

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Automatizace interakcí na sociální síti Facebook

Bc. Jakub Škvára

Vedoucí práce: Ing. Jaroslav Kuchař

27. dubna 2013

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Jaroslavu Kuchařovi, za ochotu a pomoc při psaní této diplomové práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 27. dubna 2013

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2013 Jakub Škvára. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Škvára, Jakub. *Automatizace interakcí na sociální síti Facebook*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2013.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahradte seznamem klíčových slov v angličtině oddělených čárkou.

Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

Klíčová slova Nahradte seznamem klíčových slov v češtině oddělených čárkou.

Obsah

1	Úvod	1
1.1	Motivace	1
2	Analýza	3
2.1	Možností interakcí s facebookem	3
2.2	Možnosti ovládání webového rozhraní facebooku	7
2.3	Popis programu Selenium	9
2.4	Výhody a nevýhody použití programu Selenium WebDriver .	13
2.5	Volba programovacího jazyka a databáze	14
3	Implementace	17
3.1	Rozdělení aplikace	17
3.2	Stahování informací z facebookových profilů	17
3.3	Nahrávání dat na facebookové profily	26
4	Průběh experimentu	29
4.1	První generace facebookových účtů	29
4.2	Příprava experimentu	29
4.3	Problémy se zakládáním facebookových účtů	32
4.4	Strategie přidávání přátel k falešným účtům	33
5	Závěr	37
5.1	Vyhodnocení	37
5.2	Možnosti pokračování experimentu	37
A	Slovník použitých pojmů a zkratk	39

Seznam obrázků

2.1	Schéma přihlašování k facebooku pomocí OAuth	5
3.1	Časový rozdíl u atributů title a data-utime u elementů zobrazujících čas na facebooku	21
4.1	Zadávání telefonního čísla na facebooku	30
4.2	Potvrzení přidání osoby do přátel na facebooku	34

Úvod

1.1 Motivace

- prinos - zminit paper - fb, problemy, vek ... - zamereni na facebook - cile prace - prinos - cz prostredi - fb vs tw - vice moznosti na FB, znamejsi mezi mladymi FIT = co je fb = soc sit apod - vse je popisovano k datu duben 2013, muze se menit a meni,

Analýza

Před začátkem celého experimentu jsem prošel různé možnosti, kterými by šel experiment nejlépe realizovat. Zjišťoval jsem možnosti a omezení dostupných nástrojů a sledoval jsem výhody a nevýhody možných řešení.

2.1 Možností interakcí s facebookem

2.1.1 Rozhraní internetové stránky facebook

Pro interakci s facebookem používá většina uživatelů webové rozhraní, které je na webové adrese www.facebook.com. Toto rozhraní je upraveno pomocí techniky takzvaného responzivního designu¹ pro různé typy zařízení, takže se zobrazuje jedno rozhraní stránky pro monitory osobních počítačů s velkým rozlišením a odlišný design pro chytré telefony a tablety s malými displeji a s nízkým rozlišením. Většina uživatelů je na toto rozhraní zvyklá a to i přesto, že jej facebook často drasticky mění bez možnosti návratu ke starší verzi designu.

Dříve se kvůli změnám rozhraní nejednou zvedla velká vlna nevole (například změna osobních profilů v roce [todo citace] 2009², změna zobrazení facebookové zdi v roce 2010³, zavedení timeline v roce 2011⁴ anebo přepřa-

¹http://en.wikipedia.org/wiki/Responsive_web_design

²<http://www.zive.cz/bleskovky/facebook-tento-tyden-ceka-velka-zmena-priblizise-twitteru/sc-4-a-146045/default.aspx>

³<http://www.zive.cz/bleskovky/facebook-opet-v-novem-kompletni-zmena-okolizdi/sc-4-a-150779/default.aspx>

⁴<http://computerworld.cz/internet-a-komunikace/facebook-chrli-dalsi-novinky->

ování zpráv v roce 2012⁵). Nyní se kvůli těmto problémům snaží facebook oznamovat předem veřejnosti výhody a nové možnosti úprav designu a zapíná nové funkce postupně s možností dočasně používat starou verzi.

2.1.2 Facebook Graph API

Druhou možností, jak používat facebook jsou aplikace. Většinou se jedná o aplikace pro chytré telefony, tablety a další zařízení. Tyto aplikace komunikují s facebookem pomocí takzvaného **Graph API**⁶. [todo citace] Jedná se o RESTové webové rozhraní.

Přihlašování ke Graph API se provádí následujícím způsobem. Každá aplikace má přiřazen svůj vlastní veřejný klíč (API Key), sloužící jako jednoznačný identifikátor (ID) aplikace a dále tajný neveřejný klíč (takzvaný APP Secret), pomocí kterého se k aplikaci přihlašuje. Každý uživatel, který chce aplikaci používat, musí danou aplikaci schválit na svém facebookovém profilu a povolit oprávnění, jaká data může aplikace číst a měnit.

Celý proces přihlašování k facebooku používá OAuth autorizaci^{7,8}, při které je uživatel aplikace přesměrován na stránky facebooku, zde se musí uživatel přihlásit a při prvním použití aplikace povolit přístup dané aplikaci k uživatelskému profilu a datům. Poté je uživatel přesměrován na naše stránky a aplikace dostane token, pomocí kterého se identifikuje při každé další komunikaci s REST rozhraním (jak je vidět na obrázku 2.1). Podrobnější popis přihlašování se všemi detaily je popsán na stránkách facebooku v dokumentaci (<https://developers.facebook.com/docs/concepts/login/login-architecture/>).

Existují různé knihovny pro práci s facebook Graph API, liší se hlavně použitou platformou (Google Android, Apple iOS, ...) a programovacím jazykem (serverové: PHP, Java, apod., klientské: JavaScript). Některé z těchto knihoven mohou mít přihlašovací proces kvůli omezením mírně odlišný (například JavaScript), ale hlavní princip zůstává vždy stejný.

Nemusí se vždy jednat pouze o aplikace pro interakci s facebookem, velké

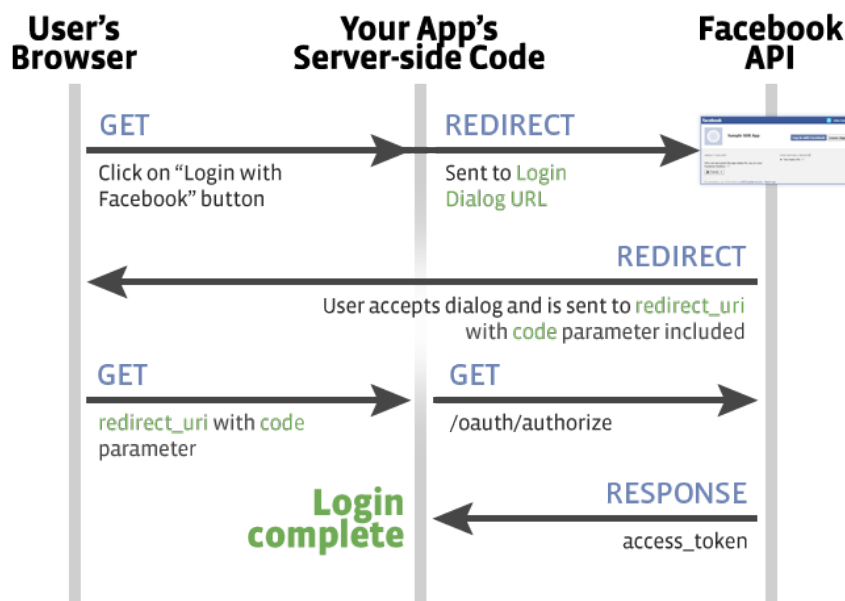
rozšiřuje-možnosti-profilu-43866

⁵<http://www.cnews.cz/clanky/facebook-se-chysta-prepracovat-zpravy-budou-se-vice-podobat-e-mailu>

⁶<https://developers.facebook.com/docs/reference/apis/>

⁷<http://oauth.net/>

⁸<https://developers.facebook.com/docs/howtos/login/server-side-login/>



Obrázek 2.1: Schéma přihlašování k facebooku pomocí OAuth

(<https://developers.facebook.com/docs/concepts/login/login-architecture/>)

množství aplikací (například hry a podobné aplikace) data z facebooku pouze čtou a získávají informace o přátelích uživatele a podobně. Takovéto aplikace jsou pak zobrazeny v okně facebooku (vlození stránky pomocí tagu `iframe`) a mohou používat facebookové prvky (tlačítka, vyskakovací okna, apod.⁹), takže méně zkušený uživatelé si ani nemusí neuvědomit, že používají aplikaci, která není vyvíjena facebookem.

