



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra

Bakalářská práce

Geolokace ve webovém prohlížeči pomocí HTML5 Geolocation API

Vypracoval: Martin Laudát
Vedoucí práce: PaedDr. Petr Pexa, Ph.D.

České Budějovice 2013

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 23.6. 2013

Martin Laudát

Anotace

Cílem bakalářské práce bude rozbor aktuálního stavu vývoje nové a perspektivní technologie geolokace ve webovém prohlížeči, která je součástí HTML5. Tato technologie umožňuje určit pozici dle poslání požadavku přímo z webového prohlížeče (Geolocation API). Práce se bude zabývat výhodami této technologie především ve využití pro podnikání na internetu a také nevýhodami, které souvisí s možným zneužitím osobních údajů se zaměřením na aktuální geografickou polohu. Bude provedeno i testování podpory této nové technologie v různých prohlížečích a především vytvořena sada praktických příkladů pro využití této technologie v reálném životě.

Klíčová slova

Geolokace, HTML5, Geolokační API, Webový prohlížeč

Abstract

The aim of the work will be analysis of actual state of development of a new and perspective technology geolocation in web browser, which is a part of HTML5. This technology enables to define position through sending requirement right from the web browser (geolocation API). The work will deal with advantages of this technology mainly in using it for bussiness on the internet and it also deals with disadvantages which are connected with possible abuse of personal datas with the aim on actual geographical position. Testing of support of this new technology will be done in different browsers and set of practical examples for using of this technology in real life will be created.

Keywords

Geolocation, HTML5, Geolocation API, Web browser

Poděkování

Touto cestou bych ráda poděkovala PaedDr. Petr Pexovi, Ph.D za jeho cenné rady a trpělivost při vedení mé bakalářské práce.

Obsah

| | | |
|----------|---|-----------|
| 1 | Cíle bakalářské práce | 6 |
| 2 | Metody bakalářské práce | 8 |
| 3 | Východiska bakalářské práce | 9 |
| 4 | Úvod | 10 |
| 5 | Novinky v HTML5 | 11 |
| 5.1 | Úvod | 11 |
| 5.2 | Novinky v HTML5 | 12 |
| 5.2.1 | Výběr nových elementů v HTML5 | 13 |
| 5.2.2 | Nová API a technologie v HTML5 | 17 |
| 5.3 | Podpora HTML5 ve starších prohlížečích | 19 |
| 6 | HTML5 Geolocation API | 21 |
| 7 | Funkce HTML5 Geolocation API | 24 |
| 7.1 | Ověření podpory Geolocation API v prohlížeči | 24 |
| 7.2 | Získání polohy ve webovém prohlížeči | 26 |
| 7.2.1 | Parametr Showlocation | 27 |
| 7.2.2 | Parametr ErrorHandler | 29 |
| 7.2.3 | Parametr Options | 30 |
| 7.2.4 | Ukázkový příklad jednorázového získání polohy | 31 |
| 7.3 | Periodické získávání polohy ve webovém prohlížeči | 36 |
| 8 | HTML5 Geolocation API s různými mapovými podklady | 38 |
| 8.1 | HTML5 Geolocation API a Google maps | 39 |
| 8.2 | HTML5 Geolocation API a Mapy.cz | 40 |
| 8.3 | HTML5 Geolocation API a Bing maps | 41 |
| 9 | Metody lokalizace mobilních zařízení | 44 |
| 9.1 | Lokalizace pomocí GPS Hardware | 44 |
| 9.2 | Lokalizace pomocí IP adresy | 45 |

| | | |
|-----------|---|-----------|
| 9.3 | Lokalizace pomocí datových sítí | 46 |
| 10 | Podpora HTML5 Geolocation API v prohlížečích | 48 |
| 11 | Využití geolokace na internetu | 50 |
| 11.1 | Dáme jídlo | 52 |
| 11.2 | Foursquare | 52 |
| 11.3 | TripAdvisor | 53 |
| 11.4 | Geocaching | 53 |
| 11.5 | Různé sociální sítě | 54 |
| 12 | Nebezpečí zveřejňování své polohy na internetu | 55 |
| 13 | Praktické příklady | 57 |
| 13.1 | Praktické příklady s mapovými podklady | 57 |
| 13.2 | Hlavní praktický příklad | 58 |
| 14 | Závěr | 59 |
| 15 | Literatura a zdroje | 60 |

1 Cíle bakalářské práce

Cílem bakalářské práce bude přiblížit odborné veřejnosti technologii Geolocation API obsaženou ve standardu HTML5. Tato práce se bude zabývat představením nejzásadnějších novinek HTML5, Detailním rozbořením možností této technologie v dnešní době, vysvětlením problematiky zjišťování polohy ve webovém prohlížeči a možnými riziky, které mohou hrozit při zveřejňování polohy na internetu, otestováním tohoto API v prohlížečích a vysvětlením jak řešit nekompatibilitu s určitými prohlížeči. V rámci této práce bude veřejně na internetu dostupná webová stránka, kde se bude nalézat sada praktických příkladů a jeden hlavní praktický příklad. Bude se jednat o sadu příkladů, které budou demonstrovat spolupráci HTML5 Geolocation API s různými mapovými podklady a hlavní praktický příklad bude komplexnější webová aplikace využívající toto API.

Menší praktické příklady se budou zabývat využitím HTML5 Geolocation API s nejnámějšími mapovými podklady, které jsou na internetu dostupné. Mezi tyto podklady patří Google Maps, Bing Maps, Mapy.cz a OpenLayers. Dále budou sepsány jednoduché návody jak s jednotlivými podklady pracovat a samozřejmě vytvoření hotových ukázek.

Hlavní projekt by se měl zabývat využitím HTML5 Geolocation API v komplexnější aplikaci než-li pouze zobrazení naměřených výsledků na určitých mapových podkladech. Tato webová aplikace by měla využívat geolokaci pro zpříjemnění uživatelského zážitku a pro možné generování zisku, mělo by se tedy jednat o aplikaci, která bude mít reálný smysl a bude řešit konkrétní problém. Hlavní příklad by měl ukazovat jak implementovat toto API mezi ostatní moderní webové technologie a vytvořit tak kompletní webovou aplikaci, jejímž jádrem budou mapové podklady a HTML5 Geolocation API.

Celkově by tato práce měla sloužit jako průvodce HTML5 Geolocation API pro webové vývojáře, kteří toto API chtějí použít a jako zdroj informací běžných uživatelů k pochopení problematiky zjišťování polohy na internetu a k uvědomění rizik, která mohou při sdílení polohy na internetu nastat.

2 Metody bakalářské práce

V úvodu se budu zabývat zásadními novinkami v HTML5, a pak přejdu k technologii Geolocation API, kde budu detailně rozebírat funkce tohoto API, budu testovat funkčnost API ve všech moderních prohlížečích a samozřejmě také v mobilních prohlížečích. Popíšu výhody a nevýhody této technologie, bezpečnostní rizika spojená se zveřejňováním své polohy na různých sociálních sítích. Provedu čtenáře tvorbou základní aplikace pro zjišťování a vypisování polohy ve webovém prohlížeči. Poté začnu popisovat mnou zvolený příklad a názorně ukáži postup tvorby za pomoci HTML5/CSS3/JavaScript, popíšu nasazení této technologie do obchodní sféry a na závěr shrnu celou práci.

3 Východiska bakalářské práce

Mým cílem je rozebrat a popsat novou technologii užitou v HTML5. A to technologii Geolokace ve webovém prohlížeči pomocí Geolocation API. V dnešní době, kdy se webové aplikace dostávají do mobilních zařízení a to ať do mobilních telefonů či tabletů se rozšiřují i možnosti využití aplikací na těchto zařízeních. Konkrétně na mobilních zařízeních je potřeba zjišťovat polohu zařízení větší a větší. Tuto potřebu mají hlavně různé sociální sítě (Facebook.com, Foursquare.com, atd...), ale také ostatní aplikace. Jednou z nejznámějších aplikací je GPS aplikace pro geocaching, kde podle GPS souřadnic hledáte schované poklady.

Do příchodu technologie HTML5 nebyla jiná možnost než pro zjišťování polohy vytvářet nativní aplikace pro konkrétní typy telefonu a konkrétní typ operačního systému. Tuto vlastnost technologie Geolocation API v HTML5 zcela ruší, protože se aplikace pro zjišťování polohy přesouvají z nativních mobilních aplikací do mobilních webových prohlížečů.

V dnešní době je podpora této technologie podporována ve všech moderních prohlížečích na PC a pomalu se začíná dostávat i do nejnovějších mobilních prohlížečů. S příchodem plné podpory této technologie a technologie HTML5 vznikne na webu prostor pro spousty aplikací různých zaměření.

4 Úvod

V dnešní době se potřeby uživatelů na internetu neustále rozšiřují. Zvyšují se nároky na funkce, které mohou webové stránky nabízet. Díky tomuto vývoji vznikají i nové moderní webové technologie. Nejzákladnější metodou tvorby webu je dnes značkovací jazyk HTML, který ve své poslední specifikaci HTML5 přináší spoustu revolučních novinek.

Mezi tyto revoluční novinky patří například sémantičtější zápis, nové elementy a nová API jako jsou například Fullscreen API, které nám umožňuje vytvářet aplikace pro celou obrazovku zařízení ze kterého web prohlížíme. Dalším novým API je Drag and Drop API, které nám umožňuje intuitivně přesouvat prvky na webových stránkách v reálném čase. Dalším API je Geolocation API, o kterém bude tato práce pojednávat. Geolocation API je velice jednoduché API, které nám umožňuje zjišťovat zeměpisnou polohu zařízení ze kterého se dotazujeme. A to vše pouze ve webovém prohlížeči, nepotřebujeme tedy vytvářet nativní aplikace pro určitý OS, ale vše obsloužíme ve webovém prohlížeči. Díky Javascriptu můžeme pak dále s těmito údaji pracovat a vytvářet tak moderní webové aplikace.

Dnes se webové stránky a aplikace vytvářejí hlavně pro uživatele, který je bude používat. Existují způsoby jak navrhnout dobré webové rozhraní na kterém se uživatel bude chovat intuitivně a nebude bloudit. Takto navržený web dokáže uživatele na webu udržet a dokonce může uživatele přimět k nějaké interakci, například nákupu. Díky novým technologiím můžeme takovéto weby dělat snadněji než dříve.

5 Novinky v HTML5

5.1 Úvod

Značkovací jazyk HTML od své první verze ušel už pořádný kus cesty. Stále je nejrozšířenějším jazykem pro popis dat nebo obsahu webových stránek. Jeho poslední verze nese název HTML5, které přináší spoustu nových funkcí, vylepšuje stávající funkce a obsahuje i několik důležitých API postavených na skriptech. Všechny tyto vlastnosti nám umožňují vytvářet moderní webové stránky s řadou užitečných funkcí a vylepšení. Díky tomu, že HTML5 obsahuje nové značkovací prvky, mohou být stránky přístupnější a srozumitelnější. Dnes nemusíme na hlavní prvky layoutu webových stránek používat pouze značku `div`, ale můžeme použít místo toho značky jako: `header`, `nav`, `section`, `article`, `footer` a další. Použitím těchto značek se sémantika webových stránek značně zpřehlední.

