

Střední škola spojů a informatiky,
Tábor, Bydlinského 2474

INTERAKTIVNÍ WEBOVÉ APLIKACE

Vypracoval: Pavel Vosyka
Třída: IT4
Konzultant: Ing. Dana Almášiová
Datum odevzdání: 28.3. 2014

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám za použití zdrojů a literatury v ní uvedených. Souhlasím s tím, aby moje maturitní práce byla využívána na Střední škole spojů a informatiky, Tábor, Bydlinského 2474.

.....

datum

.....

podpis

OBSAH

Úvod.....	4
Webová stránka a webová aplikace	4
Jednostránkové aplikace	4
Technologie.....	5
HTML5, CSS3	5
JavaScript, jQuery.....	5
AJAX, JSON, WebSockets.....	6
Výhody a nevýhody	8
Nároky na vývoj.....	8
Podpora prohlížečů	8
Interakce s uživatelem.....	8
Datová a výkonová náročnost.....	9
Ukládání stavů a historie prohlížeče	9
Indexování vyhledávači	10
Příklady použití.....	10
Nekonečná stránka	10
Odeslání emailu	10
Instant messaging - okamžitý chat (facebook.com).....	11
Online hry pro více hráčů (bombermine.com).....	12
Ukázka “NaDisk.cz”	13
Seznámení s aplikací.....	13
Problémy a řešení.....	15
Závěr	15
Zdroje a užitečné odkazy	16

ÚVOD

WEBOVÁ STRÁNKA A WEBOVÁ APLIKACE

Úvodem je vhodné alespoň trochu popsat základní pojmy použité v této práci. Ve snaze tak učinit, zkusme se krátce podívat do historie internetu. “World Wide Web”, zkráceně web, původně sloužil jako propojení mezi velkým množstvím hypertextových dokumentů. Ty měly z počátku povětšinou podobu strohé textové stránky. Lépe to začalo vypadat, když se začaly používat CSS kaskádové styly. Postupem času tyto webové stránky začaly využívat formuláře, jež uživateli umožňují určitou míru interakce. Server může pak tyto vstupy různě zpracovat. Obvyklé použití je ve formě kontaktního formuláře, který pak odesílá email. To jsou webové stránky, které fungují na principu klasického synchronního modelu request/response. To znamená, že klient pošle požadavek a čeká, dokud mu server neodpoví.

Se vzrůstajícími požadavky uživatelů vznikla potřeba vytvářet internetové aplikace, které by byly rychlejší a efektivnější. Za takovými účely však obyčejné HTML a CSS nebyly vytvořené, což se projevilo mnohými omezeními. Proto začaly vznikat různé technologie (Flash, Java applety), které tyto omezení obcházejí. Pro fungování však potřebovaly instalaci různých doplňků nebo pluginů do prohlížeče. Dnes se od těchto technologií pomalu upouští a vývojáři svou důvěru vkládají do rozšiřujícího se standardu HTML5, který by je měl bez problémů nahradit. Webová aplikace se od webové stránky obvykle liší tím, že narušuje klasický synchronní model request/response. Taková aplikace totiž dokáže sama komunikovat se serverem asynchronně. To znamená, že funguje, i když čeká na odpověď serveru. Během toho může uživateli oznámit, co se právě děje (např.: “Načítám data, prosím čekejte.”).

JEDNOSTRÁNKOVÉ APLIKACE

Dnes je obvyklé, že už i obyčejné webové stránky používají určité prvky webových aplikací, a tak se špatně popisuje jejich hranice. Vývoj v této oblasti stále pokračuje a rychle se mění. Jedním z pojmů, který se začíná používat je

“Single Page Application”. Ten by se dal přeložit jako “jednostránková aplikace”. Význam tohoto pojmu ještě není úplně ustálený. Dalo by se říci, že jde o aplikaci, která běží v jedné stránce prohlížeče. Ta si asynchronně komunikuje se serverem a podle potřeby upravuje tu jednu stránku. Taková aplikace “žije vlastním životem” nezávisle na prohlížeči. Ten na ní pohlíží stále jako na jednu stránku. Ukázku jedné takové aplikace, kterou jsem vytvořil, uvádím na konci této práce.