Facebook nepodporuje žádnou jinou možnost, jak aplikaci ovládat kromě Graph API. Kvůli častému zneužívání tohoto API byl ale facebook postupem času nucen některé funkce tohoto rozhraní omezit nebo úplně zakázat. Jedná se hlavně o možnost dát skupině nebo stránce na facebooku like anebo přidat si přátele. Tato funkce API byla ještě spolu s takzvaným likejackingem¹⁰ často zneužívána k podvodnému navyšování fanoušků face-

⁹<https://developers.facebook.com/docs/plugins/>

¹⁰Variace clickjackingu, pomocí HTML, CSS anebo JavaScriptu je na stránce

2. ANALÝZA

bookových skupin.

Na internetu se dnes běžně objevují nabídky na nákup fanoušků do facebookových skupin (příklady viz tabulka 2.1). Ve většině případů se jedná o falešné nebo ukradené facebookové účty. Některé stránky jdou ale ještě dále a nabízejí na internetu možnost koupit si na facebooku falešnou přítelkyni (www.girlfriendhire.com, namorofake.com.br a další). Možnosti jsou různé od nákupu statusů na vaši zeď nebo s vaším ozančením až po přidání do vztahu na facebooku. Nabízí se otázka, kolik účtů, které služby nabízí, je pravých a jaká je záruka, že po převodu peněz účty nezmizí. Každopádně se jedná o zajímavý typ businessu.

Tabulka 2.1: Příklady webových stránek sloužících pro nákup Like do skupin na facebooku

Stránka	Cena
http://get-likes.com/facebook-likes-store/	1000 fanoušků za 30\$
http://fblikesmart.com/	1000 fanoušků za 50\$
http://officialfacebooklikes.com/	1000 fanoušků za 80\$

Facebook proto zrušil možnost dávat stránkám like skrze Graph API. S dalším zneužíváním funkcí facebook API se může stát, že budou dále zakázány i některé další funkce. Využití facebook API je tedy poněkud omezené a s nejistou budoucností. Hlavní nedostatky facebook Graph API pro využití pro experimentální aplikaci jsou:

- nemožnost dát like skupině
- nemožnost přidat si nové přátele
- omezený výpis informací o osobě oproti webovému rozhraní
- časem mohou být zakázány nebo omezeny některé funkce

Tyto nedostatky Graph API mi nedovolovaly jej použít pro implementaci aplikace pro experiment. Kvůli těmto omezením jsem si tedy vybral raději možnost obsluhovat facebook pomocí webového rozhraní za použití specializovaných programů a nástrojů k tomu určených.

skryto facebookové tlačítko like a je překryto jiným tlačítkem, uživatel pak po kliknutí na viditelné tlačítko nevědomky aktivuje i skryté facebookové like tlačítko <http://en.wikipedia.org/wiki/Clickjacking#Likejacking>

2.2 Možnosti ovládání webového rozhraní facebooku

2.2.1 Využití HTTP protokolu

Jako první možnost ovládání webového rozhraní facebooku se nabízelo simulovat webový prohlížeč pomocí HTTP požadavků. Toto řešení je poměrně přímočaré a webový protokol HTTP je velmi dobře zdokumenovaný a jednoduše použitelný, později jsem ale tuto možnost zavrhnul, jelikož bych musel znovu implementovat již hotové řešení, kde bych musel řešit nízkoúrovňové problémy HTTP komunikace.

Použití HTTP protokolu by se nejvíce vyplatilo u statických webových stránek. Největším problémem tohoto přístupu je ale skutečnost, že facebook používá velké množství AJAXových¹¹ požadavků a dynamických JavaScriptových oken, které by mohly způsobit komplikace. Ztížilo by to především získávání informací z uživatelských profilů a podobně. Například pro zobrazení příspěvků na zdi uživatele za určité období, je potřeba dosáhnout konce stránky, kde se musí kliknout na tlačítko načíst další příspěvky a tento postup se opakuje, do té doby, než dosáhneme požadovaného stáří příspěvků, stejný způsob zobrazování dalších dat se na facebooku používá opakovaně.

Mezi další nevýhody používání HTTP komunikace patří to, že facebook pro některé části webových stránek generuje odlišný HTML kód (mění se například atributy ID u HTML značek a podobně). Nejvíce jsem se s těmito změnami setkal na hlavní stránce facebooku, kde pravděpodobně probíhá takzvané A/B uživatelské testování stránek. To znamená, že se uživatelům zobrazují různé grafické verze stránky a následně se měří rozdíly úspěšnosti konverzního poměru měřených veličin (například: počet zobrazení stránky / počet nákupů) pro jednotlivé verze (http://en.wikipedia.org/wiki/A/B_testing).

V poslední době se začal rozšiřovat nový internetový protokol nazvaný SPDY¹², který slouží jako nadstavba nad tradičním HTTP protokolem pro rychlejší načítání stránek. Facebook tento protokol již na svých serverch používá a mohly by tak vzniknout problémy při HTTP komunikaci i když

¹¹AJAX - asynchronní načítání dynamických částí webových stránek pomocí JavaScriptu (viz [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)))

¹²<http://en.wikipedia.org/wiki/SPDY>

by měl být protokol SPDY teoreticky zpětně kompatibilní, nemuselo by vše být úplně bez problémů.

Kvůli těmto problémům spojeným s používáním HTTP protokolu jsem se tedy rozhodnul využít raději některé z hotových řešení a zabývat se komunikací s webovými stránkami na vyšších vrstvách.

2.2.2 Programy simulující chování webového prohlížeče

Jako nejjednodušší z možností ovládání webového rozhraní facebookové stránky se nasklo využití programů, které simulují internetový prohlížeč. Existuje celá škála takovýchto programů. Téměř pro každý programovací jazyk existuje některý program, který dokáže zpracovávat webové stránky. Rozdíl mezi těmito programy je především v počtu a kvalitě implementovaných webových specifikací a funkcionalit.

Programy pro zobrazování webových stránek se liší především implementovaným jádrem webového prohlížeče. V následujícím seznamu je zobrazen přehled nejpoužívanějších jader pro vykreslování webových stránek.

- WebKit¹³ - Apple Safari
- Blink¹⁴ - Google Chrome
- Gecko¹⁵ - Mozilla Firefox
- Trident¹⁶ - Microsoft Internet Explorer

Kromě těchto nejpoužívanějších jader, které používají internetové prohlížeče, existuje i velké množství méně známých jader a v podstatě každý si může implementovat vlastní vykreslovací jádro pro webové stránky.

Vykreslovací jádro ovlivňuje to, jak bude webová stránka vykreslena. Mohou se lišit i způsoby vykreslení stejné webové stránky mezi jednotlivými

¹³WebKit - <http://www.webkit.org/>

¹⁴Společnost Google starající se o prohlížeč Chrome v nedávné době oznámila, že již nadále nebude používat opensource jádro WebKit, ale vytvoří vlastní fork tohoto jádra s názvem Blink - [http://en.wikipedia.org/wiki/Blink_\(layout_engine\)](http://en.wikipedia.org/wiki/Blink_(layout_engine))

¹⁵Gecko - <https://developer.mozilla.org/en-US/docs/Mozilla/Gecko>

¹⁶Trident - [http://en.wikipedia.org/wiki/Trident_\(layout_engine\)](http://en.wikipedia.org/wiki/Trident_(layout_engine))

verzemí stejného jádra. Existují i speciálně upravená jádra pro vykreslování internetových stránek na mobilních přístrojích, které často naschvál neimplementují některé pokročilejší funkce kvůli hardwarovým nárokům mobilních zařízení a rychlosti vykreslování.