HTML5 není zcela novou verzí předchozího HTML respektive XHTML, obsahuje prvky, které byly použity v HTML4 a XHTML 1.0. Díky tomu je částečně HTML5 podporováno i ve starších prohlížečích. Respektive elementy, které byly zároveň použity v předchozích specifikacích. Pro nové elementy v HTML5 musíme podporu ve starších prohlížečích zařídit. Cílem bylo vytvořit značkovací jazyk, který je modernější než jeho předchůdci, bude kompatibilní s většinou dnešních platforem, bude umět správně obstarávat chyby a bude lehce udržovatelný.

Samotný termín HTML5 je často používán pro sadu nástrojů, různých API a nových technologií pro tvorbu moderních webů. Může to být Geolocation API, které slouží pro vyhledání polohy zařízení přímo ve webovém prohlížeči, nebo to může být plátno `canvas`, které nám umožňuje ve webovém prohlížeči kreslit různé grafické prvky. Může se také jednat o novou podporu `audio` a `video`, vytváření offline aplikací či `Drag-and-drop` API a další. Je třeba zmínit, že se pod názvem HTML5 ukrývají i technologie, které tam

patřili, ale časem se oddělily do vlastních vývojových větví nebo technologie, které do HTML5 nikdy nepatřili.

Technologie HTML5 původně vznikala jako dvě různé specifikace, tyto specifikace se nazývaly Web Forms 2.0 a Web Apps 2.0, tyto specifikace byly reakcí na změny webových stránek a aplikací. Byly řešením pro moderní, svižné a efektivní webové stránky a aplikace. Je tedy logické, že se stejnými směry vydala i specifikace HTML5. Současně s vývojem HTML5 se vyvíjelo i XHTML 2.0, ale vývoj XHTML byl později ukončen. Je známým faktem, že za vývojem HTML5 stojí dvě organizace. W3C (World Wide Web Consortium), které se stará o původní specifikace HTML a CSS specifikace dalších technologií. A tou druhou organizací je WHATWG (Web Hypertext Application Technology Working Group), která se skládá z lidí ze společností Apple, Mozilla a Opera. Organizace WHATWG vznikla proto, že se lidem, kteří tuto organizaci založili, nelíbilo, že organizace W3C ignoruje potřeby a požadavky společností, které vyvíjejí webové prohlížeče. Takže organizace WHATWG začala tvořit specifikace Web Forms 2.0 a Web Apps 2.0, které byly zmiňovány výše. Ovšem po určité době došlo ke sloučení těchto dvou specifikací a to dalo za vznik nové specifikaci HTML5 na kterou později přešlo i W3C. Specifikace od WHATWG (<http://www.whatwg.com/html/>) obsahuje více nových a experimentálních funkcí než specifikace od W3C (<http://dev.w3.org/html5/spec/>). Obě zmíněné specifikace jsou si velice podobné. Obecně se dá říci, že se HTML5 skládá ze specifikace W3C a specifikace WHATWG. Závěrem je třeba říci, že HTML5 ještě není zcela dokončené a je třeba si uvědomit, že jeho použití je stále na experimentální úrovni.

5.2 Novinky v HTML5

Jak již bylo zmíněno, HTML5 přichází s několika novinkami a vylepšeními, které nám dovolují vytvářet modernější, více sémantický a celkově zajímavější web než tomu bylo doposud. Pojdme se tedy podívat na ty nejzajímavější novinky trochu blíže.

5.2.1 Výběr nových elementů v HTML5

Doctype

Definice dokumentu se s příchodem HTML5 razantně změnila. Při definování HTML5 dokumentu stačí prohlížeči říct, že se jedná o HTML dokument a víc nic už psát nemusíte. Zápis definice dokumentu je tím pádem mnohem jednodušší.

Příklad zápisu definice dokumentu HTML 4.01 Transitional:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Příklad zápisu definice dokumentu HTML5:

```
<!DOCTYPE html>
```

Article

Element article reprezentuje část obsahu stránky, který tvoří samostatnou a nezávislou část, například článek nebo komentář.¹ Element může obsahovat i další elementy a je párový.

Ukázka zápisu:

```
<article>
  <!-- obsah např. článek pro blog -->
</article>
```

Aside

Element aside slouží pro popis nějakého obsahového prvku webové stránky. Jedná se spíše o sekundární obsah. Například u nějakého článku může aside

¹HTML5 - nové vlastnosti: Article. In: ŠŤASTNÝ, Jiří. *Programujte.com* [online]. 2011 [cit. 2013-02-21]. Dostupné z: <http://programujte.com/clanek/2010082200-html5-nove-vlastnosti/>

plnit funkci doplňkových informací viz. ukázka zápisu. Nejčastější použití bude pro deklaraci nějakého postranního panelu. Jedná se o párový tag.

Ukázka zápisu:

```
<article>
<!-- hlavní obsah nějakého článku -->
  <aside>
    <!-- sekundární obsah, doplňkové informace -->
  </aside>
</article>
```

Canvas

Element canvas nemá sám o sobě žádnou funkci, tento element pouze obaluje grafiku. Jak název sám napovídá, jedná se o plátno (canvas = plátno). Na toto pomyslné plátno můžeme kreslit za pomoci scriptů. Tento element je párový.

Ukázka zápisu:

```
<canvas></canvas>
```

Figcaption a figure

Element figcaption obaluje nějaký prvek například obrázek a element figure přidává k tomuto obrázku popisek. Element figure je tedy potomkem elementu figcaption. Oba elementy jsou párové.

Ukázka zápisu:

```
<figure>
  
  <figcaption>Popisek obrázku</figcaption>
</figure>
```

Header

Element header slouží pro popis hlavičky dokumentu nebo jen hlavičky úseku dokumentu - článku. Při vytváření layoutu webu již nemusíme vytvářet `<div id="header">`, ale můžeme použít element header. Jedná se o párový element.

Ukázka zápisu:

```
<header>
  <!-- obsah hlavičky -->
</header>
```

Footer

Podobně jako header, element footer popisuje patičku dokumentu nebo jen úseku dokumentu. Opět nemusíme používat nesémantický zápis `<div id="footer">`, ale můžeme použít element footer. Jedná se o párový element.

Ukázka zápisu:

```
<footer>
  <!-- obsah patičky -->
</footer>
```

Hgroup

Element hgroup se používá pro seskupování více nadpisů (h1...h6). Tento element je párový a smí obsahovat pouze nadpisy.

Ukázka zápisu:

```
<hgroup>
  <h1>Nadpis H1</h1>
```



```
    <h2>Podnadpis H2</h2>
      <h3>Podnadpis H3</h3>
</hgroup>
```

Nav

Jak samotný název elementu napovídá, jedná se o element popisující navigaci. Tento element se používá pro popis hlavní navigace webových stránek. Jedná se o párový element.

Ukázka zápisu:

```
<nav>
  <ul>
    <li>Prvek navigace 1</li>
    <li>Prvek navigace 2</li>
    <li>Prvek navigace 3</li>
  </ul>
</nav>
```

Section

Element section slouží pro popis části dokumentu, která je samostatná a její vyjmutí nenaruší logiku celého webu. Tedy když obsah elementu section odstraníme, logika stránky zůstane stejná a nic tím nenarušíme. Může se jednat například o nějaký článek na webu. Element je párový.

Ukázka zápisu:

```
<section>
  <h2>Nadpis</h2>
  <p>Obsah</p>
</section>
```

Time

Time element slouží pro zápis časových údajů na webu. Umožňuje nám zvolit si vlastní formát času. Jedná se o párový tag.

Ukázka zápisu:

```
<time datetime="2013-02-25"> 25.listopadu 2013</time>
```

5.2.2 Nová API a technologie v HTML5

Geolocation API

Geolocation API nám umožňuje zjistit polohu zařízení, ze kterého se dotazujeme přímo ve webovém prohlížeči. Nemusíme tedy používat žádné nativní aplikace, ale stačí nám k tomu pouze webový prohlížeč s podporou HTML5 a javascriptu.

Drag and Drop API

Toto API nám umožňuje na úrovni webového prohlížeče přetahovat a přemisťovat objekty na webové stránce. Můžeme si to představit tak, že na webové stránce máme dva boxy jeden modrý a druhý žlutý. V modrém boxu je obrázek, který díky tomuto API můžeme přetáhnout do žlutého boxu, kam se obrázek umístí a zůstane tam.

Full-screen API

Full-screen API nám umožňuje spouštět aplikace v plném zobrazení, tedy přes celou plochu. V minulých letech se toto řešilo pomocí Flash aplikací, ale příchodem HTML5 už tento problém můžeme řešit přímo v prohlížeči a nepotřebujeme žádné aplikace navíc.

File API

V moderních webových aplikacích je potřeba spravovat uživatelské vstupy, které často obsahují i vkládání souborů. HTML5 File API nám umožňuje zpracovávat data, které uživatel vkládá přímo na straně uživatele. Můžeme si tedy tyto data na straně uživatele připravit a až poté odeslat na server, kde s nimi pomocí nějakého serverového jazyka jako je např. PHP můžeme pracovat dál.

Offline web applications

Díky této technologii, můžeme vytvářet webové aplikace, které mohou částečně fungovat i bez trvalého připojení k internetu. Takže můžeme mít webovou aplikaci, která při ztrátě připojení k internetu bude částečně fungovat dál. Respektive můžeme zvolit data, která se budou ukládat na uživatelský počítač a budou dostupná i bez připojení k internetu – serveru.

Audio a video

Nová technologie, která dokáže přímo v prohlížeči přehrávat audio a video soubory bez aplikací třetích stran jako je Adobe Flash. Jen za pomoci elementů audio a video. Takže pro přehrávání těchto souborů nám stačí pouze prohlížeč, který podporuje HTML5 audio a video technologii.

CSS3

Společně s novou specifikací HTML5 přišla i specifikace CSS3, která přináší spoustu nových funkcí a vylepšení. Díky CSS3 mohou webové stránky dostat úplně jiný rozměr. Mezi hlavní novinky patří možnost vytváření animací (a to i 3D), nové efekty, jako zaoblení rohů, vržený stín nebo barevný přechod, které můžeme použít na objektech. Další novinky jsou tzv. Media queries, tato funkce nám umožňuje přizpůsobovat webovou stránku zařízením, se kterými webové stránky prohlížíme, jedná se například o chytré

mobilní telefony nebo tablety. Je třeba ovšem říci, že specifikace CSS3 není ještě úplně hotová a proto je CSS3 stále v experimentální fázi. S každou novou verzí prohlížečů přichází i podpora pro nové funkce CSS3. Ve starších prohlížečích můžeme CSS3 simulovat pomocí javascriptových knihoven.

5.3 Podpora HTML5 ve starších prohlížečích

Nové elementy v HTML5 jsou podporovány ve všech „moderních“ prohlížečích – v aktuálních verzích prohlížečů. Samozřejmě je podíl starších prohlížečů stále značně vysoký, proto při vývoji moderních aplikací v HTML5 musíme myslet i na uživatele, kteří používají starší neaktualizované prohlížeče.

Pokud chceme používat HTML5 a jeho nové elementy musíme tedy nějak zařídit podporu těchto elementů i ve starších prohlížečích. Tento problém nám pomůže vyřešit javascriptová knihovna HTML5shiv². Díky této knihovně můžeme používat HTML5 elementy v prohlížečích jako Internet Explorer 6-9, Safari 4.x, iPhone 3.x a Firefox 3.x. Instalace je velice jednoduchá, stáhneme tuto knihovnu, uložíme jí do složky, kde máme vytvořenou HTML5 webovou stránku nebo aplikaci a pak pouze v hlavičce webové stránky se odkážeme na tuto knihovnu. Nic víc není potřeba. A aplikací tohoto postupu můžeme vytvářet HTML5 stránky nebo aplikace, které budou fungovat i ve starších prohlížečích.