TECHNOLOGIE

HTML5, CSS3

HTML5 navazuje na starší HTML specifikace. Přináší však podstatné novinky. Na první pohled je vidět změna ve struktuře zdrojového kódu. Ať už jde o jednoduchý doctype nebo třeba sémantické tagy. Novinkou je také jednoduché vkládání multimedialního obsahu (pomocí tagů `<audio>` nebo `<video>`). Velmi zajímavým prvkem je `<canvas>`. Ten představuje kreslicí plátno, na které pomocí JavaScriptu můžeme malovat, vykreslovat vektorové obrázky, a s trochou snahy dokonce modelovat i 3D grafiku.

Zajímavou novinkou jsou také kaskádové styly ve třetí verzi specifikace. S nimi můžeme dělat moc pěkné věci. Můžeme používat plynulé přechody z jedné hodnoty na druhou, otáčet elementy nebo třeba nastavit barevné gradienty a stíny. Vlastností je celá řada. Lze také nadefinovat animace různých elementů a pak je spouštět.

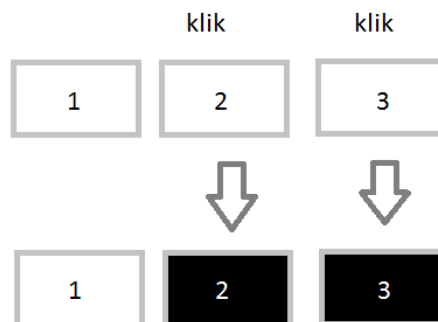
JAVASCRIPT, JQUERY

JavaScript je klientský programovací jazyk. To znamená, že je zpracováván až na straně klienta - nejčastěji ve webovém prohlížeči. K HTML5 JavaScript neodmyslitelně patří. Obstarává totiž ty mnohem zajímavější nové funkce jako je třeba geolokace (určení polohy například podle GPS), zapisování a čtení z local storage, přesouvání prvků uchycením a puštěním (drag-n-drop) nebo třeba práce se soubory či s historií prohlížeče. To jsou ty podstatnější změny, které nám pomáhají tvořit interaktivní webové aplikace.

jQuery je asi nejrozšířenější JavaScriptová knihovna. Umožňuje nám snadnější zacházení s JavaScriptem. K práci s elementy používá také CSS selektory a obsahuje spoustu předem definovaných funkcí (například pro animace). Malá ukázka práce s jQuery:

```
<html>
  <div class="tlacitka">1</div>
  <div class="tlacitka">2</div>
  <div class="tlacitka">3</div>
</html>

<script>
$( "tlacitka" ).click( function() {
  $( this ).css( "background", "black" )
    .css( "color", "white" );
});
</script>
```



Kliknutím na kterýkoli element, který má nastavenou třídu “tlacitka”, se změní CSS vlastnost “background” daného elementu na “black” a vlastnost “color” na “white”. Takže jeho pozadí zčerná a barva písma bude bílá.

AJAX, JSON, WEBSOCKETS

JavaScript nám umožňuje dělat něco na straně klienta. Díky různým technologiím, které umí asynchronně volat požadavky, může webová aplikace “žít”, aniž by vyžadovala nějakou akci od uživatele. Právě v tom je dobrý AJAX - tedy “Asynchronous JavaScript and XML” - původně sloužil k asynchronnímu volání požadavků strukturovaných do xml. Ve stejném formátu získával i odpověď.

Dnes se ve webových aplikacích jako datový formát dat používá častěji JSON - “JavaScript Object Notation”. Dovoluje nám řadit data také do polí či objektů. Je multiplatformní a dokáže ho zpracovávat každý běžně používaný programovací jazyk. Bývá datově méně náročnější než XML a to je jeho hlavní výhoda.