Druhým důležitým parametrem programů pro práci s webovým rozhraním internetových stránek je to, zda se webové stránky vykreslují jen v paměti počítače, nebo zda se výsledek vykresluje i na obrazovku. Existují programy, určené především pro testování webových stránek, které pouze kontrolují výskyt a pozici jednotlivých elementů, ale nedokáží stránku vykreslit, mají vše uloženo pouze v paměti. Díky tomu jsou ale i rychlejší a jsou i implementačně jednodušší.

Různé programy pro zpracování webových stránek jsou zobrazeny v tabulce 2.2. Jedním z nejpoužívanějších programů je pravděpodobně Selenium. Používá se především pro automatické testování webových stránek a je napsán v programovacím jazyce Java. Dalším používaným programem je PhantomJs. Jak už název napovídá, je tento program napsán v jazyce JavaScript. Hojně se používá pro vykreslování dynamických částí webového rozhraní internetových stránek přímo na serveru. Je to užitečné především pokud chceme poslat webové stránky pro vyhledávací roboty, kteří mají problémy s vykonáváním JavaScriptu. Další programy uvedené v tabulce slouží především pro testování výstupu webových stránek a liší se především programovacím jazykem a rozhraním pro práci s programem.

Tabulka 2.2: Příklady programů pro zpracování webových stránek

Název	Jádro	Programovací jazyk/API	Zobrazení
Selenium	Gecko	Java, C#, Ruby, Python, PHP, Perl, Haskell, REST API	vykreslení
PhantomJs	WebKit	JavaScript	paměť
WebInject	vlastní	XML API	paměť
HtmlUnit	vlastní	Java	paměť

2.3 Popis programu Selenium

Programů zpracovávajících webové stránky je velké množství. Jeden z nejpoužívanějších nástrojů se nazývá Selenium. Původně tento program vzniknul

2. ANALÝZA

jako nadstavba nad webovým prohlížečem Mozilla Firefox. Pomocí nainstalovaného pluginu v tomto prohlížeči jej lze ovládat pomocí programového API, které poskytuje program Selenium.

Jedná se o Javový `.jar` soubor, který se stará o spouštění a ovládání webového prohlížeče. API programu Selenium je nezávislé na programovacím jazyku a tak mohli vzniknout knihovny pro velké množství programovacích jazyků. Selenium používá webové RESTové API, takže se dají na internetu najít i neoficiální knihovny pro různé další jazyky, které nejsou podporovány přímo od výrobce tohoto programu. Dnes již existují k programu Selenium i pluginy, které umožňují místo prohlížeče Firefox používat prohlížeče Chrome, Operu, PhantomJS a další.

Hlavní výhoda programu Selenium spočívá v tom, že jsou webové stránky renderovány v reálném prohlížeči a operace, které se v prohlížeči odehrávají máme možnost vidět v reálném čase. V případě potřeby je možné převzít ruční kontrolu nad stránkou a pokračovat v interakci s webovou stránkou manuálně.

S použitím reálného prohlížeče se pojí i výhoda optimalizace webových stránek facebooku. Stránky jsou pro reálný prohlížeč připravovány a zcela jistě jsou v něm i facebookem důkladně testovány pro správné zobrazování a fungování všech částí. V případě nefunkčnosti by facebook všechny nedostatky co nejdříve napravil a určitě by velmi rychle reagoval na nahlášené problémy. (Facebook o některých nahlášených problémech ví, ale kvůli tomu, že pro něj nejsou natolik důležité se s nimi nezabývá, například nefunkčnost URL pro přidání podmínek pro používání facebookové aplikace nahlášena v prosinci 2012 - <https://developers.facebook.com/bugs/393455200735878>.)

Selenium je produkt, pod kterým se skrývá soubor několika programů, které slouží primárně k testování internetových stránek přímo v prohlížeči. Původně se jednalo jen o rozšíření (takzvaný plugin) do prohlížeče Firefox. Pomocí tohoto pluginu je možné nahrát konkrétní uživatelské chování na webové stránce (například: zadat konkrétní text do uživatelského pole, kliknout na tlačítko, otevřít novou stránku a podobně). Všechny akce se nahrávají a ukládají do souboru (s příponou `.html`) a lze je jednoduše znovu opakovaně spouštět. Celý postup je totožný s používáním funkce makro pro hromadné operace v programech jako jsou Microsoft Word anebo Adobe

Photoshop. Tento program se nyní nazývá Selenium IDE¹⁷.

Díky oblibě tohoto pluginu se postupem času vytvořil kolem této technologie celý ekosystém pro automatizované testování pomocí webového prohlížeče. Do tohoto ekosystému patří následující programy:

- **Selenium IDE** - plugin popsaný výše
- **Selenium Remote Control** - program pro ovládání prohlížeče lokálně nebo na jiných počítačích z různých programovacích jazyků
- **Selenium WebDriver** - program pro spouštění prohlížeče lokálně nebo na jiných počítačích
- **Selenium Grid** - za pomoci programu Selenium Remote Control dokáže spouštět webové prohlížeče na více serverech zároveň - hodí se především na testování stránek v odlišných prohlížečích a na různých operačních systémech

Více informace o jednotlivých produktech lze nalézt na oficiálních stránkách jednotlivých programů: <http://docs.seleniumhq.org/projects/>.

2.3.1 Popis fungování programu Selenium WebDriver

Pro náš experiment se nejlépe hodí program Selenium WebDriver. Jedná se o programátorské rozhraní k ovládání celého prohlížeče z různých jazyků. Můžeme tak pomocí běžného programovacího jazyka napsat program, který spustí prohlížeč, provede v něm určené úkoly a poté prohlížeč zavře. To vše bez potřeby jakékoliv lidské interakce, takže tyto úkony mohou být opakovatelné a prováděné na různých počítačích.

Celý mechanismus funguje na jednoduchém principu. Selenium WebDriver je v podstatě jen jeden `.jar` soubor¹⁸. Jedná se o lokální webový server, který naslouchá na určeném portu našeho počítače a vystavuje RESTové rozhraní k ovládání prohlížeče. Z našeho programu se pak můžeme připojit k webovému serveru Selenium WebDriver a posíláme na něj požadované

¹⁷<http://docs.seleniumhq.org/projects/ide/>

¹⁸spustitelný program napsaný v programovacím jazyce Java

2. ANALÝZA

akce, které se mají v prohlížeči vykonat. Jelikož je webové rozhraní programu nezávislé na programovacím jazyku, existují knihovny pro mnoho jazyků a je jednoduché dopsat si i vlastní knihovnu. Program Selenium WebDriver můžeme ovládat i pomocí nástrojů pro práci s HTTP protokolem (například cURL, HTTPie, ...). Celé rozhraní je detailně popsáno na stránkách projektu¹⁹.

Program Selenium WebDriver používá k ovládání prohlížeče takzvané *drivers*, v tomto případě se jedná o programy, které dokáží ovládat samotný prohlížeč - otevřít jej, otevřít konkrétní URL (viz ukázka: 2.1) a pohybovat se po stránce (pozor, neplést s ovladači u hardware, nazývají se také *drivers*, ale jedná se o něco zcela jiného). Původně dokázal Selenium WebDriver ovládat pouze prohlížeč Mozilla Firefox, ale postupem času vznikly *drivers* i pro ostatní prohlížeče. V této době již můžeme použít Selenium WebDriver k ovládání následujících prohlížečů: Mozilla Firefox, Google Chrome, Microsoft Internet Explorer, Apple Safari, Android browser (defaultní prohlížeč na mobilních telefonech s operačním systémem Google Android), iPhone browser (defaultní prohlížeč na mobilních telefonech Apple iPhone), PhantomJS (prohlížeč implementovaný v jazyce JavaScript, běžící jen v paměti počítače), HtmlUnit (prohlížeč implementovaný v jazyce Java, běžící jen v paměti počítače).

Mimo *driverů* pro webové prohlížeče existují pro program Selenium WebDriver i dva speciální *drivers*: `EventFiringWebDriver` (slouží k odchytávání JavaScriptových událostí, které v prohlížeči vzniknou, můžeme na ně pak v našem programu reagovat) a `RemoteWebDriver` (slouží ke vzdálenému ovládání prohlížečů na jiných počítačích). S těmito *drivers* máme možnost ovládat všechny hlavní internetové prohlížeče a to jak desktopové, tak i mobilní.