Jestliže ovšem chce přidat podporu nějakého HTML5 API, musíme použít složitější postup. Základním prvkem je Modernizr³ – knihovna, která nám zjišťuje podporu nových elementů a technologií v prohlížeči. Tedy jestliže Modernizr zjistí, že prohlížeč nové technologie podporuje, tak nic neudělá a my můžeme pracovat dál. Jestliže ovšem zjistí, že prohlížeč tyto technologie nepodporuje, vyvolá událost, která informuje uživatele, že jeho prohlížeč není

²Více na: <https://github.com/aFarkas/html5shiv>

³Více na: <http://www.modernizr.com>

kompatibilní s těmito technologiemi a funkcemi. Tohoto můžeme využít a při zjištění, že prohlížeč danou technologii nepodporuje, můžeme podstrčit alternativní řešení – tzv. polyfills⁴. Jedná se o většinou javascriptové knihovny, které simulují moderní technologie jako je například podpora formátu *.SVG ve starších prohlížečích. Můžeme tak docílit toho, že v moderních prohlížečích budeme používat nativní technologie obsažené v HTML5 a ve starších prohlížečích budeme tyto technologie simulovat pomocí těchto polyfillů.

⁴Seznam polyfillů: <https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-Browser-Polyfills>

6 HTML5 Geolocation API

Hlavním cílem této práce je představit si a popsat novou technologii obsaženou v HTML5 a to geolokační rozhraní – Geolocation API. Účelem HTML5 Geolocation API je zjišťovat polohu zařízení, ze kterého se dotazujeme přímo ve webovém prohlížeči, a s těmito informacemi dále pracovat. Můžeme zjistit aktuální polohu zařízení (v souřadnicovém systému), můžeme zjistit aktuální nadmořskou výšku, přesnost získaných dat, natočení zařízení, ze kterého se dotazujeme, rychlost pohybu zařízení, ale můžeme zjišťovat polohu i průběžně a vytvářet tak záznam o pohybu. Dále nám umožňuje jednoduše obsluhovat chyby pro příjemný uživatelský zážitek. K těmto informacím můžeme přistupovat pomocí javascriptu a v javascriptu s nimi můžeme i dále pracovat. Toto API není nijak složité, hlavní metodou tohoto API je metoda `getCurrentPosition`. Tato metoda obsahuje tři parametry.

Těmito parametry jsou:

- Parametr `showLocation` – určuje callback metodu, kterou prohlížeč zavolá po získání informace o poloze⁵.
- Další parametr je `ErrorHandler` – určuje callback metodu, která se zavolá při vzniku nějakého problému.
- Posledním parametrem je `Options` – tento parametr nám umožňuje nastavit další možnosti.

Zápis metody `getCurrentPosition` bude vypadat tedy nějak takto:

```
getCurrentPosition (showLocation, ErrorHandler, options);
```

Geolocation API je revoluční technologií, protože již nemusíme být vázáni na nativní aplikace, ale můžeme pracovat s polohou zařízení přímo ve webovém prohlížeči. Před uvedením HTML5 a tohoto API jsme museli vytvářet

⁵HTML5 – geolokační rozhraní. *Programujte.com - web o programování, webdesignu, počítačové grafice, databázích, elektrotechnice a designu* [online]. 2012 [cit. 2013-03-14]. Dostupné z: <http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/>

nativní aplikace pro každou platformu zvlášť, což bylo velice komplikované a složité. Dnes nám stačí jedna webová aplikace, která běží v prohlížeči pro všechny dnešní běžné platformy, jako jsou Windows, Android a iOS. Geolocation API je celkem připravené na nasazení do ostrého provozu, ovšem je třeba si uvědomovat, že je to stále technologie na experimentální úrovni a i když je podpora v moderních prohlížečích velice slušná, tak ve starších prohlížečích tato technologie nemusí fungovat vůbec. Proto při vývoji aplikace, která bude Geolocation API používat, musíme na tuto skutečnost myslet a nechat si otevřené zadní vrátka, do kterých můžeme podstrčit nějakou simulaci Geolocation API v případě, že na zařízení, ze kterého se budou uživatelé dotazovat, nebude Geolocation API plně funkční.

HTML5 Geolocation API se ovšem nestará o proces zjištění aktuální polohy. O tento proces se stará samo zařízení, ze kterého se dotazujeme. Geolocation API dá pouze podnět k získání polohy ze zařízení a o samotné zjištění se pak postará zařízení, ze kterého se dotazujeme. Z toho vyplývá, že technologii pro určení aktuální polohy si volí samo zařízení z jeho dostupných zdrojů. Takže způsob, jakým zjistíme naši aktuální polohu, závisí pouze na zařízení. Způsobů zjišťování polohy je několik, mezi hlavní tři způsoby patří: zjištění polohy skrze IP adresu, zjištění polohy přes GPS modul a posledním způsobem je zjištění polohy pomocí dat z přístupových bodů jako jsou wifi sítě, mobilní sítě atd. Po vytvoření požadavku na získání aktuální polohy si zařízení samo vybere jakou metodu zvolí, respektive má-li k dispozici GPS modul zjistí polohu pomocí tohoto modulu, pokud zařízení GPS modul nemá zjistí polohu pomocí dalších způsobů – IP adresa nebo datové sítě. Samozřejmě nejpřesnější metodou je zjišťování polohy pomocí GPS modulu. Zbylé dva způsoby už tak přesné nejsou. Zjišťování polohy přes IP adresu zařízení se provádí tak, že se zjistí fyzická adresa providera vašeho internetového připojení, tudíž jestliže má váš provider sídlo na druhé straně republiky, získaná poloha bude právě z tohoto místa. Tato metoda je proto velice nepřesná. Posledním způsobem je zjišťování polohy pomocí datových sítí jako jsou WIFI sítě nebo mobilní sítě operátorů. Zjištění probíhá tak, že se zjistí vzdálenost zařízení z několika těchto přístupových bodů – vysílačů.

Přesnost této metody je ovlivněna počtem těchto vysílacích stanic v okolí zařízení, ze kterého se dotazujeme. Čím více přístupových bodů tím větší přesnost získané polohy.

7 Funkce HTML5 Geolocation API

V předchozí kapitole jsem představil HTML5 Geolocation API. V této kapitole ukážu, jak s tímto API pracovat. Použití tohoto API je velice snadné a s tímto API se dobře pracuje.

HTML5 Geolocation API můžeme ovládat pomocí javascriptu. Jelikož se jedná o webovou technologii je potřeba chápat základní vztahy a pravidla tvorby webu. Opět je důležité si uvědomit, že toto API a celkově celé HTML5 je v experimentální fázi, takže při použití je třeba přemýšlet, zda-li můžu danou technologii použít nebo ne.

7.1 Ověření podpory Geolocation API v prohlížeči

HTML5 nám dovoluje jednoduše a elegantně otestovat podporu HTML5 Geolocation API v prohlížeči. Můžeme použít javascriptovou knihovnu Modernizr, ale v tomto případě je mnohem snazší použít jednoduchou javascriptovou podmínku. Vždy je důležité otestovat, zda-li uživatelův prohlížeč podporuje Geolocation API a zda-li uživatel dal svolení ke zjištění jeho polohy. V případě že nastane jedna z těchto událostí, je třeba o tom dát uživateli jasně vědět.

Příklad 1 – Ověření podpory Geolocation API

```
if (navigator.geolocation) {  
    /* Prohlížeč podporuje Geolocation API a uživatel dal  
    svolení pro zjištění polohy */  
} else {  
    /* Prohlížeč nepodporuje Geolocation API nebo uživatel  
    nedal svolení pro zjištění polohy */  
}
```

Příklad 2 – Funkční zápis ověření podpory Geolocation API

```
if (navigator.geolocation) {  
    var options = {timeout:60000};  
    navigator.geolocation.getCurrentPosition(showLocation, errorHandler,  
    options);  
} else {  
    document.getElementById("error").innerHTML = "Podpora pro geolokaci  
    není dostupná."  
}
```

Na příkladu číslo 1 vidíme, jak správně zapsat podmínku pro ověření podpory Geolokace. Na příkladu číslo 2 je funkční zápis tohoto věření. V podmínce se kontroluje podpora Geolokace. Objekt navigator je objekt, který uchovává informace o prohlížeči, ze kterého se dotazujeme. Tento objekt obsahuje informace jako jsou: název prohlížeče, verze prohlížeče, pro jakou platformu je prohlížeč kompilován atd.⁶ Pokud tedy zjistíme, že prohlížeč podporuje Geolocation API a uživatel dal svolení ke zjištění jeho polohy, můžeme zavolat funkci pro zjišťování polohy - `getCurrentPosition`, která obsahuje 3 parametry (`showLocation`, `errorHandler`, `options`). O to celé se musí stát do 60000 milisekund, pokud se tak nestane automaticky je podmínka nesplněna, protože nebyl splněn časový limit. Pokud tedy nastane jedna z výše popsaných možností, tak do divu, který bude mít identifikátor „error“ vpíšeme hlášku: Podpora pro geolokaci není dostupná. Můžeme si vymyslet,

⁶Více na: http://www.w3schools.com/jsref/obj_navigator.asp

co chceme, můžeme použít objekt `alert` a vyhodit hlášku do message boxu, ale já jsem zvolil metodu zápisu do HTML stránky. Pokud nechceme použít pouze chybovou hlášku, ale chceme geolokaci simulovat můžeme použít polyfill, který nám nahradí nativní Geolocation API. Mezi tyto polyfilly patří například Geolocation polyfill⁷. Tato javascriptová knihovna nám zajistí funkčnost zjišťování polohy i v prohlížečích, které nativní HTML5 Geolocation API nepodporují. Je třeba ovšem počítat s tím, že funkčnost nemusí být stejná jako u HTML5 Geolocation API.

7.2 Získání polohy ve webovém prohlížeči

Po úspěšném ověření podpory HTML5 Geolocation API můžeme přejít do fáze získání aktuální polohy. O vyvolání požadavku na zjištění polohy se stará metoda `getCurrentPosition`. Tato metoda nám zajistí získání polohy od prohlížeče, který využije své dostupné zdroje. Metoda `getCurrentPosition` obsahuje 3 parametry. `showLocation`, `errorHandler` a `options`.

