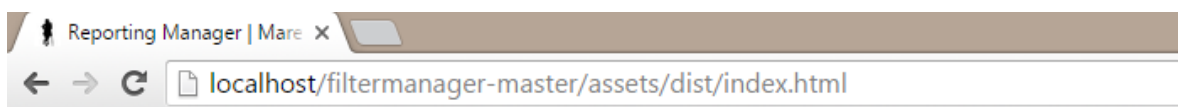
Obrázek 34 - Aplikace - funkce makeApiCallFiltersList [Vlastní archiv autora]

Ostatní funkce v ConnectionManageru fungují obdobně, jen se mění hodnoty v JSON poli posílaném na server, a to podle toho, jaký odesíláme požadavek.

2.6.8 Testování a spuštění aplikace

Pro zobrazení a debuggování aplikace byla vybrána varianta Localhostu, tedy Apache serveru vytvořeného na domácím počítači. Pro spuštění localhostu byl použit program Xampp, což je open source řešení, které se nedoporučuje používat v podnikatelské sféře z důvodu slabší bezpečnosti. Pro programování bakalářské práce je toto řešení však dostačující. K jeho spuštění je vyžadováno jen základní nastavení a Microsoft Visual C++ 2008 Redistributable ve verzi 32 nebo 64 bitů.

Pro zobrazení localhostu můžeme použít jakýkoli prohlížeč. Pro rozšířenost a kompatibilitu s HTML, CSS a JavaScriptem byl zvolen prohlížeč Google Chrome. Aplikace byla uložena na osobním počítači v adresáři C:\xampp\htdocs. Pro spuštění prohlížeči Google Chrome přes localhost uvedeme do příkazové řádky celou cestu ks souboru index.



Obrázek 35 - Google Chrome - cesta k souboru index.html [Vlastní archiv autora]

Závěr

Digitální analytika je neustále se vyvíjející obor, který v poslední době zaznamenává velké obliby hlavně pro online marketingový sektor. Tam se využívá nejen pro sledování návštěvnosti, ale také pro účely internetového podnikání, a to zejména ke sledování klíčových ukazatelů KPI či pro vyhodnocování reklamních kampaní. Pro internetové podnikání je obor digitální analytiky nezbytný.

Cílem této bakalářské práce bylo popsat způsoby fungování nástroje Google Analytics a tvoření aplikace, která by pomohla usnadnit nastavení Google Analytics.

Teoretickou část lze považovat jako úvod do problematiky webové analytiky a do nástroje Google analytics. Na začátku teoretické části byla probírána webová analytika a její historie, významné používané nástroje až po nástroj Google Analytics. Byla zde také probírána důležitá témata týkající se webové analytiky, jako jsou například cookies, Click Stream analýza, metriky a paradox dat. Po probírání základních charakteristik byl představen nástroj Google Analytics, a to zejména z technické stránky - od implementace kódu až po nasazení Enhanced Ecommerce.

Praktická část je zaměřená právě na tvorbu aplikace, která nahrává filtry do nástroje Google Analytics. Byl vytvořený drátový a grafický model aplikace, která je spuštěna na lokálním serveru.

V počátcích práce se aplikace měla věnovat veškerému nastavení, avšak studiem problematiky v teoretické části byla zjištěna omezenost Google API, kde se dají spravovat z nastavení, které bylo zamýšleno, jen filtry. Z tohoto důvodu byl změněn název aplikace z GA manageru na Filter Manager.

Aplikace je schopna během pár vteřin nahrát přes 30 filtrů, které jsou implementovány přímo v kódu. Tyto filtry se odesílají ve formě JSON objektů a uživateli této aplikace ušetří přibližně dvě hodiny práce. Tento čas se odvíjí od znalosti filtrů Google Analytics a zkušeností z praxe. Uživateli, který nemá zkušenosti s Google Analytics, tato aplikace ušetří i čas na hledání referral spamů a duplicit, které se projeví po určité době a při vyšší návštěvnosti webu. Pro tyto uživatele aplikace dokáže ušetřit čas v řádech jednotek dnů. Kromě filtrace referral spamů tato aplikace pomocí filtrů upraví zdroje a média tak, aby se zamezilo duplicitám

v datech. Pro přehlednost v datech jsou použity i takzvané lower case filtry, tedy filtry měnící u vybraných dimenzí písmena na malá pro jednotnost dat.