Listing 2.1: Příklad otevření url v seleniu pomocí CURL

```
# Pomoci tohoto prikazu vytvorime novou session
# (otevreme prohlizec firefox)
curl -X POST http://localhost:4444/wd/hub/session \
-d '{"desiredCapabilities": {"browserName": "firefox"}}'

# Z predchoziho prikazu jsme ziskali sessionId
# sessionId pouzijeme v prikazu pro otevreni stranky
# :sessionId je potreba nahradit vysledkem predchoziho
```

¹⁹<https://code.google.com/p/selenium/wiki/JsonWireProtocol>

```
# dotazu
curl -X POST \
http://localhost:4444/wd/hub/session/:sessionId/url \
-d "{url: 'http://www.facebook.com'}"
```

Vývojáři prohlížeče Opera vydali 12. února 2013 článek²⁰ o tom, že budou pro svůj prohlížeč do budoucna používat jádro WebKit, které používají prohlížeče Google Chrome a Apple Safari. Pár měsíců po tomto oznámení vydali vývojáři prohlížeče Google Chrome 3. dubna 2013 na svém webu článek²¹ o tom, že vytvoří vlastní vykreslovací jádro Blink²². Bude se jednat o fork doposud používaného opensource jádra WebKit. Do budoucna bude určitě zajímavé sledovat, jak se celá situace nakonec vyvine, nicméně funkčnost programu Selenium WebDriver by to nemělo omezit.

2.4 Výhody a nevýhody použití programu Selenium WebDriver

Selenium WebDriver je program sloužící k ovládání internetového prohlížeče pomocí velkého množství programovacích jazyků. Největší výhodou používání programu Selenium WebDriver k zpracování webových stránek je, že se stará o všechnu práci spojenou s komunikací skrze HTTP protokol, vykreslováním i ovládáním webových stránek a zpracováním všech dynamických skriptů ve stránce.

Selenium WebDriver nám umožňuje přistupovat k výsledným vyrenderovaným elementům prohlížeče a to jak textově, tak i k HTML zdrojům jednotlivých elementů na stránce. Problém nastává při následném zpracování, protože data ze stránky nejsou vždy jednotná jako v případě použití Graph API. Musíme tedy získaná data následně zpracovávat a upravovat.

Máme tedy k dispozici naprosto stejná data, jako uživatel internetového prohlížeče. Nemusíme se zabývat nízkourovňovými záležitostmi spojenými s fungováním internetu. Dále můžeme použít různé vykreslovací programy a webové prohlížeče a to včetně programů určených pro mobilní zařízení. Tento nástroj se nejvíce používá k automatickému testování webových stránek

²⁰<http://my.opera.com/ODIN/blog/300-million-users-and-move-to-webkit>

²¹<http://my.opera.com/ODIN/blog/300-million-users-and-move-to-webkit>

²²<http://www.chromium.org/blink>

v různých prohlížečích a v této oblasti je velmi rozšířen. Můžeme k němu tedy nalézt velké množství návodů, knihoven, doplňků a rad na internetu.

Výhody využití programu Selenium WebDriver

- hojně používaný program
- možnost změny vykreslovacího programu
- velké množství vykreslovacích programů k použití
- možnost ručního ovládání browseru
- velké množství návodů, zdrojů a rad na internetu
- RESTové rozhraní pro ovládání browseru
- velké množství programátorských knihoven pro práci s tímto programem
- možnost přistupovat přímo k HTML zdrojům jednotlivých elementů na stránce

Nevýhody využití programu Selenium WebDriver

- neexistuje oficiální knihovna pro jazyk PHP od tvůrců programu Selenium
- nutnost spouštět celý browser - paměťově i časově náročnější oproti použití samostatné knihovny
- nutnost dále zpracovávat získaná data z textu nebo z HTML zdrojů elementů stránky

2.5 Volba programovacího jazyka a databáze

2.5.1 Volba programovacího jazyka

Pro implementaci celé aplikace jsem si vybral programovací jazyk PHP ²³. Jedná se o dynamicky typovaný programovací jazyk, který se používá přede-

²³Open source programovací jazyk: <http://www.php.net/>

vším k vytváření webových stránek a internetových aplikací. Tento jazyk jsem si vybral především proto, že s ním mám osobně největší zkušenosti a přijde mi na menší aplikace ideální.

Díky dynamickému typování lze v tomto jazyce vytvořit aplikace velmi rychle. Protože aplikace neobsahuje složitý model a hlavní funkcionalitou je převážně volání RESTových služeb a ukládání a vybírání dat z databáze, přišel mi tento jazyk jako nejlepší možnost pro implementaci programu tohoto typu.

Celá aplikace facebooku je napsána v jazyce PHP (i když používají speciální kompilátor pro tento jazyk - HipHop for PHP²⁴). Mimo jiné programy facebook uvolnil k volnému používání na internetu knihovnu v PHP pro používání programu Selenium WebDriver²⁵. Tuto knihovnu v aplikaci také používám pro komunikaci se Selenium WebDriverem.

Při rozhodování, který programovací jazyk pro aplikaci zvolím, jsem uvažoval i o jazyku Java, který je na rozdíl od PHP staticky typovaný. Jeho hlavní výhodou oproti jazyku PHP pro tento projekt byla knihovna přímo od tvůrců programu Selenium. Naopak nevýhodou byla nutnost výsledný program před spuštěním vždy překompilovat a tedy i pomalejší vývoj celé aplikace. Výhody jazyka Java jsou vidět až u větších a rozsáhlejších aplikací a pro takto malou aplikaci by se tyto výhody vykoupené pomalejším vývojem pravděpodobně nevyplatily.

Mezi další nevýhody jazyka Java patří to, že pokud chceme nasadit aplikaci, je složitější sehnat hosting a je složitější nastavit celou aplikaci. Aplikaci v jazyce PHP je možné nasadit i na sdílený hosting s dalšími aplikacemi a konfigurace je jen minimální. Selenium WebDriver může běžet i na jiném serveru, protože veškerá komunikace porohíhá pomocí HTTP protokolu.

2.5.2 Volba databáze

Jako databázi pro ukládání dat jsem pro začátek zvolil MySQL. Jedná se o open source relační databázi, kterou aktuálně vlastní společnost Oracle (viz <http://www.mysql.com/>). Tato databáze je nejčastěji používána ve spojení právě s programovacím jazykem PHP. Do této databáze jsou ukládány in-

²⁴HipHop for PHP - http://en.wikipedia.org/wiki/HipHop_for_PHP

²⁵<https://github.com/facebook/php-webdriver>

2. ANALÝZA

formace o založených účtech, se kterými pracuje stahovací program. [V této databázi jsou také uloženy různé skupiny uživatelských účtů (například muži, ženy, apod).]

Původně jsem chtěl pro ukládání stažených dat použít také MySQL databázi, ale rozhodl jsem se jinak. Pro ukládání dat stažených z uživatelských profilů se používá NoSQL (nerelační) databáze MongoDB. MongoDB je dnes jedna z nejrozšířenějších nerelačních databází. Jedná se o open source dokumentově orientovanou NoSQL databázi o kterou se stará společnost 10gen (<http://www.mongodb.org/>). Jedna z největších předností této databáze je, že do ní lze ukládat jakákoliv nestrukturovaná data ve formátu JSON (JavaScript Object Notation - <http://www.json.org/>) a lze pak s těmito daty nadále manipulovat a provádět nad nimi agregační dotazy.

Databáze MongoDB byla vybrána hlavně kvůli tomu, že umí pracovat s jakýmkoliv nestrukturovanými daty, to znamená, že cokoliv, co získáme z facebooku můžeme rovnou uložit. Takovéto chování se hodí především k ukládání příspěvků ze zdi facebookových uživatelů a informací z profilových stránek osob. Nemusíme se starat o to, že někteří uživatelé mají omezenou viditelnost počtu přátel nebo počtu skupin.