Příklad 3 – Zápis metody `getCurrentPosition()`

```
getCurrentPosition(showLocation, ErrorHandler, options);
```

⁷Podrobnosti: <https://github.com/inexorabletash/polyfill/blob/master/geo.js>

7.2.1 Parametr Showlocation

Parametr showlocation je callback metoda, které je předán objekt po zjištění polohy. Objekt, který je předán callbacku (v našem případě showLocation()), je potomek třídy, která implementuje rozhraní Position⁸. Rozhraní Position obsahuje dva atributy, které jsou pouze pro čtení. Tyto atributy jsou timestamp a coords. Atribut timestamp v sobě uchovává časový otisk zjištění aktuální polohy. Atribut coords implementuje další rozhraní, které nese název Coordinates. Rozhraní Coordinates obsahuje tyto atributy:

- Latitude
- Longitude
- Altitude
- Accuracy
- Altitude accuracy
- Heading
- Speed

Atribut Latitude nese informace o zeměpisné šířce, formát tohoto údaje je desetinné místo. Tudíž pokud chceme získat přesnou délku, je třeba tento údaj převést na minuty a sekundy. Stejně tak atribut Longitude, který nese informace o zeměpisné délce je třeba převést, pokud chceme získat přesný formát s minutami a sekundami. Další atribut je Altitude, tento atribut obsahuje údaj o nadmořské výšce zařízení, ze kterého se dotazujeme. Jestliže zařízení nemůže získat údaj o nadmořské výšce, tak atribut vrátí prázdnou hodnotu respektive hodnotu null. Atribut Accuracy nám udává přesnost zaměření zeměpisné šířky a délky v metrech. Altitude accuracy, jak už název napovídá, plní stejnou funkci jako Accuracy, ale Altitude accuracy udává

⁸Geolokace v prohlížeči — Zdroják. In: *Zdroják — o tvorbě webových stránek a aplikací* [online]. 2010 [cit. 2013-03-20]. Dostupné z: <http://www.zdrojak.cz/clanky/geolokace-v-prohlizeci/>

přesnost naměřené nadmořské výšky a hodnota je opět v metrech. Atribut Heading vrací údaj o směru natočení zařízení, ze kterého se dotazujeme. Získání údaje o natočení funguje tak, že nám vrátí hodnotu úhlu, o který je zařízení natočeno od severu, tedy pokud budeme natočeni na jih, hodnota pootočení bude 180°. Pokud zařízení nedokáže zjistit polohu otočení, nebo se jedná o statické zařízení, hodnota atributu bude null⁹. Posledním atributem rozhraní Coordinates je atribut speed. Tento atribut sleduje rychlost horizontálního pohybu zařízení. Tedy pokud se zařízení pohybuje, tak tento atribut nám vrátí hodnotu rychlosti pohybu v metrech za sekundu.

Příklad 4 – Jednoduchý výpis callback funkce showLocation

```
function showLocation(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    var altitude = position.coords.altitude;
    var accuracy = position.coords.accuracy;
    var altitudeAccuracy = position.coords.altitudeAccuracy;
    var heading = position.coords.heading;
    var speed = position.coords.speed;
    document.getElementById("result").innerHTML = "Šířka: " + latitude +
    "Délka: " + longitude + "Výška: " + altitude + "Přesnost: " + accuracy +
    "Přesnost Výšky: " + altitudeAccuracy + "Otočení: " + heading +
    "Rychlost: " + speed;
}
```

Funkce v příkladu 4 nám do elementu s identifikátorem result vypíše zjištěné hodnoty. Funkce nám zobrazí údaje o zeměpisné šířce a délce, nadmořské výšce, přesnosti naměřených údajů, pootočení zařízení vůči severu a rychlost pohybu zařízení.

⁹5.4 Coordinates interface. In: *Geolocation API Specification* [online]. 2012 [cit. 2013-03-20]. Dostupné z: <http://dev.w3.org/geo/api/spec-source.html>

7.2.2 Parametr ErrorHandler

Dalším parametrem metody `getCurrentPosition` je parametr `ErrorHandler`. Jedná se o callback metodu, která je zavolána pokud dojde k nějakému problému při zjišťování polohy zařízení. Jedná se o nepovinnou metodu. Tato metoda umožňuje zachytit a ošetřit chyby, které mohou při zjišťování polohy nastat. Je důležité tuto metodu správně a vždy použít, protože díky ní můžeme uživatele elegantně a srozumitelně informovat o vzniklé chybě při pokusu zjistit polohu zařízení. `ErrorHandler` implementuje rozhraní `PositionError`, které obsahuje dva argumenty, a to `code` s kódem chyby a `message` s detailním popisem chyby¹⁰. Rozhraní `PositionError` obsahuje tyto atributy:

- `PERMISSION_DENIED = 1`
- `POSITION_UNAVAILABLE = 2`
- `TIMEOUT = 3`
- `Code`
- `Message`

Atribut `code` musí vždy nést číselný údaj, který patří dané chybě, tedy prvním třem atributům. První atribut `PERMISSION_DENIED` znamená, že uživatel nedal svolení ke zjištění jeho polohy.

Druhý atribut `POSITION_UNAVAILABLE` znamená, že aktuální pozice je nedosažitelná, tedy že jí neleze zjistit. A poslední atribut `TIMEOUT` znamená, že zjišťování trvá příliš dlouho. Ošetření chyb můžeme jednoduše ošetřit za pomoci konstrukčního příkazu `SWITCH`.

¹⁰HTML5 – geolokační rozhraní. *Programujte.com - web o programování, webdesignu, počítačové grafice, databázích, elektrotechnice a designu* [online]. 2012 [cit. 2013-03-14]. Dostupné z: <http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/>

Příklad 5 – Jednoduchý výpis callback funkce ErrorHandler

```
function ErrorHandler(error) {
    switch (error.code) {
        case 1:
            document.getElementById("error").innerHTML =
                "Přístup byl odepřen!"
            break;
        case 2:
            document.getElementById("error").innerHTML =
                "Pozici neleze zjistit!"
            break;
        case 3:
            document.getElementById("error").innerHTML =
                "Vypršel časový limit žádosti!"
            break;
        default:
            alert("Chyba: " + error.message);
            break;
    }
}
```

Tato funkce se nám stará o obstarávání chyb. Při vzniku nějaké chyby uživatele informuje o tom, co se stalo. Při vzniku chyby `PERMISSION_DENIED` funkce vypíše do HTML elementu s ID `error` chybovou hlášku – Přístup byl odepřen! Pokud nastane chyba `POSITION_UNAVAILABLE` do stejného HTML elementu nám vypíše hlášku - Pozici nelze zjistit! A pokud bude zjišťování trvat nepřiměřeně dlouho, funkce nám vypíše - Vypršel časový limit žádosti!

7.2.3 Parametr Options

Dalším parametrem metody `getCurrentPosition` je parametr `option`. Tento parametr implementuje rozhraní `PositionOptions`, který nám umožňuje zvolit

si rozšířené nastavení geolokace ve webovém prohlížeči. Rozhraní `PositionOptions` obsahuje tyto rozšiřující atributy nastavení:

- `enableHighAccuracy`
- `timeout`
- `maximumAge`

Atribut `enableHighAccuracy` je typu `boolean` a jestliže mu nastavíme hodnotu `TRUE`, tak při zjišťování polohy se bude aplikace vždy snažit o využití nejpřesnějšího zdroje pro zjištění polohy – GPS modul, pokud bude k dispozici. Atributem `timeout` můžeme nastavit hodnotu času v milisekundách pro zjištění polohy. Pokud tento časový limit překročíme, vyvolá se chyba `TIMEOUT`. Poslední atribut `maximumAge` nese časový údaj opět v milisekundách o tom, jak starou informaci o poloze může prohlížeč použít z cache paměti prohlížeče. Použití parametru `options` demonstruje následující příklad.

Příklad 6 – Použití parametru `options`

```
getCurrentPosition (showLocation, ErrorHandler,  
{ enableHighAccuracy: true, timeout: 500000, maximumAge: 0 });
```

7.2.4 Ukázkový příklad jednorázového získání polohy

Všechno nejlépe vysvětlí ukázkový příklad. Pro tvorbu ukázkových příkladů jsem se rozhodl použít front-end responzivní framework `Foundation`¹¹. Jedná se o připravenou šablonu `HTML5` webové stránky. `Foundation` ušetří při vývoji spoustu práce, protože obsahuje připravenou stromovou strukturu webové stránky. Dále obsahuje připravené styly pro globální `HTML` elementy, knihovnu `jQuery` a samozřejmě také obsahuje styl `normalize.css` – moderní náhrada za `CSS reset`. Dále v sobě implementuje knihovnu `modernizr`, kterou jsem již zmiňoval a spoustu dalších podpůrných funkcí, vlastností a knihoven.

¹¹Ke stažení: <http://foundation.zurb.com/>

Příklad jednorázového zjištění polohy (příklad-1_Jednorazove-zjistení)

```
<script type="text/javascript">
var map;

// Zjištění polohy =====
function showLocation(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    var accuracy = position.coords.accuracy;
    var altitude = position.coords.altitude;
    var altitudeAccuracy = position.coords.altitudeAccuracy;
    var heading = position.coords.heading;
    var speed = position.coords.speed;
    document.getElementById("result").innerHTML =
    "Šířka: " + latitude +
    "<br/>Délka: " + longitude +
    "<br/>Výška: " + altitude +
    "<br/>Přesnost: " + accuracy +
    "<br/>Přesnost Výšky: " + altitudeAccuracy +
    "<br/>Otočení:" + heading +
    "<br/>Rychlost: " + speed;
}

// Ošetření chyb =====
function errorHandler(error) {
    switch (error.code) {
        case 1:
            document.getElementById("error").innerHTML =
            "Chyba: Přístup byl odepřen.";
            break;
        case 2:
```

```

        document.getElementById("error").innerHTML =
            "Chyba: Poloha není k dispozici.";
        break;
        case 3:
            document.getElementById("error").innerHTML =
                "Chyba: Vypršel časový limit.";
            break;
        default:
            document.getElementById("error").innerHTML =
                "Chyba: " + error.message;
            break;
    }
}

// Volání polohy =====
function getLocation(){
    if (navigator.geolocation) {
        var options = maximumAge:600000, timeout:15000,
            enableHighAccuracy: true;
        navigator.geolocation.getCurrentPosition(
            showLocation, errorHandler, options);
    } else {
        document.getElementById("error").innerHTML =
            "Podpora pro geolokaci není dostupná.";
    }
}

</script>
<body onload="getLocation()">

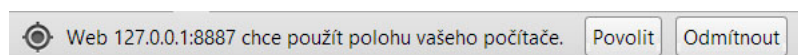
<div class="row">
<div class="large-12 columns">
<h2>Příklad 1 - Jednorázové zjištění polohy</h2>

```

```
<p class="small">Geolokace ve webovém  
prohlížeči pomocí HTML5 Geolocation API</p>  
<hr />  
</div>  
</div>  
  
<div class="row">  
<div class="large-12 columns">  
    <p id="result"></p>  
    <p id="error"></p>  
<a href="" class="button">Zpět</a>  
</div>  
</div>  
</body>
```

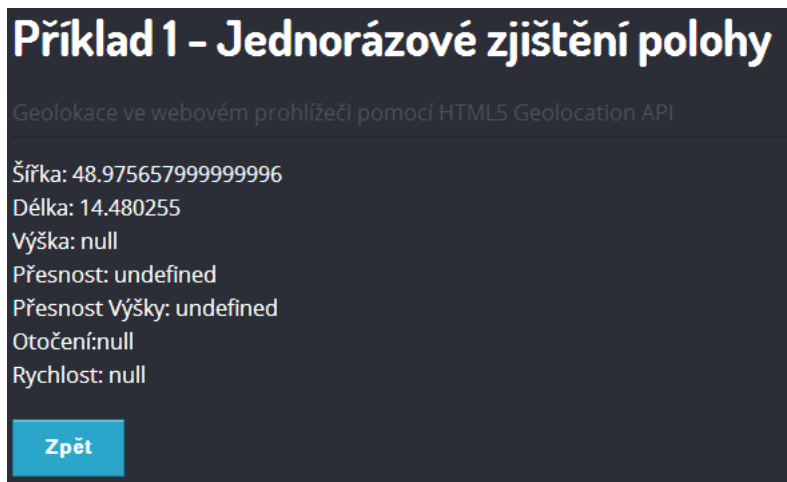
Tento příklad demonstruje jednorázové zjištění polohy ve webovém prohlížeči. Jedná se o nejzákladnější ukázkou geolokace ve webovém prohlížeči. Příklad zajišťuje zjištění polohy a její výpis do těla dokumentu. V případě výskytu chyby se vypíše popis chyby.