XML

```
<data>
<nazevAlba>Nejkrasnejsi album na svete</nazevAlba>
<fotka>
  <id>1</id>
  <odkaz>http://fotka.cz/1.jpg</odkaz>
  <popis>Moje prvni fotka</popis>
</fotka>
<fotka>
  <id>2</id>
  <odkaz>http://fotka.cz/2.jpg</odkaz>
  <popis>Moje druha fotka</popis>
</fotka>
<fotka>
  <id>3</id>
  <odkaz>http://fotka.cz/3.jpg</odkaz>
  <popis>Uz jsem profesionalni fotograf</popis>
</fotka>
</data>
```

JSON

```
{
  "data": {
    "nazevAlba": "Nejkrasnejsi album na svete",
    "fotka": [
      {
        "id": "1",
        "odkaz": "http://fotka.cz/1.jpg",
        "popis": "Moje prvni fotka"
      },
      {
        "id": "2",
        "odkaz": "http://fotka.cz/2.jpg",
        "popis": "Moje druha fotka"
      },
      {
        "id": "3",
        "odkaz": "http://fotka.cz/3.jpg",
        "popis": "Uz jsem profesionalni fotograf"
      }
    ]
  }
}
```

Asynchronní volání nám tedy umožňuje dotazovat se na server a tím třeba aktualizovat nějaká data (např. počet příchozích emailů) bez znovunačtení stránky. Systém otázka-odpověď. Z následujícího metaforického příkladu vás jistě napadne, že tento způsob není vždycky vhodný: “Už tam budem? Ne. Už tam budem? Ne. Už tam budem? Ne.” Lepší by bylo něco třeba takového: “Hele, řekni mi až tam budem. ... (dlouho nic) ... Jsme tady!” A přesně to je výhoda technologie WebSockets. Klient naváže oboustranné spojení přes něž může volně probíhat komunikace. A ještě vhodnější je použití tam, kde je potřeba hodně komunikovat. Je to totiž velmi rychlé.

Tato technologie ale vyžaduje zvláštní podporu jak ze strany serveru tak ze strany klienta. Podle webu caniuse.com aktuální verze většiny prohlížečů (mimo Opera Mini) WebSockets plně podporují. A tomu podle statistiky odpovídá asi 74% uživatelů, což je celkem slušné, ale ne stoprocentní. Snad brzy aktualizuje i zbytek uživatelů a díky tomu bude možné i pro ně tvořit zajímavější aplikace.

Ukázky použití těchto technologií si předvedeme dále (chat, bombermine).

VÝHODY A NEVÝHODY

NÁROKY NA VÝVOJ

Pokud chceme hodnotit nároky na vývoj interaktivní webové aplikace, je potřeba mít s něčím srovnání. Je logické, že jejich vývoj bude obtížnější, než vývoj obyčejných webových stránek. Srovnáme-li ale webovou aplikaci se stolní nebo mobilní aplikací, pravděpodobně zjistíme, že je jednodušší udělat jednu webovou aplikaci, která se nám bude zobrazovat v běžném prohlížeči, který funguje na většině zařízeních. Vyvíjet aplikace pro několik systémů (Windows, Windows Phone, OS X, iOS, Android, Symbian) je obvykle náročnější.