Facebook v roce 2010 vyvinul vlastní NoSQL databázi nazvanou Cassandra (http://en.wikipedia.org/wiki/Apache_Cassandra), která slouží k vyhledávání nad velkým objemem dat. Facebook tuto databázi později nahradil další NoSQL databází nazvanou HBase (<http://hbase.apache.org/>). Tyto databáze jsou ale určeny především k vyhledávání a jejich výhoda se ukáže až při ukládání mnohem většího množství dat, než bylo pro daný experiment potřeba.

Implementace

3.1 Rozdělení aplikace

3.1.1 Architektura celé aplikace

Aplikace sloužící pro otestování celého experimentu má dvě základní funkce. První z nich je sbírání dat o uživatelích facebooku z jejich profilů a informací na jejich účtech. Druhou funkcí je ovládní fiktivních účtů, to znamená, nahrávání dat na facebook a interakce s dalšími uživateli. Data uživatelů z facebooku slouží k analýze a k dalšímu porovnání.

Jako programovací jazyk bylo pro celou aplikaci zvoleno PHP. Pro tento jazyk existuje kvalitní framework jmenující se NetteFramework (nette.org), který se hodí především na menší aplikace. Jedná se o MVP (Model - View - Presenter) framework z dílny českého autora. Tento framework se stará především o zpracování webových formulářů a zpracování požadavků webového prohlížeče, zpracování HTML šablon a celkovou architekturu celé aplikace. Více informací o tomto frameworku je k dispozici v oficiální dokumentaci (doc.nette.org). Pro webové aplikace, které jsou rozsahem podobné jako je naše experimentální aplikace je tento framework ideální.

3.2 Stahování informací z facebookových profilů

Důležitou součástí aplikace je získávání informací o uživatelích z jejich facebookových profilů. Jak již bylo zmíněno v předchozí kapitole, vybral

jsem pro stahování program Selenium WebDriver, který je spouštěn pomocí jazyka PHP. Sesbíraná data jsou ukládána do databáze MongoDB nad kterou lze jednoduše vykonávat dotazy pro získání jakýchkoliv statistických dat.

O veškeré zpracování stránek facebooku se stará program Selenium WebDriver, pomocí kterého se načte konkrétní stránka (například zeď anebo přátelé určitého uživatele), z této stránky se poté přečtou jednotlivé fragmenty (například příspěvek na zdi anebo jméno přítele na facebooku) a tato data jsou pak pomocí PHP zpracována, upravena a uložena do databáze MongoDB.

Facebook je celosvětová sociální síť a je tedy přeložen do velkého množství jazyků (včetně pirátské angličtiny a angličtiny s písmenky vzhůru nohy). Facebook je celý primárně v anglickém jazyce a do ostatních jazyků je celá aplikace přeložena. Překlad do všech jazyků probíhá pomocí speciální facebookové aplikace (<https://www.facebook.com/?app=1&sk=translations>), kam mohou své překlady zasílat všichni uživatelé facebooku. (Dříve bylo docela dlouhou dobu v českém překladu špatně přeloženo slovo neděle jako slunce - anglicky Sun = slunce a zkratka neděle.) Jazyk procházených stránek pomocí programu Selenium WebDriver byl tedy vždy nastaven na anglický jazyk, abych nedocházelo k problémům s překlady.

Pro každý náš facebookový účet uložený v aplikaci je spuštěna následující sekvence operací:

- stažení příspěvků na vlastní zdi
- stažení všech přátel, kteří si náš účet přidali
- stažení posledních příspěvků každého uživatele, kterého má účet v přátelích

3.2.1 Zpracování stahovaných informací

Všechny statusy (příspěvky na zdi uživatele), které jsou stahovány ze zdi jsou pročištěny od nedůležitých dat a upravovány do stejné podoby. Data, která jsou získávána z facebooku, jsou někdy zobrazena rozdílně. Například zobrazení like u statusů. Pro analýzu statusů potřebujeme ukládat pouze počet like, které status obdržel. Facebook bohužel ale někdy místo počtu liků zobrazuje i jména lidí, kteří like udělili. Stává se tak v případě, že like

udělil, někdo z lidí, které má účet v přátelích anebo pokud je u statusu pouze jeden like.

Pro účely experimentu se ukládá ke statusu pouze počet like, který je pro nás nejdůležitější. V programu se o získání počtu like stará funkce, která sečte počet jmen a počet like u statusů a vrátí výsledný součet celkových like.

Podobná situace nastává u komentářů u facebookových statusů. Pokud status obsahuje dva až čtyři nebo méně komentářů, tak se komentáře zobrazí i s textem. (Počet plně zobrazených komentářů se různí a není dán pevně pro každý status.) Pokud status obsahuje více komentářů, tak se zobrazí text pouze dvou (až čtyř) posledních příspěvků v komentářích u statusu a všechny předchozí komentáře se zobrazí až po kliknutí na odkaz pro zobrazení starších komentářů. Naštěstí facebook v textu tohoto odkazu zobrazuje počet předchozích komentářů. Takže pro získání počtu komentářů nemusíme odkaz rozklikávat.

Text dlouhých komentářů se navíc na facebooku zkracuje, pokud je příliš dlouhý. Většinou se zobrazí pouze prvních pět řádků komentáře. Pro zobrazení celého textu je potřeba rozkliknout odkaz na konci zkráceného textu a zobrazí se pokračování textu komentáře.

Pro komentáře u skupin na facebooku se používá jiný typ zobrazení než u statusů a osobních profilů. U statusů lidí na facebooku se zobrazují komentáře chronologicky podle data vložení, jen v jedné úrovni bez zanoření a pokud chce někdo reagovat na předchozí komentář, musí označit autora předchozího komentáře pomocí znaku @ a jména jeho profilu na facebooku. U facebookových skupin se komentáře zobrazují ve dvou úrovních s jedním zanořením, to znamená, že pokud chce někdo reagovat na určitý komentář, tak se nový příspěvek odsadí, ale pouze u první reakce a je vidět jistá hierarchie. Nadále se komentáře u příspěvků skupin neřadí podle data přidání, ale podle oblíbenosti a relevance (tento algoritmus není veřejný, ale většinou rozhoduje počet like u komentářů). Stejný způsob komentářů facebook zobrazuje i u svého doplňku Facebook Comments²⁶, který nabízí pro ostatní webové stránky na vložení jako plugin.

²⁶<https://developers.facebook.com/docs/reference/plugins/comments/>

3.2.2 Problémy se stahovaným časem

Další problém při stahování dat z facebooku nastal při získávání času statusů a komentářů. Pro uživatele zobrazuje facebook na svých stránkách v různých formátech. První možností je relativní časový úsek od vložení, například: about a minute ago (přibližně před minutou), X minutes ago (před X minutami), X hours ago (před X hodinami) a podobně. Takovéto údaje se zobrazují u komentářů vložených před méně než 24 hodinami.

Dalším zobrazovaným časovým údajem bývá: yesterday (včera), Monday (pondělí) a podobně. Jedná se o statusy a komentáře vložené před více než 24 hodinami a méně než týdnem. Poslední možností zobrazení časového údaje je přesné datum, které se zobrazuje ve formátu: 15 April (15 duben). Navíc se liší zobrazený čas u statusů a komentářů pod nimi. U starších komentářů se zobrazuje i čas vložení, například: 2 February at 16:54 (2 únor v 16.45), kdežto u statusů se zobrazuje čas jen v ojedinělých případech.