Po spuštění příkladu musíme potvrdit, že souhlasíme se zjištěním naší polohy.



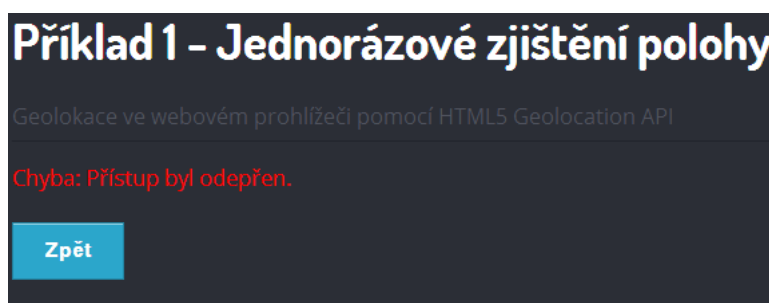
Obrázek 1: Požadavek prohlížeče o povolení zjištění vaší polohy

Po kliknutí na tlačítko se nám zobrazí výsledek zjišťování polohy.



Obrázek 2: Výpis výsledku zjišťování polohy

Jelikož jsem testoval tento příklad na stolním počítači bez GPS modulu, tak se nepodařilo zjistit nadmořskou výšku, přesnost, přesnost nadmořské výšky, otočení a rychlost. Ale zeměpisnou šířku a délku se podařilo získat bez problému. Pokud by se vyskytla jakákoliv chyba, aplikace nám vypíše zprávu o chybě.



Obrázek 3: Chybové hlášení

Zde se nepodařilo získat autorizaci ke zjištění polohy, proto aplikace vyhodila chybovou hlášku – Přístup byl odepřen.

7.3 Periodické získávání polohy ve webovém prohlížeči

Polohu zařízení nemusíme zjišťovat pouze jednorázově, ale díky metodě `watchPosition()` můžeme polohu zaznamenávat i v periodických intervalech. Tato metoda plní stejnou funkci jako metoda `getCurrentPosition` jen s tím rozdílem, že aktuální polohu získává pravidelně. Zápis metody je stejný je jako u metody `getCurrentPosition`.

Příklad 7 – Zápis metody `watchPosition()`

```
watchPosition (showLocation, ErrorHandler, options);
```

Použití je tedy stejné jako u metody `getCurrentPosition()`. Jediný rozdíl je v tom, že metoda `watchPosition` nám jako hodnotu vrací identifikátor požadavku, kvůli pozdějšímu zrušení.¹²

Postup tvorby aplikace, která nám bude periodicky zjišťovat polohu zařízení je podobný jako při tvorbě aplikace pro jednorázové zjišťování. Je tam jen pár věcí navíc. Jako první věc přidáme do těla dokumentu tlačítko, které bude zapínat zjišťování.

```
<input type="button" onclick="getLocationUpdate()" value="Start">
```

¹²HTML5 – geolokační rozhraní. *Programujte.com - web o programování, webdesignu, počítačové grafice, databázích, elektrotechnice a designu* [online]. 2012 [cit. 2013-03-14]. Dostupné z: <http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/>

Na začátku bloku s javascriptem definujeme dvě proměnné watchID a navGeo.

```
<script type="text/javascript">
var watchID;
var navGeo;
```

A pak pouze změníme funkci getLocationUpdate, která by měla vypadat takto:

```
function getLocationUpdate() {
  if (navigator.geolocation) {
    navGeo = navigator.geolocation;
    watchID = navGeo.watchPosition(showLocation);
  }else{
    document.getElementById("error").innerHTML =
      "Podpora pro geolokaci není dostupná.";
  }
}
```

8 HTML5 Geolocation API s různými mapovými podklady

HTML5 geolocation API samo o sobě toho moc nedokáže. Ale teprve s použitím mapových podkladů dostává toto API úplně jiný rozměr. Protože dokážeme zjištěné výsledky promítnout do mapových podkladů a zobrazit tak mnohem přívětivější informace než pouze zeměpisné souřadnice. Další výhodou je možnost použití samotného API podkladových map. Můžeme tedy aplikace přizpůsobit přesně našim potřebám.

Práce s HTML5 Geolocation API a mapovými podklady je velice snadná. Stačí nám pouze webová stránka, která bude obsahovat blok s názvem *map_canvas*. Název si samozřejmě můžeme zvolit jaký chceme.

```
<div id="map_canvas"></div>
```

Do tohoto bloku budeme nahrávat mapu a vypisovat získané údaje. Tento blok samozřejmě můžeme pomocí CSS stylovat. Můžeme mu například dát určitou šířku a délku, barvu a velikost rámečku, nebo nějakou CSS3 vlastnost jako například vržený stín apod.

Jako mapové podklady na kterých budu demonstrovat použití s HTML5 Geolocation API jsem vybral následující:

- Google maps
- Mapy.cz
- Bing Maps
- OpenLayers

Jedná se o výběr neznámějších dostupných mapových podkladů, které se dají na internetu nalézt. Veškerý problém spočívá pouze v tom, že ve funkci

showLocation přidáme kód, který nám vytvoří, nastaví a zobrazí mapu. A pka už pouze do pozice, kterou nám HTML5 Geolocation vrátí, vypíšeme grafický ukazatel.

8.1 HTML5 Geolocation API a Google maps

Začneme tedy nejrozšířenějšími mapovými podklady na internetu a to Google maps. Veškeré informace o Google maps API nalezneme zde:

<https://developers.google.com/maps/documentation/javascript/?hl=cs>

Nesmíme zapomenout nalinkovat Google maps API do naší stránky:

```
<script src="https://maps.googleapis.com/maps/api/
js?v=3.exp&sensor=false"></script>
```

Poté do funkce showLocation přidáme následující řádky kódu:

```
var myLatLng = new google.maps.LatLng(latitude, longitude);

    var mapOptions =
        zoom: 16,
        center: myLatLng,
        mapTypeId: google.maps.MapTypeId.HYBRID

var map = new google.maps.Map(document.getElementById
("map_canvas"), mapOptions);

var marker = new google.maps.Marker(
    position: myLatLng,
    map: map,
    title:"Hello World!");
```


V prvním kroku přebereme údaje naměřené HTML5 Geolocation API a uložíme si je do proměnné. V dalším kroku nastavíme samotnou mapu. Zoom znamená přiblížení mapy, center nám určuje, kam se umístí střed mapy, v našem případě se mapa vycentruje na souřadnice naměřené Geolocation API. Poslední parametr nám určuje typ mapy. Další možnosti lze nalézt v dokumentaci. V dalším kroku „vepíšeme“ mapu do našeho oddílu, který jsme si nazvali map_canvas. A v poslední části kódu akorát vytvoříme grafický ukazatel pozice, který umístíme na naměřené souřadnice. Výsledkem je tedy mapa s ukazatelem naší pozice.¹³

8.2 HTML5 Geolocation API a Mapy.cz

Dalšími mapovými podklady jsou v Čechách dobře známé Mapy.cz. Dokumentace je k nalezení zde:

<http://api4.mapy.cz/>

Opět musíme nalinkovat toto API do naší stránky:

```
<script src="http://api4.mapy.cz/loader.js"></script>
<script>Loader.load()</script>
```

Funkce showLocation pak bude vypadat takto:

```
var lonD = longitude;
var latD = latitude;
var center = SMap.Coords.fromWGS84(lonD, latD);
var m = new SMap(JAK.gel("map_canvas"), center, 14);

    m.addDefaultLayer(SMap.DEF_OPHOTO);
    m.addDefaultLayer(SMap.DEF_OPHOTO0203);
    m.addDefaultLayer(SMap.DEF_OPHOTO0406);
    m.addDefaultLayer(SMap.DEF_TURIST);
    m.addDefaultLayer(SMap.DEF_HISTORIC);
```

¹³Praktická ukázka: <http://bp.martinlaudat.cz/priklady/3-google-maps-api/>

```

m.addDefaultLayer(SMap.DEF_BASE).enable();
m.addDefaultControls();

var layerSwitch = new SMap.Control.Layer();
layerSwitch.addDefaultLayer(SMap.DEF_BASE);
layerSwitch.addDefaultLayer(SMap.DEF_OPHOTO);
layerSwitch.addDefaultLayer(SMap.DEF_TURIST);
layerSwitch.addDefaultLayer(SMap.DEF_OPHOTO0406);
layerSwitch.addDefaultLayer(SMap.DEF_OPHOTO0203);
layerSwitch.addDefaultLayer(SMap.DEF_HISTORIC);
m.addControl(layerSwitch, left:"8px", top:"9px");

var layer = new SMap.Layer.Marker();
m.addLayer(layer);
layer.enable();

var options = ;
var marker = new SMap.Marker(center, "myMarker", options);
layer.addMarker(marker)

```

Princip je úplně stejný jako u Google maps. Opět předáme naměřené hodnoty do proměných, vycentrujeme mapu přesně na naměřenou pozici. Vytvoříme mapu s individuálním nastavením a vytvoříme a umístíme grafický ukazatel na naměřené souřadnice. Na první pohled je vidět, že kód je delší než u Google map, ale je to úplně stejný postup jen se liší zápis kvůli odlišnosti Google maps API a Mapy.cz API. Opět by měla být výsledkem mapa se ukazatelem naší pozice.¹⁴

8.3 HTML5 Geolocation API a Bing maps

Předposlední ukázkou je spojení HTML5 Geolocation API s mapovými podklady od Microsoftu – tedy mapami Bing maps. Dokumentace Bing maps je zde: <http://msdn.microsoft.com/en-us/library/dd877180.aspx>

¹⁴Praktická ukázka: <http://bp.martinlaudat.cz/priklady/4-mapy.cz-api/>

Jako v předchozích příkladech i zde musíme nalinkovat API:

```
<script src="http://ecn.dev.virtualearth.net/mapcontrol/  
mapcontrol.ashx?v=6.3"></script>
```

Kód ve funkci showLocation bude vypadat následovně:

```
var lon = longitude;  
var lat = latitude;  
var map = new OpenLayers.Map('map_canvas');  
  
    map.addLayer(new OpenLayers.Layer.OSM());  
    var lonLat = new OpenLayers.LonLat( lon ,lat )  
        .transform(  
            new OpenLayers.Projection("EPSG:4326"), //převod  
            souřadnicových soustav pro pozdější použití  
            map.getProjectionObject()  
        );  
  
    var markers = new OpenLayers.Layer.Markers( "Markers" );  
    map.addLayer(markers);  
  
    markers.addMarker(new OpenLayers.Marker(lonLat));  
  
    map.setCenter (lonLat, 16);  
}
```

Jak je vidět, tak princip nastavení je opět velice podobný předchozím příkladům. Jediné co trochu jiné je to, že musíme převést získané souřadnice do srozumitelnějšího formátu pro OpenLayers. Tento příklad by měl opět zobrazit mapu s grafickým ukazatelem naší pozice.¹⁵

¹⁵Praktická ukázka: <http://bp.martinlaudat.cz/priklady/6-openlayers-api/>

Jak je vidět na všech příkladech, práce s mapovými podklady je velice jednoduchá a princip je u všech zmíněných podkladů stejný. Díky tomu je pouze na nás jaké mapové podklady zvolíme. Rozhoduje tedy spíše API samotných mapových podkladů. Protože ve většině případů budeme potřebovat udělat aplikaci složitější než jenom výpis naší polohy a proto budou rozhodující možnosti určitého API, které mapové podklady nabízí.