PODPORA PROHLÍZEČŮ

Je pravda, že staré verze prohlížečů mohou být někdy problémem. Nejčastější obtíže jsou se starými verzemi Internet Explorer. Je dobré znát cílovou skupinu uživatelů a jejich prohlížeče. Třeba zjistíme, že už stačí podporovat jen IE9 a vyšší, což nám výrazně usnadní práci. Pokud chceme zjistit, které prohlížeče umí nějakou konkrétní funkci, poradí nám již zmíněný web caniuse.com. Dalším dobrým pomocníkem je modernizr.com, který nám pomůže podchytit situace, když něco někde nefunguje a můžeme to obejít náhradním řešením. Tyto způsoby řešení můžeme snadno najít na internetu, protože obvykle nejsme první, kdo se s problémem setkal. Ale nemusíme to dělat za každou cenu. Někdy nevádí, když v některých prohlížečích nebudou fungovat zaoblené rohy a podobné maličkosti. Toto obcházení je dobré dělat zejména u zásadních funkcí. Pokud v některém prohlížeči není možné aplikaci zprovoznit, je dobré na to uživatele upozornit a třeba ho vybídnout, aby si nainstaloval modernější prohlížeč.

INTERAKCE S UŽIVATELEM

Interakce s uživatelem je největší výhoda interaktivních webových aplikací. Protože aplikace není úplně závislá na prohlížeči, záleží jen na vývojáři, jak se postará o to, aby uživatel vždycky věděl, co se s aplikací děje. Možnosti jsou téměř neomezené. A pokud se to dělá dobře, uživatel to ocení.

DATOVÁ A VÝKONOVÁ NÁROČNOST

Když aplikace asynchronně komunikuje se serverem, stačí jí, když dostane jen surová data (např. ve formátu JSON), protože si je pak umí sama zpracovat. To je výrazně úspornější, než kdyby měl server odesílat kompletní stránku včetně částí, které by se už jen opakovaly. Tuto úsporu ocení uživatelé nejen díky tomu, že jim ušetří množství přenesených dat od operátora (FUP), ale také kvůli rychlejší odezvě.

Protože se prohlížeč nestará jen o prosté vykreslení přijatých dat, ale i jejich zpracování skrze běžící aplikaci, je zde možnost udělat chybu (zacyklení, složité operace v rychlém intervalu), která by mohla prohlížeč příliš zatěžovat, což se projeví jeho zpomalením nebo třeba i pádem. Pro mobilní uživatele by to mohlo znamenat přehřívání telefonu a rychlejší vybíjení baterie.

UKLÁDÁNÍ STAVŮ A HISTORIE PROHLÍŽEČE

Protože veškerou obsluhu dat a komunikaci se serverem zpracovává stránka v prohlížeči, samotný prohlížeč nedokáže rozpoznat, co se děje, a myslí si, že jsme pořád na jedné stránce. Problém tedy vzniká s tlačítky “zpět” a “vpřed”, kterými by mohl uživatel omylem zavřít celou aplikaci.

Různými tlačítky může uživatel dostat aplikaci do určitého stavu. Uvedme si pro lepší představu nějaký příklad. Stránka www.fotky.cz bude nedotaženou jednostránkovou aplikací. Na ní návštěvník uvidí několik uživatelů. Když na jednoho klikne, aplikace stáhne seznam alb vybraného uživatele a zobrazí je. Výběrem alba se stáhne seznam fotek, které si pak může prohlédnout. Tento návštěvník bude chtít jednu zobrazenou fotku poslat svému kamarádovi. Klikne proto, jak je zvyklý, do adresního řádku a zkopíruje odkaz, který pak pošle emailem s textem “To je ale hezká fotka”. Kamarád si otevře odkaz a překvapeně zírá na stránku s mnoha uživateli a vůbec netuší, o jaké fotce je řeč. Chyba nastala v tom, že aplikace špatně ukládala svoje stavy.

Oba zmíněné problémy mají řešení. Po nějaké změně, ke které by se uživatel mohl chtít vrátit, je důležité a s pomocí JavaScriptu možné, změnit URL adresu. Aby nám ale prohlížeč nechtěl na novou adresu přecházet, změníme jen adresu za znakem “#” (který přidáme), tedy [www.int-web-](http://www.int-web-#)

app.cz/#/zmena1. V našem příkladu by to pak mohlo vypadat takto: www.fotky.cz/#/jmenoUzivatele/nazevAlba/fotkaCislo42. Aplikace by měla umět tuto adresu přečíst, rozeznat a následně zobrazit požadovaný stav. Jak už jsme se dozvěděli, za pomoci JavaScriptu můžeme také manipulovat s historií prohlížeče. V podstatě mu jen řekneme, kterou adresu si má do historie naší stránky uložit, čímž zajistíme správné fungování tlačítek “zpět” a “vpřed”.