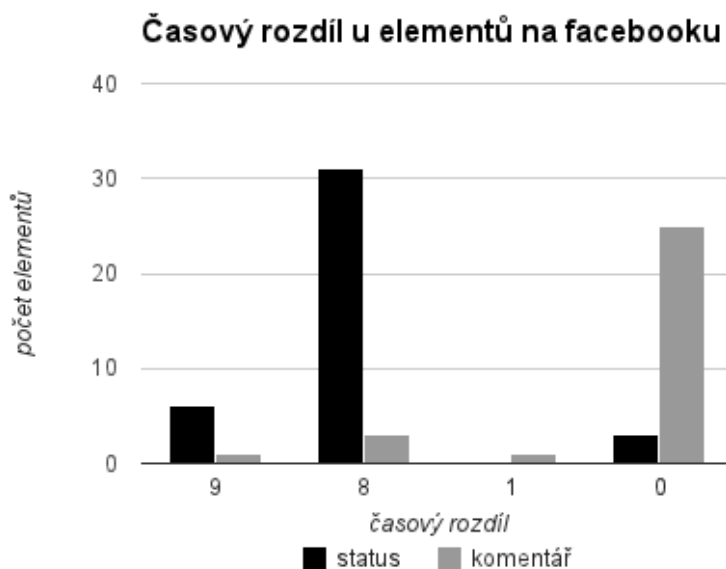
Nevýhoda takového zobrazení časových údajů je, že nejsou vůbec přesné a časový rozptyl je až 24 hodin. Zkoumal jsem proto zdrojový kód stránky, zda jsou ve stránce dostupné i další údaje týkající se času vložení statusu a komentářů. Zjistil jsem, že element obsahující čas vložení má atribut `title` s textovou časovou reprezentací a atribut `data-utime` s takzvaným unixovým časem (počet sekund od 1. ledna 1970), příklad elementu je vidět ve výpisu 3.1.

Listing 3.1: Příklad elementu obsahující čas

```
<abbr
  title="Tuesday, 23 April 2013 at 22:47"
  data-utime="1366750038"
  class="timestamp livetimestamp">
  10 minutes ago
</abbr>
```

Stejné elementy používá facebook pro zobrazení času u statusů a komentářů. Zkoumal jsem vztah mezi těmito dvěma časovými údaji uvedenými u elementu. Zjistil jsem, že tyto dva časové údaje nejsou stejné. Stáhnul jsem tedy z facebooku 70 elementů s časovými údaji, z toho bylo 30 komentářů a 40 statusů. Zjistil jsem, že u 28 elementů byly údaje v rámci elementu stejné, u jednoho elementu se čas lišil o 1 hodinu, u 34 elementů byly časové údaje rozdílné o 8 hodin a u 7 elementů se čas lišil o 9 hodin. Přehledně

jsou všechna data shrnuta v grafu 3.1.



Obrázek 3.1: Časový rozdíl u atributů title a data-utime u elementů zobrazujících čas na facebooku

U některých elementů se dokonce lišil čas o rozdílný počet hodin u statusu a odpovídajících komentářů pod ním. Tento problém bude pravděpodobně způsoben různými časovými zónami. Facebook má několik datových center po celém světě. Časový rozdíl 8 hodin odpovídá časovému posunu mezi západem USA a Greenwichským časem (základní časové pásmo). Rozdíl mezi Českou republikou a Greenwichským časem je pak 1 hodina. Kombinací těchto časových posunů pravděpodobně dojde k takovýmto rozdílům, které způsobují nekonzistence mezi časy uvedenými u stejného elementu.

Všchna data byla získána ze tří rozdílných uživatelských zdí na facebooku. Největším překvapením pro mne bylo to, že se různé rozdíly v časových údajích vyskytovaly na stejných zdech a dokonce se lišily i časové údaje u statusu a pod ním přidaných komentářích. Je možné, že facebook vybíral statusy z jiných datových center než komentáře a tím vzniknuly takovéto nekonzistence.

Nakonec jsem po tomto zjištění otestoval, který ze dvou stahovaných časových údajů odpovídá realitě. Test byl naštěstí poměrně jednoduchý.

3. IMPLEMENTACE

Vše jsem vyzkoušel vložním statusu a komentáře na facebook a poté jsem kontroloval, jak se obě data liší oproti realitě.

Z tohoto testu jsem zjistil, že realitě odpovídá údaj vložený v atributu `title`, tedy textová reprezentace data vložení. Pro data ukládání do naší databáze jsem tedy používal tuto variantu pro parsování času.

Hlavní nevýhodou použití atributu `title` pro parsování data je, že datum je uloženo ve formátu: Tuesday, 23 April 2013 at 22:47. Nemáme tedy k dispozici údaj, ve které sekundě byl příspěvek vložen, ale pro náš experiment není tento údaj tolik důležitý. Pokud bychom potřebovali doplnit ve které sekundě byl status nebo komentář vložen, mohli bychom tento údaj získat z druhého atributu elementu (`data-utime`).

3.2.3 Změna webového rozhraní facebooku

Největším problémem při použití programu selenium WebDriver a vlastně jakéhokoliv programu pracujícím s webovým rozhraním facebooku je fakt, že toto rozhraní se občas mění. Někdy se jedná o změny drobné, které běžní uživatelé této sítě ani nepozorují. Jednou za čas na facebooku ale dojde k razantnější změně webového rozhraní.

Během března 2013 facebook oznámil novou verzi profilové stránky (timeline) [todo citace <http://newsroom.fb.com/News/584/Improvements-to-Timeline>]. Jádno se o jednu z větších úprav webového rozhraní facebooku. Tato nová funkčnost byla navíc spouštěna postupně, takže někteří facebookoví uživatelé mají starou verzi timeline a některé účty již používají verzi novou. Vše funguje tak, že pokud je na vašem profilu zapnutá nová verze timeline, tak vidíte novou verzi na profilových stránkách všech vašich přátel a svojí profilové stránce. Ke staré verzi timeline se již po zapnutí facebookem nejdě nikdy vrátit. Nejde ani ovlivnit, kdy bude nová verze zapnuta právě pro váš profil.

Pro čtení dat z facebooku pomocí programu Selenium WebDriver se použije XPath (XML Path Language). Což je jazyk pro vybírání elementů z XML a HTML dokumentů. Vše funguje tak, že se vybírají konkrétní elementy anebo se vybere konkrétní element procházejí se jeho podelementy. Konkrétní elementy se vybírají podle atributů u jednotlivých elementů.

Obával jsem se, že při změně vizuálního vzhledu timeline na facebooku budu muset předělávat velkou část stahování tak, aby aplikace mohla pra-

covat s novým HTML zdrojovým kódem. Naštěstí se ale HTML elementy na stránce měnily pouze minimálně. Takže stačilo upravit některé XPath dotazy a zbytek aplikace byl zcela funkční.

Změna timeline, ale měla jiný dopad. Na staré timeline se zobrazoval počet přátel, které má uživatel přidán na facebooku (pokud to neměl uživatel zakázáno v nastavení). Se změnou na novou vizuální verzi timeline, ale tato položka zmizela a údaj o celkovém počtu přátel je dostupný až po přijetí přátelství od dotyčného člověka. V této chvíli tedy neexistuje žádný způsob, jak zjistit celkový počet přátel na facebooku u cizích profilů.

3.2.4 Stahovaná dat z facebooku

Pro účely experimentu byla u statusů ukládána následující data získaná z facebooku za použití programu Selenium WebDriver:

- myWall - zda byl status stažen ze zdi našeho fiktivního účtu nebo ze zdi někoho z přátel fiktivního účtu
- user - jméno uživatele, který status přidal
- fromUser - z kterého účtu byl status stažen
- status - text statusu
- commentsCount - počet komentářů
- likesCount - počet like
- timestamp - čas vložení statusu

Dále byli ukládáni do databáze všichni přátelé. Struktura těchto dat byla následující:

- name - Uživatelské jméno na profilu
- url - Url adresa profilu
- friendsCount - počet všech přátel stahovaného účtu, pokud to bylo možné z profilu zjistit

3. IMPLEMENTACE

- mutual - počet společných přátel stahovaného účtu, pokud to bylo možné z profilu zjistit
- fromUser - kterého uživatele je stahovaný účet přítelem
- downloaded - čas stažení informace

Poslední stahovanou kolekcí do databáze MongoDB byly profily jednotlivých přátel našich profilů. Jedná se o informace o osobě, které o sobě každý na facebooku může vyplnit. Ukládaná data do této databáze jsou následující:

- user - Uživatelské jméno profilu
- friend - zda se jedná o přímého přítele nebo ne
- friendsCount - počet všech přátel stahovaného účtu, pokud to bylo možné z profilu zjistit
- photosCount - počet všech fotek stahovaného účtu, pokud to bylo možné z profilu zjistit
- likesCount - počet všech like skupin stahovaného účtu, pokud to bylo možné z profilu zjistit
- eduwork - pole informací o vystudované škole a zaměstnání
- relationship - informace o vztahu účtu
- Bio - popis na profilu
- Birthday - datum narození
- Gender - pohlaví
- Interested In - o které pohlaví se daný účet zajímá
- Hometown - rodné město
- Current City - město, ve kterém daný účet právě žije
- Languages - jazyky, kterými účet umí mluvit
- Relationship Status - zda je daný účet ve vztahu

- relationship - s kým je daný účet v jakém typu vztahu
- Religious Views - náboženské vyznání
- Mobile Phones - telefonní čísla
- Address - adresa
- Email - e-mail
- Quote - oblíbené citace
- Screen Name - kontakty na různé komunikační kanály (Skype, Icq, ...)
- yearly - důležité mezníky v životě uživatele seřazené podle data (narození apod.)
- downloaded - čas stažení profilu

[todo kontrola]

Všechny údaje, které začínají velkým písmenkem jsou parsovány přímo z facebookové stránky. Jména klíče se tedy shodují s názvy oddílů na facebookové stránce. Začínají tedy velkými písmeny. Pro databázi MongoDB není ve velikosti písmen u klíčů žádný problém.