9 Metody lokalizace mobilních zařízení

Pokud chceme tvořit aplikace, které budou zjišťovat polohu zařízení. Musíme nejdříve zjistit polohu zařízení – kde se právě nachází. Bez těchto údajů nelze dále pracovat s HTML5 Geolocation API. Je hned několik možností jak zjistit aktuální polohu zařízení. Mezi ty hlavní patří zjišťování polohy pomocí GPS hardware, zjištění polohy pomocí IP adresy zařízení a zjištění polohy pomocí datových sítí – WIFI a GSM sítě. Nejpřesnější výsledky získáme se zařízeními, které disponuje GPS hardwarem. V dnešní době je to většina chytrých telefonů a tabletů. Pokud ovšem zařízení nedisponuje GPS hardwarem, musíme se pokusit o zjištění pomocí alternativních metod. Vyhledání zařízení pomocí datových sítí je relativně přesné hlavně v oblasti s vysokou silou signálu a to až už WIFI signálu nebo signálu GSM sítí. Zjišťování polohy pomocí IP adresy je nejméně přesné, protože záleží na vašem typu připojení. Může se tedy stát, že místo vaší IP adresy se bude zobrazovat IP adresa poskytovatele vašeho připojení.

Je důležité ještě zmínit, že samotné Geolocation API se nestará o zjišťování polohy zařízení. Geolocation API pouze požádá prohlížeč a zjištění polohy a zbytek je na prohlížeči. Prohlížeč sám určí nejpřesnější zdroj lokace, který má k dispozici a ten využije pro zjištění polohy. Zjištěné údaje pak pouze předá Geolocation API kde s nimi můžeme pracovat dál.

9.1 Lokalizace pomocí GPS Hardware

Jedná se o nejpřesnější metodu zjištění polohy. Tato metoda funguje na principu komunikace zařízení se satelity, které obíhají kolem země. Jedná se o soustavu 27 družic z toho 3 jsou záložní pro případ poruchy některé z družic. Družice obíhají kolem Země v šesti různých dráhách, ve výšce zhruba 20 000km. Každá družice oběhne zemi dvakrát denně (jeden oběh trvá 11 h 58 min)¹⁶.

¹⁶Jak funguje GPS?. In: *Svět hardware* [online]. 2006 [cit. 2013-06-05]. Dostupné z: <http://www.svethardware.cz/jak-funguje-gps/21826-4>

K určení polohy na Zemi nám stačí signál ze tří družic. Pokud chceme zjišťovat i nadmořskou výšku budeme potřebovat signál alespoň ze čtyř družic. Logicky platí pravidlo, že čím více družic se zařízením komunikuje tím přesněji se poloha určí.

Každá družice vysílá elektromagnetický signál – mikrovlnný paprsek, který oznamuje její přítomnost každému, kdo má přijímač schopný tento signál přijmout.¹⁷ Díky vzdálenosti družic existuje malé zpoždění od odeslání signálu družicí po příjem signálu zařízením. Díky tomuto zpoždění dokážeme určit vzdálenost zařízení od družice, pokud tyto zpoždění spočítáme alespoň ze tří družic, tak získáme údaje zeměpisné polohy (šířka a délka) s přesností na několik metrů.

GPS přijímače se často označují jako pasivní. To proto, že tyto přijímače pouze přijímají signál z družic a zpět družicím nic neodesílají. GPS se dnes používá v mnoha oborech lidské činnosti. Například v letectví, dopravě, zemědělství nebo například pro vojenské účely. Ale oboru, kde se GPS používá je samozřejmě mnohem více a další budou přibývat. A také díky moderním informačním technologiím můžeme my obyčejní lidé používat GPS pro běžné úkony jako je například automobilová navigace.

9.2 Lokalizace pomocí IP adresy

Lokalizace pomocí IP adresy je nejméně přesná metoda lokalizace, protože se nezakládá na určení polohy, ale pouze zjišťuje údaje o držiteli IP adresy. Princip je takový, že po získání IP adresy se dotazujeme na databázi IP adres, komu testovaná adresa patří. Nejznámější takovou databází je databáze WHOIS¹⁸. Tato data báze je nejznámější a nejvíce rozšířená. Existují i další ale většinou jsou placené. Dále se můžeme dotazovat na záznamy z DNS serverů nebo informace o ISP poskytovatele našeho připojení. Největší nevýhoda je, že pokud nemáme statickou IP adresu, ale na internet

¹⁷O technologii GPS. In: *Technologie Mio* [online]. 2012 [cit. 2013-06-05]. Dostupné z: <http://eu.mio.com/cs.cz/global-positioning-system-jak-gps-funguje.htm>

¹⁸Adresa databáze: <http://whois.net/>

se připojujeme přes IP adresu (bránu) poskytovatele připojení, tak výsledky vyhledávání budou směřovat právě na poskytovatele a ne na nás. Přesnost se dá těžko určit, ale většinou lokalizace pomocí IP adresy trefí, alespoň stát, ze kterého se dotazujeme.

9.3 Lokalizace pomocí datových sítí

Dalším způsobem jak lokalizovat zařízení je lokalizace pomocí datových sítí. Jedná se o lokalizaci pomocí WIFI sítí a GSM. Lokalizace pomocí WIFI sítí probíhá tak, že se otiskne seznam všech viditelných hotspotů, které se nacházejí v okolí zařízení. Tyto informace se porovnají s databází a na základě shody MAC adres sítí se zjistí pozice zařízení. Na internetu je hned několik databází hotspotů, ale nejrozsáhlejší vytvořil Google. Velice záleží na počtu sítí v okolí a na síle signálu těchto sítí. Čím více bude v okolí sítí s dobrým signálem tím přesnější výsledky získáme.

Mimo WIFI sítí můžeme lokalizovat zařízení i pomocí GSM sítí. Ovšem hlavní podmínkou je, že zařízení ze kterého se dotazujeme má potřebný hardware pro podporu GSM sítí. Jedná se tedy hlavně o mobilní telefony a některé tablety které nativně GSM sítě podporují. Jsou tři základní způsoby zjištění polohy zařízení v síti GSM. První metodou je tzv. CGI (Cell Global Identity) tento způsob využívá znalost operátora o rozmístění jejich vysílačů, tedy BTS stanic. Operátor, který GSM síť provozuje zná přesné polohy svých BTS a díky tomu dokáže určit polohu zařízení. Bohužel je tato metoda nepřesná.



Obrázek 4: Zjištění polohy zařízení metodou CGI

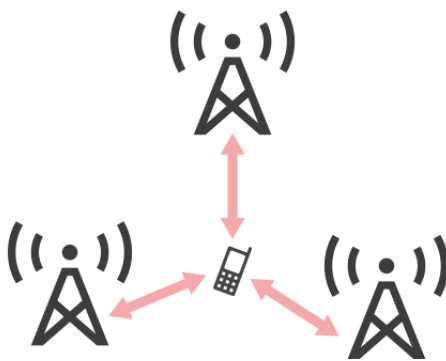
Jak je vidět na obrázku číslo 4 tato metoda zjištění polohy nám určí pouze oblast ve které se zařízení nachází. Proto se nejedná o přesnou metodu.

Další a mnohem přesnější metodou je metoda CGI+TA (Cell Global Identity + Timing Advance), která jak již název napovídá používá pro zjištění polohy ještě jeden parametr - TA. Tedy Timing advance, což je časový údaj zpoždění signálu mezi zařízením a BTS stanicí. Díky tomuto parametru dokážeme polohu určit mnohem přesněji, protože dokážeme určit i vzdálenost zařízení od BTS a to dokonce s přesností zhruba 550m.



Obrázek 5: Zjištění polohy zařízení metodou CGI+TA

Poslední metodou o které bych se chtěl zmínit je metoda UL-TOA (Uplink Time of Arrival). Tato metoda zjišťuje pouze časový interval komunikace zařízení se sítí. Tedy časový rozdíl, kdy zařízení začalo vysílat a kdy dorazil signál k vysílači. Pokud je zařízení v dostupnosti alespoň tří vysílačů, můžeme určit přesnost zhruba na 150 metrů. Jak to zhruba probíhá je vidět na obrázku číslo 6.



Obrázek 6: Zjištění polohy zařízení metodou UL-TOA

Je ovšem třeba zdůraznit, že přesného zaměření dosáhneme pouze v ideálních podmínkách. Přesnost měření například ovlivňují odrazy nebo zalomení signálu. Proto tato metoda není moc vhodná v městském prostředí.

10 Podpora HTML5 Geolocation API v prohlížečích

Při tvorbě každé webové aplikace nebo webové stránky je důležité myslet na to, že ne všichni používají stejný operační systém, stejný prohlížeč nebo stejné rozlišení. Proto musíme dbát na to, aby to co vytvoříme bylo funkční na většině zařízení, ze kterých mohou uživatelé na web přistupovat. Pro nás a je nejzásadnější podpora prohlížečů. Jelikož je Geolocation API a HTML5 relativně mladá technologie, tak je možné, že nebude fungovat ve všech prohlížečích - respektive ve starších prohlížečích.

Dnes se na trhu webových prohlížečů pohybuje několik velkých hráčů a kterých se má smysl bavit. Jedná se Google Chrome, Mozilla Firefox, Opera, Safari a Internet Explorer. Dále tyto prohlížeče můžeme rozdělit na moderní a zastaralé. Do kategorie moderních prohlížečů patří pouze poslední verze zmíněných prohlížečů a do kategorie zastaralých logicky prohlížeče neaktuální.

Nejdříve popíšu podporu v moderních prohlížečích. HTML5 Geolocation API je podporováno ve všech moderních prohlížečích. Testování jsem provedl v následujících verzích, které jsou právě aktuální.

- Google Chrome 27
- Mozilla Firefox 21
- Opera 12.15
- Safari 6.0.5
- Internet Explorer 10

Ve všech těchto prohlížečích byla funkčnost naprosto bezproblémová. Dá se tedy říci, že podpora geolokace je velmi dobrá.

Dá se tedy předpokládat, že v budoucích verzích prohlížečů bude podpora také, pokud se výrobci prohlížečů z nějakých důvodů nerozhodnou geolokaci odebrat, ale to je velice nepravděpodobné.

A jaká je podpora v zastaralých verzích prohlížečů? Rozhodl jsem se testovat pouze starší verze Internet Exploreru, protože si díky automatickým aktualizacím není třeba testovat ostatní prohlížeče. Je jasné že někteří uživatelé mají automatické aktualizace vyplé, ale to je tak zanedbatelná část, že není třeba se tím zabývat. Pro nás tedy bude rozhodující Internet Explorer jehož starší verze se stále ještě hojně používají. Je to logické, protože většina uživatelů nepotřebuje jiný prohlížeč, nebo z nějakých důvodů nemůže svůj prohlížeč aktualizovat. A proto se dnes na počítačích stále vyskytují verze Internet Exploreru 7 nebo 8. Dle serveru statcounter.com¹⁹ je podíl Internet Exploreru 8 i dnes uctihodných 11,83% a podíl Internet Exploreru 9 16,14%. Je tedy třeba se těmito prohlížeči i nadále zabývat a připravovat webové stránky nebo aplikace i pro tyto prohlížeče.