Aplikace je přiložená k bakalářské práci na přenosném médiu.

Seznam obrázků

Obrázek 1 - Analog - Koláčový graf prohlížečů [26]	5
Obrázek 2 - Urchin - Aplikační prostředí [24]	6
Obrázek 3 - Paradox dat [Vlastní archiv autora]	7
Obrázek 4 - Analytics - Segmentace - Časová osa [Vlastní archiv autora]	11
Obrázek 5 - Analytics - Segmentace – Porovnání [Vlastní archiv autora]	12
Obrázek 6 - Cookie první strany [Vlastní archiv autora]	14
Obrázek 7 - Cookie třetí strany [Vlastní archiv autora]	15
Obrázek 8 - Struktura nástroje [Vlastní archiv autora]	22
Obrázek 9 - Vyhledávání na webu - query parametr [Vlastní archiv autora]	26
Obrázek 10 - Vyhledávání na webu - nastavení [Vlastní archiv autora]	26
Obrázek 11 - Axure - Nový projekt [Vlastní archiv autora]	27
Obrázek 12 - Axure - Drátový model aplikace [Vlastní archiv autora]	28
Obrázek 13 - Axure - Nastavení droplistu [Vlastní archiv autora]	29
Obrázek 14 - Axure - Interakce [Vlastní archiv autora]	29
Obrázek 15 - Axure - Události a Interakce [Vlastní archiv autora]	30
Obrázek 16 - Axure - Spuštění preview [Vlastní archiv autora]	31
Obrázek 17 - Adobe Photoshop - Nové plátno [Vlastní archiv autora]	32
Obrázek 18 - Adobe Photoshop - Právítka [Vlastní archiv autora]	33
Obrázek 19 - Adobe Photoshop - Finální Design [Vlastní archiv autora]	34

Obrázek 20 - Winless - Aplikační prostředí [Vlastní archiv autora]	37
Obrázek 21 - Google API - Založení projektu [Vlastní archiv autora]	40
Obrázek 22 - Google API - Přidání credentials [Vlastní archiv autora]	41
Obrázek 23 - Google API - Nastavení Origin a URIs [Vlastní archiv autora]	42
Obrázek 24 - Aplikace - Client ID a Scope [Vlastní archiv autora]	42
Obrázek 25 - Aplikace - Autorizace [Vlastní archiv autora]	43
Obrázek 26 - Aplikace - Řetězec funkcí typu Promises [Vlastní archiv autora]	43
Obrázek 27 - Aplikace - Filter Manager proměnné a constructor [Vlastní archiv autora] ..	44
Obrázek 28 - Aplikace – funkce apiDataDownload [Vlastní archiv autora]	44
Obrázek 29 - Aplikace - funkce makeApiCallProfileFilterLinks [Vlastní archiv autora] ...	45
Obrázek 30 - Aplikace - funkce AccountFiltersGetter [Vlastní archiv autora]	45
Obrázek 31 - Aplikace - řetězec funkcí apiDataDownload2 [Vlastní archiv autora]	46
Obrázek 32 - Aplikace - funkce popupDomainSelector [Vlastní archiv autora]	46
Obrázek 33 - Aplikace - JSON s filtrem na odstranění SPAM botů [Vlastní archiv autora]	47
Obrázek 34 - Aplikace - funkce makeApiCallFiltersList [Vlastní archiv autora]	48
Obrázek 35 - Google Chrome - cesta k souboru index.html [Vlastní archiv autora]	48