INDEXOVÁNÍ VYHLEDÁVAČI

Z pochopitelných důvodů běžné vyhledávače (Google, Seznam, Bing) indexující stránky nepoužívají JavaScript. A to může být trochu problém v případě, že celá aplikace běží pouze v JavaScriptu. Vyhledávače jsou totiž obvykle velmi podstatné pro návštěvnost webu. Je proto potřeba zajistit, aby aplikace alespoň napsala, k čemu slouží, i když má prohlížeč vypnutý JavaScript. Takové řešení je prosté. Webová stránka bude v základu obsahovat nějaký popis webu. A pokud je JavaScript zapnutý, aplikace tento popis přepíše svým obsahem.

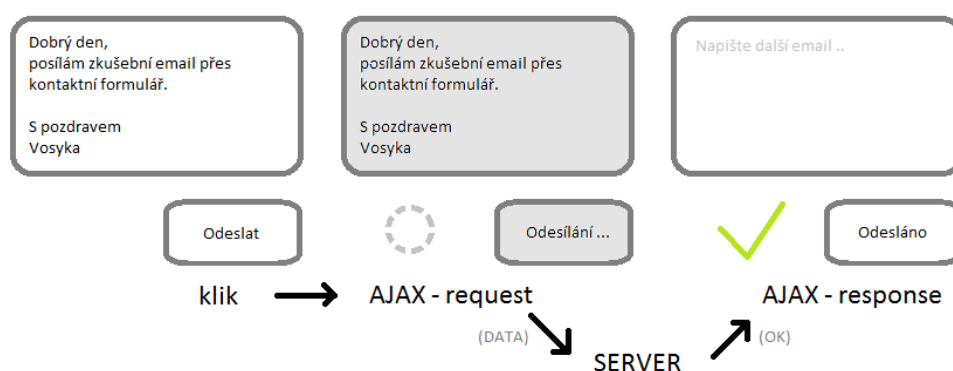
PŘÍKLADY POUŽITÍ

NEKONEČNÁ STRÁNKA

Jistě jste si toho už někde všimli. Pokud stránka nabízí velké množství obsahu, který bychom mohli chtít vidět, nemá smysl zobrazovat všechny najednou. Nejen že by dlouho trvalo, než by se nám obsah načetl, je také pravděpodobné, že uživatele nezajímá veškerý obsah, ale jen ten nejnovější. Velice oblíbené je to například na sociálních sítích. Pokud se chcete podívat na facebookovou zeď vašeho kamaráda, nemá smysl čekat, než se zobrazí všechny jeho příspěvky za posledních 5 let. Aplikace vám zobrazí například posledních 20 příspěvků a pokud pozná, že chcete vidět další (když budete rolovat stránku až na její konec) požádá server o další dávku příspěvků, které pak vykreslí na konec stránky. Elegantní řešení, které uživatel možná ani nepostřehne. A nemusí se ani proklikávat jednotlivými stránkami.

ODESLÁNÍ EMAILU

Mnoho webových stránek obsahuje kontaktní formulář. Jen některé z nich ale využívají dostupné technologie. Obvykle napíšete zprávu, a po kliknutí na odesílací tlačítko se vám stránka znovu načte s tím, že byl formulář odeslaný. V těch horších případech vás to dokonce nasměruje na externí script, ze kterého se musíte vracet zpět. V ideálním případě napíšete zprávu, a po kliknutí na odesílací tlačítko se zpráva odešle asynchronně na pozadí. O aktuálním dění vás bude informovat nějaká animace, která skončí hláškou o úspěšně odeslané zprávě.