Provedl jsem tedy testy v Internet Exploreru 8 a 9. Ve verzi 9 funguje geolokace bez jakýchkoliv problémů. V Internet Exploreru 8 už bohužel tak pozitivní výsledky nejsou, protože zde geolokace nefunguje vůbec. Takže pro Internet Explorer 8 je nutno používat nějakou simulaci - polyfill. Můžeme zvolit například polyfill Geo.js²⁰. Jedná se o javascriptovou knihovnu, která nám simuluje HTML5 Geolocation API, je jasné že funkčnost nebude úplně stejná jako s použitím samotného Geolocation API, ale ve většině případů nám to může pomoci vytvořit aplikace funkční i v Internet Exploreru 8 a v případně verzích.

¹⁹Veškeré statistiky: gs.statcounter.com

²⁰Stážení a informace: <https://github.com/inexorabletash/polyfill/blob/master/geo.js>

11 Využití geolokace na internetu

Tato kapitola se bude zabývat možnostmi využití geolokace na internetu, které ukáží na reálných službách. S příchodem moderních technologií pro tvorbu webových stránek se rozšířili možnosti tvorby webových aplikací a proto vzniklo spousta nových aplikací a služeb. Vývojáři si dobře uvědomují sílu nových možností které mají. A právě díky těmto vývojářům vznikají aplikace o kterých se nám před několika lety ani nezdálo. A vývoj webových technologií jde stále dopředu a proto budou dále vznikat nové a modernější aplikace než doposud. Mezi dnešní moderní aplikace se dají zařadit sociální sítě, aplikace nahrazující desktopové programy, různá webová úložiště a v neposlední řadě několik služeb využívajících geolokaci. Je to důsledkem toho, že si spousta lidí uvědomila, že dokážou vytvořit aplikace, které budou užitečné a dokážou generovat nějaké zisky. Samozřejmě ne všichni to dělají pro peníze, ale ti největší hráči na "dnešním" webu - sociální sítě a různé služby - generují miliardové zisky. Je tedy zřejmé, že tvořit moderní aplikace má v dnešní době opravdu smysl.

Jako další využití geolokace na internetu vydím usnadnění uživatelského zážitku z webu. Dnešní webové stránky a aplikace by měly být tzv. user-centered, tedy takové aplikace, které se uživateli dobře ovládají a uživatel z nich má dobrý pocit. Rozepsat všechny kroky návrhu kvalitního rozhraní a uživatelského zážitku by zaplnilo minimálně ještě jednu bakalářskou práci. Jedná se totiž o několik komplikovaných kroků, které ovšem přinášejí výsledky, pokud je dokážeme správně využít. Tyto kroky dnes bezpodmínečně patří do moderního workflow vytváření webových aplikací. HTML5 Geolocation API nám při vytváření kvalitního uživatelského zážitku může velice pomoci, ale také uškodit. Pokud se nedokážeme zamyslet nad uživatelskými potřebami, pak nám tato technologie může uškodit. V opačném případě nám může velice pomoci. Díky této technologii můžeme uživateli zobrazit spousta informací aniž by musel odejít z webu na jiný web. Například můžeme přímo ve webové stránce vygenerovat trasu od uživatele do cíle (např. sídlo firmy)

a to vše na našem webu a na všech zařízeních. Díky tomuto získá uživatel okamžitě výsledek jeho interakce a hned se dozví co potřebuje aniž by opustil web. A toto byl pouze jeden příklad z mála, které uživateli dokážou pomoci. Díky napojení na mapové podklady dokážeme vytvořit aplikace se kterými se bude uživateli snadno pracovat. A je jenom na nás, aby jsme toto všechno dokázali využít v náš prospěch.

V dnešní době je jeden z největší zisků na internetu z reklam. Reklamy jsou všude a dokonce je dnes více reklam než prostoru kde mohou být zveřejněny. V reklamách na internetu se točí obrovské množství peněz a proto je každá dobře cílená reklama velmi cenná. Díky HTML5 Geolocation API můžeme vytvořit reklamní systém, který bude zobrazovat reklamy na základě pozice zařízení. Můžeme přesně určit polohu uživatele a podstrčit mu reklamy služeb či podniků z jeho bezprostředního okolí. Toto řešení má určitě mnohem větší smysl a bude mnohem účinnější než dosavadní reklamní systémy. Ikdyž například Google Adwords takto pracují také.

Cílem této práce je rozebrat a vysvětlit problematiku HTML5 Geolocation API, proto nás budou zajímat aplikace které tuto technologii využívají. Takových to aplikací je spousta, ale já jsem se snažil vybrat ty nejdůležitější a nejpoužívanější v České Republice. Vybral jsem tedy z českých zástupců službu Dáme jídlo, a ze zahraničních populární sociální síť Foursquare a ještě službu TripAdvisor. A ještě jednu hru která je založená na geolokaci - Geocaching. Jak jsem již zmiňoval, aplikací a služeb je na internetu spousta, ale já se pokusil vybrat nejpoužívanější. Dále bych ještě rád zmínil hru od společnosti Google, tato hra se jmenuje Ingress²¹. Tato hra zatím ještě není veřejně dostupná, ale mělo by se jednat o zajímavou geolokační hru o objevování portálu. Dle prvních recenzí si trůfám říct, že něco takového dosud neexistuje a rozhodně se bude jednat o zajímavou hru.

²¹více na: <http://www.ingress.com>

11.1 Dáme jídlo

Dáme jídlo (dříve PizzaTime.cz) je služba za kterou stojí Tomáš Čupr, který také založil Slevomat. Jedná se o agregátor dovozu hotových jídel. Tedy tato služba agreguje restaurace a zprostředkovává rozvoz jídel k zákazníkům. Díky tomu tedy nemusím složitě prohledávat webové stránky restaurací, ale mám to pěkně všechno na jednom místě. Samozřejmě jsou mobilní aplikace pro všechny dnešní platformy. A právě v těchto mobilních aplikacích se používá geolokace, která zajistí to, že pro spuštění aplikace se zobrazí nabídky restaurací v okolí zařízení ze kterého se dotazujeme. To tedy znamená že se knám dostanou pouze restaurace v okolí a tudíž není potřeba filtrovat ze všech výsledků jen ty restaurace, které jsou v blízkosti zařízení. Tato služba nabízí i další funkce, jako hodnocení, sbírání kreditů atd. Ale pro účel této práce je důležitý pouze smysl geolokace, kterou aplikace používá. Více podrobností lze dohledat na oficiálních stránkách - <http://www.damejidlo.cz>

11.2 Foursquare

Foursquare je asi neznámější služba, která využívá geolokaci. Jedná se o sociální síť, která je založená na zveřejňování polohy. Tím že zveřejníte svou polohu (tzv. check-in) na určitém místě získáte body. Pokud například některé místo navštívíte několikrát můžete odemknout jednu z tzv. badges tedy odznak. Například odznak za navštívení místa na kterém jste ještě nikdy nebyl. A pokud chodíte na nějaké místo často, například do oblíbeného klubu, a pravidelně se tam checkujete, tak pokud budete mít nejvíc check-inů z osob který ten klub navštěvují, tak se tam stanete starostou. Tedy osobou která tam chodí nejčastěji a budete zveřejněni u profilu podniku. Můžete si také přidat přátele, kteří foursquare rovněž používají a soutěžit s nimi o to, kdo nasbírá co nejvíce bodů. Foursquare je samozřejmě také dostupný jako aplikace pro všechny moderní platformy. Oficiální web je zde - <http://www.foursquare.com>

11.3 TripAdvisor

Tripadvisor je služba především pro lidi, kteří často cestují. Díky této aplikaci lze dohledat recenze na hotely, restaurace nebo třeba letoviska. Díky dostupnosti mobilní aplikace na všechny dnešní mobilní platformy se z této služby stává ideální společník na cesty. Právě v mobilních aplikacích díky geolokaci můžeme rychle zjistit recenze či názory jiných lidí na služby či restaurace v našem okolí. Bohužel tato služba není moc známa v České Republice a recenze na české podniky jsou napsány především v anglickém jazyce. Ale při cestách do zahraničí nám může být tato služba velice užitečná a může nám ušetřit i nějaké finance. Všechny informace naleznete na oficiálních stránkách - <http://www.tripadvisor.com>

11.4 Geocaching

Geocaching je turistická hra, kde je hlavním cílem najít poklady (tzv. kešky). Princip je vtom, že si na webu vyhledáme souřadnice nějaké kešky, pomocí navigace se dostaneme na toto místo a zkusíme najít kešku. Většinou bývají kešky dobře schované a není úplně jednoduchá je najít. Po nalezení můžeme vyměnit obsah kešky a zapsat se do logbooku - deníku nálezců, který často bývá součástí kešky. Pak opět kešku znova schováme pro další hledače. Každý může kešky vytvářet po vytvoření a schování kešky autor zveřejní její polohu na jeden z mnoha serverů zabývajících se geocachingem, kde si ostatní dohledají souřadnice. I v tomto případě existují mobilní aplikace pro všechny moderní platformy a díky nim lze hledat kešky nedaleko zařízení na kterém aplikaci spustíme. Podrobnosti lze najít na oficiálních webových stránkách - <http://www.geocaching.com>

11.5 Různé sociální sítě

Nelze opomenout zjišťování polohy na různých sociálních sítích jako je Facebook nebo Twitter. Na těchto sítích můžeme u svých příspěvků zveřejnit svojí polohu.

12 Nebezpečí zveřejňování své polohy na internetu

Geolokace na internetu nám v mnoha případech dokáže usnadnit život a to jak soukromý tak i profesionální. Ale geolokace nám dokáže také uškodit, díky tomu, že zveřejňujeme svojí polohu na internetu, zveřejňujeme i svoje soukromí s přesností na metry. Mnoho uživatelů nad tímto vůbec nepřemýšlí a bezmyšlenkovitě svojí polohu zveřejňují na internetu. Toto chování může způsobit celou řadu problémů a vystavit uživatele do nebezpečí.

Svojí polohu nezveřejňujeme pouze na sociálních sítích, ale je třeba si uvědomit že GPS informace se dnes ukládají i k fotografiím. Tato funkce může být užitečná pro pozdější zorientování v pořízených fotkách, ale také nevědomky prozradí naši polohu. Dnešní chytré telefony obsahují kvalitní fotoaparáty a ve většině případu tyto telefony automaticky ukládají GPS informace k pořízeným fotografiím. Poté stačí pouze takovouto fotku zveřejnit na veřejnou fotogalerii nebo sociální síť a každý může mít přístup k přesné poloze, kde byla fotografie pořízena a kde se právě nachází uživatel, který ji pořídil. Je znám případ amerického vojáka, který vyfotil své kolegy na hlídce v Afghánistánu a okamžitě tyto fotky zveřejnil na webové galerii Google Picasa. U těchto fotografií, které byly veřejně dostupné byly uloženy i GPS informace o pozici.

Stačilo jen, aby tyto fotografie našel nepřítel na internetu a okamžitě věděl přesnou pozici této hlídky a tím se dostal do značné výhody. Zde se jednalo o extrémní nebezpečí, protože voják, který tyto fotografie zveřejnil tím vystavil sebe a svoje spolubojovníky do ohrožení života.