Seznam použitých zdrojů

1. Adding analytics.js to Your Site. In: *Google: Google Developers* [online]. [cit. 2016-03-08]. Dostupné z: https://developers.google.com/analytics/devguides/collection/analyticsjs/#the_javascript_tracking_snippet
2. APIs Explorer. In: *Developers Google* [online]. [cit. 2016-03-05]. Dostupné z: <https://developers.google.com/apis-explorer/#p/analytics/v3/>
3. BOHÁČKOVÁ KUNOVÁ, Klára a Robert NĚMEC. Klíčové ukazatele výkonnosti (KPI) – jak sledovat to nejdůležitější. In: *RobertNemec.com* [online]. 2009 [cit. 2016-03-05]. Dostupné z: <http://robertnemec.com/klicove-ukazatele-vykonnosti-kpi/>
4. Cíle. In: *Google: Návod na Analytics* [online]. [cit. 2016-03-08]. Dostupné z: <https://support.google.com/analytics/answer/1012040?hl=cs>
5. Cookies. In: *Adaptic* [online]. [cit. 2016-03-05]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/cookies/>
6. Cookies. In: *Jak psát web* [online]. [cit. 2016-03-05]. Dostupné z: <http://www.jakpsatweb.cz/enc/cookies.html>
7. Dimenze a metriky. In: *Google: Support Google* [online]. [cit. 2016-02-11]. Dostupné z: <https://support.google.com/analytics/answer/1033861?hl=cs>
8. Drátěný model webu (wireframe). In: *Comerto* [online]. [cit. 2016-01-25]. Dostupné z: <http://www.comerto.com/tvorba-webovych-stranek-a-internetovych-obchodu/jak-tvorime-weby/drateny-model-webu-wireframe>
9. Ecommerce Tracking. In: *Google: Google Developers* [online]. [cit. 2016-03-01]. Dostupné z: <https://developers.google.com/analytics/devguides/collection/analyticsjs/ecommerce>
10. Elektronický obchod. In: *Google: Support Google* [online]. [cit. 2016-03-05]. Dostupné z: <https://support.google.com/analytics/answer/1037249?hl=cs>
11. Enhanced Ecommerce. In: *Google: Google Developers* [online]. [cit. 2016-03-01]. Dostupné z: <https://developers.google.com/analytics/devguides/collection/analyticsjs/enhanced-ecommerce>
12. FIRST PARTY COOKIE CONFUSION. *Bugdaddy.com* [online]. [cit. 2016-01-08]. Dostupné z: <http://www.bugdaddy.com/geekybug/first-party-cookie-confusion/>
13. Hello Analytics API: JavaScript quickstart for web applications. In: *Developers Google* [online]. 2016 [cit. 2016-03-05]. Dostupné z: <https://developers.google.com/analytics/devguides/config/mgmt/v3/quickstart/web-js#clientId>
14. JANEČEK, Tomáš. XAMPP návod - server ve vlastním počítači. In: *WebZpravodaj.cz* [online]. 2013 [cit. 2016-01-08]. Dostupné z: <http://www.webzpravodaj.cz/234-xampp-navod-server-ve-vlastnim-pocitaci/>
15. JANĚSKA, Jaroslav. *Principy webové analytiky*. Zlín, 2013. Univerzita Tomáše Bati ve Zlíně.
16. JANSÁ, Marek. *Využití Google Analytics v e-shopu*. Praha, 2012. Vysoká škola ekonomická v Praze.