INSTANT MESSAGING - OKAMŽITÝ CHAT (FACEBOOK.COM)

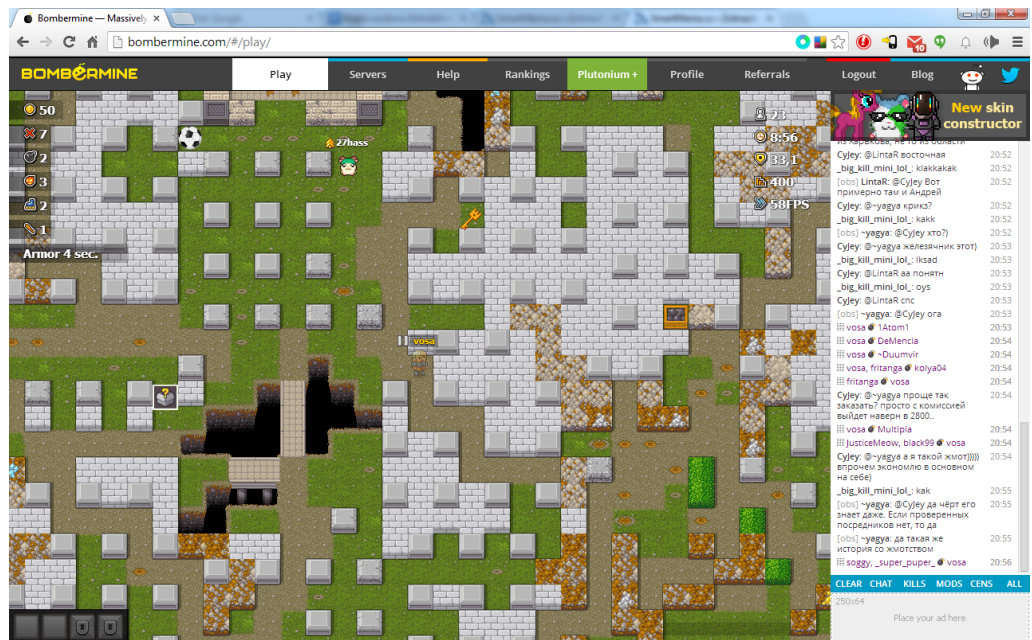
V dnešní době, kdy už je ICQ přežitkem a Skype už používají i naše (některé) babičky, asi není třeba vysvětlovat k čemu ten "instant messaging" slouží. Jak ale může něco takového fungovat ve webovém prohlížeči? Použitelné technologie jsme si již popsali. Známe tedy jejich výhody i nevýhody. Dokážete odhadnout, které asi používá facebook pro chat v prohlížeči?

Není obtížné to zjistit. Stačí otevřít okno vývojářských nástrojů, třeba v prohlížeči Chrome, a přepnout se na záložku síť, kde vidíme veškerou komunikaci aktuální stránky s internetem. Přihlásíme se tedy na facebook a spustíme chat s nějakým kamarádem. Možná vás překvapí, kolik požadavků se tam vytváří. Slouží například ke zjištění stavu ostatních přátel, kontrole nových statusů a také ke komunikaci s naším kamarádem. Stačí, že kliknu do textového pole a něco začnu psát. Prohlížeč o tom hned pošle informaci, která

je znázorněna i našemu kamarádovi v podobě tří teček. Můžeme si také všimnout jednoho doposud neodpovězeného požadavku. Není to tím, že by byl server třeba nedostupný. Serveru je jasné, že tento požadavek čeká na případnou novou zprávu od kamaráda. A protože mi zatím nic nenapsal, také server zatím neodpovídá. Aplikace ví, co to znamená, a tak spojení po 40-ti vteřinách ukončí a zeptá se znovu. Facebook zde tedy nepoužívá WebSockets (nevýhody jsme si popsali) ale AJAX. Tedy přesněji AJAJ, protože k výměně dat používá JSON.