Některé z těchto informací nemusí být na profilech dostupné. Záleží vždy na konkrétním uživateli facebooku, zda je ochoten o sobě tyto informace prozradit a zveřejnit.

Jediná povinná informace, která se zadává již při registraci na facebooku je pohlaví, takže pohlaví se na stránce s osobními informacemi o uživateli zobrazí vždy. Počet zobrazených informací na této stránce se navíc může lišit pro uživatele, které mají profil v přátelích a pro uživatele, kteří jsou pro profil cizí. Každá sekce má své vlastní nastavení viditelnosti. Možnosti viditelnosti jsou: veřejné (viditelné pro všechny), přátelé (viditelné jen pro přátele), jen já (není viditelné pro nikoho kromě uživatele samotného) a vlastní nastavení (lze nastavit kteří konkrétní uživatelé nebo skupiny uživatelů mohou údaj vidět a pro které uživatele nebo skupiny je údaj naopak neviditelný neviditelný).

Nová verze facebookvého vzhedu timeline zavedla na stránce s osobními informacemi nové sekce jako jsou: Hudba, Filmy, Seriály, Knihy, Místa, Like, Skupiny, Fitness a Poznámky. Protože byla aplikace nastavena pro starší verzi facebookové timeline, nepřidával jsem do stahování tyto informace, jelikož některé účty měly ještě starou verzi timeline a získávané informace by byly tudíž nekonzistentní.

3.3 Nahrávání dat na facebookové profily

3.3.1 Možné způsoby interakce na facebooku

Druhá část aplikace pro experiment byla určena pro nahrávání dat na facebookvé profily. Hlavním cílem pro přidávání dat na profily byla simulace chování reálných uživatelů a zvýšení věrohodnosti účtů. Na facebooku je velké množství možností k interakci s ostatními uživateli. Nejdůležitějším prvkem na facebooku je zeď se statusy přátel. Statusy je možné komentovat, líkovat a sdílet (share = poslat na svou zeď pro zviditelnění svým přátelům).

Kromě statusů lze na facebooku komunikovat pomocí zpráv. Systém je podobný jako u e-mailových zpráv s tím rozdílem, že pomocí zpráv lze komunikovat pouze v rámci facebooku. Zprávy jsou viditelné pouze pro uživatele přidáné do komunikace, stejně jako je tomu u klasických e-mailových zpráv.

Dalším komunikačním kanálem je chat, který probíhá v reálném čase. Facebookový chat je propojen se zprávami, takže veškeré informace, které jsou posílány přes chat, jsou viditelné i ve zprávách. Výhoda chatu oproti zprávám je, že můžeme vidět, kteří uživatelé jsou právě online a zda si zaslanou zprávu již přečetli. Facebookový chat funguje pomocí otevřeného protokolu XMPP (Extensible Messaging and Presence Protocol). Tento protokol se používá pro komunikaci pomocí takzvaného Jabberu, takže pro používání facebookového chatu lze využít jakýkoliv program podporující Jabber.

Další možností, jak na sebe na facebooku upozornit určitou osobu je možnost někoho štouchnout (poke). Funkčnost tlačítka štouchnout je jednoduchá. Pokud vás někdo štouchne, přijde vám upozornění a můžete štouchnutí opakovat. Štouchnutí je znázorněno ikonou lidské ruky s napřaženým ukazováčkem.

Při implementaci jsem se soustředil především na automatické nahrávání statusů na uživatelské profily. V implementované aplikaci je sekce Tasks (úkoly), kde je možné nastavit jaké akce se mají pro které profily provést. V aplikaci jsou implementovány následující automatické akce publikování na facebook. Každá akce má jméno a parametry.

- `postStatus(text, [attachement])` - přidá na zeď uživatele status (parametr `text`), pokud je určena i příloha (`attachement`), což může být obrázek, video nebo odkaz, tak se vloží na facebook jako speciální objekt i s popisem
- `likeGroup(id)` - přidat se ke skupině (Group nebo Page), parametrem je id skupiny
- `addFriend(id)` - požádá konkrétního uživatele o přátelství na facebooku, parametrem je id uživatele
- `rsvpEvent(id, [rsvpType])` - přidat se ke konkrétní události na facebooku, parametrem je id události, druhým parametrem je účast, možné hodnoty jsou: `attend` (účastnit se), `maybe` (možná), `decline` (zamítnout účast), pokud není druhý parametr uveden, jako výchozí se bere možnost `attend`
- `inviteEvent(id)` - pozvat všechny facebookové přátele na událost, parametrem je id události

Ostatní typy možnosti komunikace na facebooku nebyly automaticky zpracovávány. Nebylo by složité doimplementovat některé další funkce jako štouchnutí všech přátel, ale pro tento experiment jsem se chtěl zaměřit na tyto hlavní funkce.

Facebook navíc začal od dubna 2013 povolovat maximálně pouze 100 pozvaných uživatelů na událost najednou a každá událost může mít maximálně 300 nevyřizovaných pozvánek. Facebook tak nutí organizátory akcí vybírat jen hosty, které bude daná akce zajímat a budou se chtít účastnit. [todo <http://www.insidefacebook.com/2013/04/09/facebook-puts-limits-on-event-invites/>, <https://www.facebook.com/help/202545109787461/>]

Celá aplikace byla spouštěna lokálně na mém počítači. Teoreticky by se dalo vše nahrát na vzdálený server a ovládat přes internet, ale tím bychom ztratili výhodu kontroly prováděných operací v prohlížeči za použití

3. IMPLEMENTACE

programu Selenium WebDriver. Existují i internetové služby, které nabízejí pronájem výpočetního času prohlížečů, takže lze mít celou aplikaci lokálně a spouštět program Selenium WebDriver přes internet vzdáleně. Problém je ale stejný jako v předchozím případě.

Pokud by se aplikace rozrostla anebo by ji používalo více uživatelů, pravděpodobně by se přenesení na vzdálený server vyplatilo. Přineslo by to ale problémy s nutností správy serveru a ošetřování neočekávaných stavů v aplikaci.

Průběh experimentu

-stahovani friend - friend dat

4.1 První generace facebookových účtů

Před samotným startem celého experimentu jsem potřeboval otestovat, jak přibližně bude vše fungovat a jak se budou účty přibližně chovat. Založil jsem proto na facebooku pět testovacích účtů, které jsem nazval **první generace**. Pomocí těchto účtů jsem zjišťoval omezení i zádrhele celého experimentu a jak by vše mělo fungovat. Pomocí těchto účtů jsem vyvíjel aplikaci pro automatické stahování a nahrávání dat na facebook (více podrobně bude celý mechanismus popsán později).