17. KAUSHIK, Avinash. *Web analytics: an hour a day*. 1. Indianapolis, Ind.: Sybex, c2007. ISBN 9780470130650.
18. KAUSHIK, Avinash. *Webová analytika 2.0: kompletní průvodce analýzami návštěvnosti*. Vyd. 1. Brno: Computer Press, 2011. ISBN 9788025129647.
19. KUMSTA, Tomáš. *Implementace Google Tag Manageru*. Praha, 2015. Vysoká škola ekonomická v Praze.
20. LEINVEBER, Jiří. *Nástroj Google Analytics a jeho užití*. Praha, 2013. Vysoká škola ekonomická v Praze.
21. MAHARRY, Dan. *TypeScript Revealed*. 1. New York: Apress, 2013. Expert's voice in .NET. ISBN 978-1-4302-5726-4.
22. MICHÁLEK, Martin. Vše, co ještě nevíte o „EU cookies“. In: *Zdroják* [online]. 2015 [cit. 2015-11-29]. Dostupné z: <https://www.zdrojak.cz/clanky/vse-co-jeste-nevite-o-eu-cookies/>
23. Nastavení vyhledávání na stránkách a zobrazení dat vyhledávání. In: *Google: Náповěda Analytics* [online]. [cit. 2016-03-08]. Dostupné z: <https://support.google.com/analytics/answer/1012264?hl=cs>
24. Popis souborů Cookie. *Microsoft: Microfoft Support* [online]. [cit. 2015-11-29]. Dostupné z: <https://support.microsoft.com/cs-cz/kb/260971>
25. Použití na klientské straně. In: *Lesscss.cz* [online]. [cit. 2016-02-25]. Dostupné z: <http://www.lesscss.cz/#-pouziti-na-klientske-strane>
26. Příklady struktury účtu. In: *Google: Náповěda Analytics* [online]. [cit. 2016-03-05]. Dostupné z: <https://support.google.com/analytics/answer/1102152?hl=cs>
27. Přizpůsobení zdrojů organického vyhledávání. In: *Google: Náповěda Analytics* [online]. [cit. 2016-03-08]. Dostupné z: <https://support.google.com/analytics/answer/2795821?hl=cs>
28. SEDLÁKOVÁ, Jitka. Google Analytics II. – přehledy a metriky. In: *Primakurzy.cz* [online]. 2014 [cit. 2016-03-05]. Dostupné z: <http://www.primakurzy.cz/google-analytics-ii-prehledy-metriky/>
29. Urchin6.jpg. In: *Lexiconn.com* [online]. [cit. 2016-03-05]. Dostupné z: <https://www.lexiconn.com/images/urchin6.jpg>
30. Uživatelská oprávnění. In: *Google: Náповěda Analytics* [online]. [cit. 2016-03-01]. Dostupné z: <https://support.google.com/analytics/answer/2884495?hl=cs>
31. WAISBERG, Daniel. *Google Analytics Integrations*. 1. Wiley, 2015. ISBN 978-1-119-05306-4.
32. Web Server Statistics for analog documentation. In: *Chiark.greenend.org.uk* [online]. 2011 [cit. 2016-03-05]. Dostupné z: <http://www.chiark.greenend.org.uk/~sret1/stats/>
33. WEBER, Jonathan. Comparing Google Analytics Premium and Google Analytics. In: *LunaMetrics* [online]. 2015 [cit. 2016-03-05]. Dostupné z: <http://www.lunametrics.com/blog/2015/09/30/comparing-google-analytics-premium-and-google-analytics/>
34. *Webtrends: Website Measurement & Analytics* [online]. [cit. 2015-12-10]. Dostupné z: <http://www.webtrends.com/>
35. What is Digital Analytics? In: *Digital Analytics Association* [online]. [cit. 2015-12-11]. Dostupné z: <http://www.digitalanalyticsassociation.org/>

Seznam příloh

Příloha 1 [Vlastní archiv autora]

```
<html>
<head>
  <!--[if lte IE 8]><script src="js/respond.min.js"></script><![endif]-->
  <meta charset="utf-8">
  <title>Reporting Manager | Marek Aldorf</title>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="description" content="Reporting manager">
  <meta name="author" content="Marek Aldorf">
  <meta name="viewport" content="width=1024">
  <link rel="stylesheet" href="css/style.css">
  <link href='https://fonts.googleapis.com/css?family=Press+Start+2P&subset=latin-ext,latin' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Lato&subset=latin,latin-ext' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Racing+Sans+One&subset=latin,latin-ext' rel='stylesheet' type='text/css'>
  <link href='https://fonts.googleapis.com/css?family=Ubuntu:400,500,700&subset=latin,latin-ext' rel='stylesheet' type='text/css'>
</head>
<body>
  <header class="header">
    <h1 id="reloader">Filter Manager</h1>
  </header>
  <section class="main">
    <script src="./scripts/common.js"></script>
    <button id="auth-button" class="action-button shadow animate red">Authorize</button>
    <button id="acc-button" class="action-button shadow animate green">Check</button>
    <button id="new-button" class="new-button shadow animate grey">New filters</button>
    <br>
    <select id="accountSelect"></select>
    <br/>
    <select id="webPropertySelect"></select>
    <br/>
    <select id="profileSelect"></select>
    <br/>
  </section>
  <footer>
    <p> This web app is a school project created by @ Marek Aldorf 2016 mail: peregrinbral14@gmail.com </p>
  </footer>
  <script src="https://apis.google.com/js/client.js"></script>
</body>
</html>
```

Příloha 2 - Přehled dvou účtů [30]