ONLINE HRY PRO VÍCE HRÁČŮ (BOMBERMINE.COM)

Konečně se dostáváme k velmi pěkné ukázce využití WebSockets v praxi. Už víme, že právě tato technologie nám umožňuje komunikovat se serverem velmi rychle a hlavně oboustranně. A toho hojně využívá online hra Bombermine. Pojdme si ji rychle představit. Možná znáte starou oblíbenou hru “Atomic Bomberman” nebo třeba “Dyna Blaster”. Chodíte s panáčkem, pokládáte bomby, čímž můžete odstranit zdi nebo své spoluhráče. Můžete také sbírat další bonusy, díky kterým můžete třeba chodit rychleji, odkopávat bomby, pokládat více bomb nebo mít delší plamen. Na takovém principu funguje i nový bombermine. Jenže si ho zahrajete v prohlížeči. Na velké mapě s mnoha hráči. Klidně až se stovkou hráčů najednou. Veškeré akce (pohyb, položení bomby apod.), které s panáčkem uděláte, se musí okamžitě odeslat na server, který tuto změnu musí nejen zpracovat, ale také okamžitě oznámit ostatním hráčům. Stejně tak musí informovat i vás o akcích od ostatních hráčů. Probíhá zde velké množství oboustranné komunikace a je nezbytné, aby byla vyřízena co nejrychleji. Pro tento typ aplikací jsou WebSockets naprosto vhodné.



UKÁZKA “NADISK.CZ”

SEZNÁMENÍ S APLIKACÍ

Účelem této mé aplikace je umožnit nebo usnadnit ukládání dat z některých internetových služeb na disk počítače pro použití offline. Aktuálně to funguje pouze pro služby rajce.net (dále jen Rajče) a instagram.com (dále jen Instagram).

Rajče je internetové fotoalbum. Můžete nahrávat svoje fotky či videa a také prohlížet fotografie druhých uživatelů. Jednou z funkcí je možnost stažení vašeho alba v jednom zazipovaném souboru. Cizí alba si ale tímto způsobem stáhnout nemůžete. Můžete sice jednotlivě klikat na každou fotku pravým tlačítkem myši a vybrat možnost “Uložit jako”, ale to není příliš efektivní. Použijte NaDisk.cz (dále jen NaDisk).

Instagram je sociální síť, jejíž hlavním smyslem je sdílet fotografie nebo až 15-ti vteřinová videa, pořízená (nejčastěji) mobilním telefonem, s druhými. Prohlížet si je pak můžete i v prohlížeči, ale není zde žádná možnost tyto fotky stáhnout. Dokonce ani prosté kliknutí pravým tlačítkem myši na fotku nedovolí

stažení. To mě přimělo vytvořit službu NaDisk, která měla tento nedostatek řešit.

PRINCIP FUNGOVÁNÍ A POUŽITÉ TECHNOLOGIE

NaDisk funguje tak, že na základě vstupu od uživatele, jímž může být například odkaz na album, nebo přezdívka uživatele dané služby, získá aplikace seznam požadovaných dat (seznam fotek v albu) a jejich URL adresy. Tyto adresy pak posílá serveru, který si je stahuje do dočasné složky a následně je zazipuje. Server aplikaci vrátí odkaz na URL adresu hotového balíku zip, který si pak uživatel může stáhnout.

The screenshot shows the NaDisk BETA web interface. At the top, there is a red header with 'NaDisk BETA' on the left and '...pro časy offline...' on the right. Below the header, a grey navigation bar contains the text 'Rajče >> e-vosa >> Italy 2013 léto'. The main content area features a statistics table on the left and a large '42%' progress indicator on the right. The statistics table lists: Fotky: 282, Videá: 282 (?), Pole: Co sem?, Staženo: 116/282, and Stav: Stahování na server. Below the statistics, a caption reads 'Kliknutím na obrázky níže, můžete fotografie stáhnout jednotlivě'. Underneath the caption is a grid of approximately 40 small photo thumbnails.