Dalším způsobem je zveřejňování polohy na sociálních sítích. Na nejrozšířenější sociální síti dneška Facebooku můžeme zveřejňovat příspěvky jako statusy, fotografie, videa a další. Jenže s těmito příspěvky zveřejňujeme i svojí polohu. Dokonce Facebook mobilní aplikace pro operační systém Android

zveřejňuje polohu i při odeslání zprávy na chatu. Proto je třeba si uvědomit co všechno vlastně sdílíme a kdo všechno se k těmto informacím může dostat.

Tím, že zveřejňujeme svojí polohu se můžeme dostat jednoduše do problémů. Jedna žena z USA zveřejnila polohu restaurace, kde byla právě na večeři se svými přáteli. Za několik minut obdržela zprávu o tom, že dotyční přesně ví kde se žená právě nachází a že jí jdou vykrást dům. Žena se svými přáteli okamžitě jela zkontrolovat dům. V tomto případě se jednalo pouze o žert, ale vždy to takto dobře dopadnout nemusí. Nebo díky častému zveřejňování své polohy se můžeme stát obětí sledování, kdy si pachatel může zaznamenávat kudy pravidelně chodíme nebo kam chodíme často a může nás sledovat. V horším případě si na nás pachatel může někde počkat a dokonce nás napadnout nebo znásilnit. Může také dojít ke zneužívání online komunikace k obtěžování, zastrašování nebo šikaně vybraných osob, což spolu s vědomím toho, že osoba ví, kde se zastrašovaný nachází nebo často pohybuje, může být velmi efektivním faktorem při nahnání strachu.²² Dalším rizikem může být vydírání, kdy zveřejníme svou polohu na místě, kde by jsme neměli být. V neposlední řadě může dojít dokonce ke krádeži identity a to může být veliký problém. Díky geolokaci může pachatel vysledovat naše bydliště a dalším sledováním může získat další osobní informace pro krádež identity.

Jak je vidět rizik zveřejňování polohy na internetu existuje celá řada a další budou zcela jistě přibývat. Je proto dobré si svoje soukromí chránit, jak to jen jde a přemýšlet o tom, co všechno na internetu zveřejňujeme. V mnoha případech nám geolokace může usnadnit život, ale pokud si nedokážeme uvědomit rizika, tak nám geolokace může uškodit a to dokonce i tak, že se díky ní můžeme dostat do ohrožení života. A to by byla cena nejvyšší za neznalost. Snad se bezpečnost na internetu bude zlepšovat a nad podobnými scénáři nebudeme muset přemýšlet, ale do té doby je to hlavně na uživateli, co všechno o sobě na internetu zveřejní.

²²Chytrý telefon, který ví, kde je aneb Smartphone a jeho úskalí v oblasti soukromí. In: SLOUPENSKÁ, Hana. *Inflow Roč.6, no.5 (2013)* — Inflow.cz [online]. 2013 [cit. 2013-06-19]. Dostupné z: <http://www.inflow.cz/chytry-telefon-ktery-vi-kde-je-aneb-smartphone-jeho-uskali-v-oblasti-soukromi>

13 Praktické příklady

Zároveň s touto bakalářskou prací vznikla i webová stránka, kde jsou dostupné všechny příklady a bude tam dostupná i tato práce. Adresa této webové stránky je:

<http://bp.martinlaudat.cz>

Na této webové stránce jsou k dispozici veškeré praktické příklady, které byly vytvořeny. Zdrojové kódy těchto příkladů jsou nekomprimované a kdykoliv si je může zobrazit.

Celý tento web i praktické příklady jsou vytvořeny pomocí responzivního HTML5/CSS3/JS frameworku Zurb Foundation²³. Jedná se o framework, který umožňuje vytvářet webové stránky, které se přizpůsobují zařízení ze kterého na tento web přistupujeme. Takže by tento web měl být dobře ovladatelný jak na počítačích, tak i na mobilních zařízeních, jako jsou tablety či chytré mobilní telefony. Tuto metodu jsem zvolil protože si myslím, že geolokace ve webovém prohlížeči má hlavně smysl právě na mobilních zařízeních.

13.1 Praktické příklady s mapovými podklady

První část praktické části práce je sada příkladu, které ukazují základní práci s HTML5 Geolocation API a využití tohoto API s různými mapovými podklady. Jedná se o celkem šest ukázek. První ukázka demonstruje pouze jednorázové zjištění polohy, kde po spuštění příkladu se zobrazí hodnoty naměřené Geolocation API. Druhá ukázka je velice podobná první ukázce jen s tím rozdílem, že se jedná o periodické zjišťování polohy. Zbytek příkladů se zabývá využitím Geolocation API s mapovými podklady Google maps, Mapy.cz, Bing maps a OpenLayers. Po spuštění těchto příkladu se zobrazí mapa s grafickým ukazatelem naměřené polohy a naměřené údaje se zobrazují i pod mapou jako textové údaje.

²³Podrobnosti a stažení: <http://foundation.zurb.com>

Všechny tyto příklady slouží jako hotové řešení popisované problematiky v kapitolách 7 a 8.

13.2 Hlavní praktický příklad

Jedná se o ukázkou implementace HTML5 Geolocation API do komplexnější webové aplikace. Jako tuto aplikaci jsem zvolil projekt, na kterém spolupracuji s kolegy ve společnosti INIZIO Internet Media s.r.o., kde aktuálně pracuji. Jedná se o aplikaci, která se nazývá Předzahrádka a tato aplikace je dostupná jako odkaz z mého webu - <http://bp.martinlaudat.cz> nebo na samostatné adrese:

<http://www.predzahradka.cz>

Tato aplikace slouží jako průvodce venkovními zahrádkami u restaurací v Českých Budějovicích. Cílem bylo vytvořit aplikaci, která bude mapovat možnosti těchto venkovních zahrádek. Jedná se především o možnosti připojení k internetu, připojení k elektrické energii. Díky této aplikaci si uživatel může snadno vyhledat venkovní zahrádku kam může jít pracovat s notebookem, protože si předem dohledá informace o dostupnosti připojení k internetu a k elektrické síti, dále se může podívat na oficiální web restaurace, na všechny dostupné sociální sítě, kde daná restaurace vystupuje a na fotografii restaurace.

Tato aplikace využívá moderní webové technologie, jako jsou HTML5, PHP, MySQL, jQuery a další. Díky HTML5 Geolocation API aplikace dokáže zjistit polohu uživatele, který aplikaci právě používá a díky tomu dokáže zobrazit zahrádky, které jsou v blízkém okolí uživatele, ten si navíc může výsledky nechat vyfiltrovat pomocí jednoduchých přepínačů. Díky těmto vlastnostem je hledání velice rychlé a uživatel nemusí dlouho pátrat, které venkovní zahrádky jsou nejbližší. Dříve by se podobný problém musel řešit nativní aplikací pro každé zařízení zvlášť, ale s příchodem HTML5 už tomu tak není. Stačí nám pouze zařízení s webovým prohlížečem, které HTML5 podporuje.

14 Závěr

Nové moderní webové technologie mezi které patří i HTML5 nám umožňují vytvářet vyspělé webové aplikace. HTML5 Geolocation API je pouze malá část toho co dnes tvoří moderní webové aplikace. Přesto díky tomuto API můžeme dělat aplikace, které zpříjemní uživatelský zážitek z webové stránky či aplikace. Práce s HTML5 Geolocation API je velice jednoduchá a zvládne jí každý kdo se o tvorbu webů alespoň trochu zajímá. Toto API toho samo o sobě moc neumí, dokáže nám zjistit pouze několik údajů o pozici kde se nacházíme. S těmito údaji můžeme dále pomocí javascriptu pracovat. Ovšem ten pravý rozměr HTML5 Geolocation API dostává při použití s mapovými podklady. Díky tomu, že můžeme použít mapové podklady jako například Google maps nebo Mapy.cz, můžeme vytvářet komplexní webové služby které toto API budou využívat. Na internetu existuje hned několik služeb a aplikací využívajících geolokaci a další budou určitě přibývat. I když se jedná o moderní technologii, tak podpora v prohlížečích je velmi solidní a díky několika metodám zjišťování polohy nejsme vázáni pouze na GPS hardware.

V rám této bakalářské práce byly představeny zásadní novinky, které obsahuje standard HTML5. Bylo detailně rozebráno HTML5 Geolocation API, bylo popsáno využití tohoto API s různými mapovými podklady a následně vytvořeny praktické ukázky. V práci jsem rozebral nejdůležitější způsoby zjišťování polohy na internetu a jejich výhody a nevýhody. Otestoval jsem podporu tohoto API v prohlížečích a doporučil metody pro simulaci geolokace ve starších prohlížečích. Popsal jsem využití tohoto API na internetu a vysvětlil jsem toto využití na nejznámějších aplikacích, které toto API používají. Vypsal jsem hlavní rizika, které může časté a bezhlavé zveřejňování své polohy na internetu způsobit a na závěr jsem vypracoval praktické příklady, které jsem umístil na veřejně dostupnou webovou stránku.

Všechny cíle, které byly na začátku práce stanoveny, byly splněny.

15 Literatura a zdroje

[1] *HTML5: up and running*. 1st ed. Sebastopol, CA: O'Reilly, 2010. ISBN 05-968-0602-7.

[2] *HTML5.cz - vše co potřebujete vědět o HTML5* [online]. Dostupné z: <http://www.html5.cz/>

[3] GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. *HTML5 a CSS3 pro webové designéry*. Vyd. 1. Brno: Zoner Press, 2011, 286 s. Encyklopedie webdesignera. ISBN 978-80-7413-166-0 (BROŽ.).

[4] *Geolocation API Specification* [online]. 2011. Dostupné z: <http://dev.w3.org/geo/api/spec-source.html>

[5] LUBBERS, Peter, Brian ALBERS a Frank SALIM. *HTML5: programujeme moderní webové aplikace*. Vyd. 1. Brno: Computer Press, 2011, 304 s. ISBN 978-80-251-3539-6 (BROŽ.).

[6] HOLDENER, Anthony T. *HTML5 Geolocation*. Sebastopol, CA: O'Reilly, 2011, 95 s. ISBN 14-493-0472-9.

[7] *Geolokace v prohlížeči*. MALÝ, Martin. Zdrojak.cz [online]. 2010. Dostupné z: <http://zdrojak.root.cz/clanky/geolokace-v-prohlizeci/>

[8] *HTML5 - nové vlastnosti*: Article. In: ŠŤASTNÝ, Jiří. Programujte.com - web o programování, webdesignu, počítačové grafice, databázích, elektrotechnice a designu [online]. 2011 [cit. 2013-02-21]. Dostupné z: <http://programujte.com/clanek/2010082200-html5-nove-vlastnosti/>

- [9] *HTML5 - geolokační rozhraní*. Programujte.com - web o programování, webdesignu, počítačové grafice, databázích, elektrotechnice a designu [online]. 2012 [cit. 2013-03-14]. Dostupné z: <http://programujte.com/clanek/2011052400-html5-geolokacni-rozhrani/>
- [10] *O technologii GPS*. In: Technologie Mio [online]. 2012 [cit. 2013-06-05]. Dostupné z: http://eu.mio.com/cs_cz/global-positioning-system_jak-gps-funguje.htm
- [11] *Chytrý telefon, který ví, kde je aneb Smartphone a jeho úskalí v oblasti soukromí*. In: SLOUPENSKÁ, Hana. Inflow Roč.6, no.5 (2013) — Inflow.cz [online]. 2013 [cit. 2013-06-19]. Dostupné z: <http://www.inflow.cz/chytry-telefon-ktery-vi-kde-je-anebsmartphone-jeho-uskali-v-oblasti-soukromi>