Fotky:	282
Videa:	282 (?)
Pole:	Co sem?
Staženo:	116/282
Stav:	Stahování na server

42%

Kliknutím na obrázky níže, můžete fotografie stáhnout jednotlivě

Původně jsem to celé zpracovával pouze na straně serveru bez použití aplikace, která by starala o jiné věci. Stačil jeden formulář, do kterého uživatel zadal odkaz na album z Rajčete. Problém byl v tom, že zpracování toho skriptu trvalo dlouho, a uživatel neměl nejmenší tušení o tom, co se děje, zda-li se dočká odpovědi, která v některých případech nemusela ani přijít, protože při stahování velkého alba vypršela doba PHP skriptu.

Řešením bylo rozložit stahování fotek z celého alba do několika menších úseků. A k tomuto dělení se skvěle hodila JavaScriptová aplikace, která za pomoci AJAXu komunikuje se serverem. Diktuje mu vstup od uživatele, přijímá seznamy alb nebo adresy dat, které pak zase posílá v rozumné dávce ke stažení. Nakonec mu řekne o zazipování stažených dat a čeká na výslednou adresu hotového balíku. Během toho všeho aplikace uživateli zobrazuje, co se právě děje.

Aplikace používá JavaScriptovou knihovnu jQuery, která mimo jiné usnadňuje používání AJAXu. Přenášena data jsou ve formátu JSON. Na serveru běží PHP, které se stará o komunikaci s aplikacemi třetích stran (Rajče, Instagram) a práci s daty. K tomu používá knihovnu phpQuery.

PROBLÉMY A ŘEŠENÍ

Jedním z problémů této aplikace je aktuálně neschopnost ukládání stavů a práce s historií prohlížeče. Takže stačí kdekoli v aplikaci kliknout na tlačítko zpět, například po zadání špatného vstupu, a celá aplikace se ukončí. Nebo když se uživatel rozhodne aktualizovat stránku, aplikace se restartuje.

Řešení jsme si již uvedli. Technicky to tedy není problém. Zádrhel spočívá v tom, že už se to nevyplatí řešit, protože se služba nestala natolik populární, nakolik jsem předpokládal. Špatně vybraná cílová skupina a také propagace může způsobit, že i dobře navržená aplikace poslouží jen jako malá ukázka k maturitě a pak pravděpodobně zanikne.

ZÁVĚR

Smyslem práce bylo popsat poměrně nové technologie, díky kterým je celkem snadné tvořit kvalitní a zajímavé interaktivní webové aplikace. Je pravděpodobné, že právě tímto směrem se bude příští vývoj ubírat.

Stále ale platí, že je potřeba uvážit, kdy je vhodné je použít a kdy raději ne. V některých případech by nás použití nadstandardních technologií mohlo jenom zpomalit. Proto je důležité používat to, v čem jsme nejproduktivnější.

ZDROJE A UŽITEČNÉ ODKAZY

- http://en.wikipedia.org/wiki/Single-page_application
- <http://cs.wikipedia.org/wiki/HTML5>
- <http://cs.wikipedia.org/wiki/CSS3>
- <http://cs.wikipedia.org/wiki/AJAX>
- <http://www.zdrojak.cz/clanky/web-sockets/> (technologie WebSockets)
- <http://caniuse.com/websockets> (podpora prohlížečů)
- <http://knesl.com/articles/view/veci-ktere-me-su-na-startupech>
(všechno nemusí být SPA)
- <http://bombermine.com> (WebSokety v praxi)
- <http://facebook.com> (AJAX komunikace)
- <http://nadisk.cz> (ukázka mé nedotažené jednostránkové aplikace)