Pomocí první generace jsem ověřil, že facebook nekontroluje, pokud se z jedné IP adresy přihlašuje více účtů. Je to i logické, jelikož v institucích jako jsou školy, knihovny a podobně se může k facebooku přihlašovat více účtů, které jsou k internetu připojeny skrze jednu

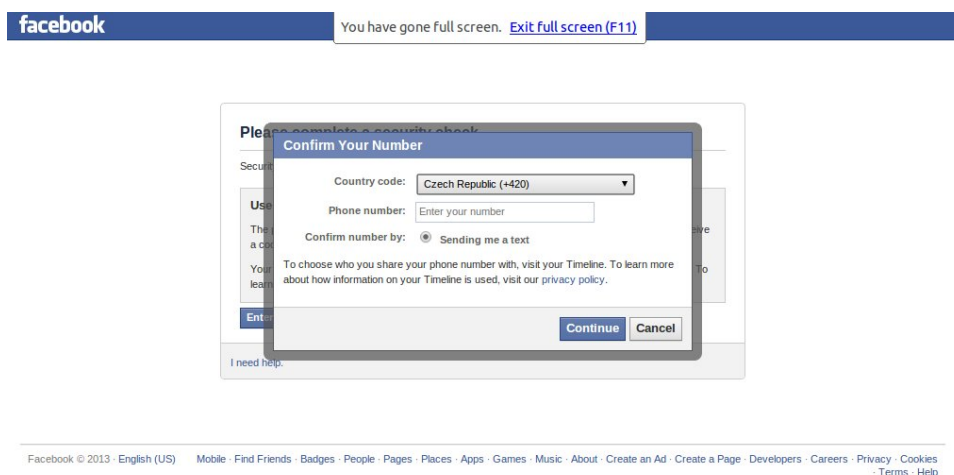
4.2 Příprava experimentu

4.2.1 Zakládání e-mailových adres

Před samotným experimentem bylo potřeba založit facebookové účty pomocí kterých se bude experiment realizovat. Zakládání falešných účtů bylo poměrně složité, jelikož jsem chtěl původně spustit celý experiment v je-

4. PRŮBĚH EXPERIMENTU

den den. Facebook při vytváření nového účtu kontroluje vždy IP adresu²⁷ počítače a nedovolí ze stejné IP adresy založit více než jeden účet denně. Pokud se někdo pokusí založit více než jeden účet, zobrazí se na facebooku obrazovka s nutností opsat kontrolní CAPTCHA²⁸ obrázek a po úspěšném opsání facebook zobrazí stránku (viz obrázek 4.1) s polem pro zadání telefonního čísla, na které má být zaslán aktivační kód. Nepokoušel jsem se pak už dále dostat přes tento krok, jelikož telefonní čísla se složitě shání a jedná se o osobní údaj, který jsem nechtěl facebooku sdělovat.



Obrázek 4.1: Zadávání telefonního čísla na facebooku

Nejdříve bylo potřeba založit e-mailové adresy, sloužící k vytvoření falešných účtů na facebooku. Všechny e-mailové adresy byly založeny na české doméně a měly koncovku `.cz`, protože jsem chtěl provést experiment na uživatelích facebooku z České republiky, aby bylo vše věrohodnější. Pro vytváření e-mailových adres jsem použil především bezplatnou e-mailovou službu od portálu seznam.cz (jedná se v první řadě o internetový vyhledávač, e-mailové schránky jsou provozovány jako jedna z doplňkových služeb). Na tomto portálu lze vytvářet e-mailové adresy s koncovkou `@seznam.cz` a `@email.cz`.

Pro každý falešný účet na facebooku jsem nejdříve připravil náhodné jméno a příjmení. Hlavním požadavkem na jméno bylo, aby bylo co nejvíce náhodné, ale zároveň i uvěřitelné. Nechtěl jsem proto vytvářet nová

²⁷IP adresa - číselná adresa počítače, pomocí které se připojuje k internetu

²⁸CAPTCHA - jedná se o ochranu internetových stránek proti robotům, nejčastěji je realizovaná nutností opisání textu z deformovaného obrázku

jména, ale raději využít již existující. Ve velkém množství případů se jednalo o spojení různých křestních jmen a příjmení získaných ze stažených facebookových profilů. U některých jmen byl změněn i rod příjmení. Nadále jsem pro vytváření jmen použil službu www.zlatestranky.cz²⁹ (v současné době stránky obsahují pouze seznam právnických osob a změnila se grafika celých stránek, ale v době kdy jsem účty zakládal byl na stránkách i seznam fyzických osob).

Struktura e-mailových adres byla následující: `jmenoprijmeni@domena`. Přednost měla doména `seznam.cz`, pokud byla e-mailová adresa s tímto jménem již zabraná, použil jsem doména `email.cz` a pokud nebyla k dispozici ani tato e-mailová schránka, tak se za jméno vložila číslice (například: `jannovak5@seznam.cz`) anebo byla použita modifikace křestního jména (domácký výraz, zdvojnásobení a podobně).

Při vytváření e-mailových adres jsem na žádná omezení nenarazil. Webových stránek nabízejících bezplatně e-mailové adresy je velké množství a to i pokud potřebujeme e-mailovou adresu pouze z české domény. Po úspěšné registraci na facebook bylo vždy potřeba aktivovat pomocí e-mailu. Facebook ihned po registraci zašle na e-mail zadávaný při registraci zprávu s odkazem. Aktivace účtu probíhá otevřením ověřovacího odkazu z e-mailu.

Facebook poté na zadaný e-mail posílá upozornění na různé aktivity, které se na facebooku dějí (například žádost o přátelství, zaslání zprávy apod.). Typy zasílaných zpráv z facebooku lze nastavit v uživatelském nastavení. Pro experiment byly e-mailové adresy důležité pouze pro povolení facebookového účtu.

4.2.2 Internetové služby poskytující dočasné e-mailové schránky

Před začátkem experimentu jsem uvažoval nad použitím některé ze služeb pro dočasné e-mailové adresy. Jedná se o služby, které vám při návštěvě vygenerují náhodnou e-mailovou adresu (například: `e1461664@rmqkr.net`). Takovéto e-mailové adresy fungují jen po omezeně dlouhou dobu a všechny e-mailové zprávy, které jsou odeslány na tuto adresu se zobrazí přímo na navštívené stránce (pouze pro návštěvníka, pro kterého byla daná e-mailová adresa vygenerována). Příklady dočasných e-mailových

²⁹ Jedná se o internetový seznam telefonních čísel fyzických a právnických osob v České republice

služeb: <http://10minutemail.com/>, <http://mailinator.com/>,
<https://meltmail.com/> a další.

Zjistil jsem ale, že facebook registrace z takovýchto dočasných e-mailových adres blokuje a po registraci je potřeba zadat telefonní číslo, na které přijde ověřovací kód, stejně jako při registraci více účtů z jedné IP adresy. Také se mi nepodařilo najít službu s dočasnými e-mailovými adresami, které by obsahovaly doménu s koncovkou `.cz`. Použil jsem proto služby českých e-mailových serverů. Hlavní výhodou je to, že k e-mailovým schránkám budu mít přístup i později a nebude problém pokud by bylo potřeba dohledat některé informace posílané e-mailem (například zapomenuté heslo nebo důvod zablokování facebookového účtu).

4.3 Problémy se zakládáním facebookových účtů

Zakládal jsem tedy profily na facebooku již dopředu, abych mohl experiment spustit najednou ve stejný čas. Bohužel se mi ale stalo to, že dříve založené účty na facebooku byly neaktivní a když jsem se k nim přihlásil po nějaké době, tak facebook požadoval zadání telefonního čísla, pro ověření totožnosti uživatele (stejně jako pokud se pokoušíme zaregistrovat více účtů denně z jedné IP adresy). Problém byl v tom, že facebook bere účty jako věrohodné až pokud má několik přátel a vyvíjí aktivitu na facebooku.

Stalo se tedy to, že jsem zakládal účty, aktivoval jsem je pomocí e-mailové adresy, ale už na účtech nebyla žádná další aktivita. Když jsem se poté v den spouštění experimentu k účtům vrátil, tak se mi objevilo textové pole s prosbou o ověření účtu zadáním telefonního čísla. Musel jsem tedy spustit experiment jen s funkčními účty s tím, že jsem po spuštění experimentu zakládal další nové účty.

Teoreticky by bylo možné celý proces zakládání účtů plně zautomatizovat. E-mailové adresy pro registrované účty by šlo jednoduše zařídit na vlastní doméně nastavením doménového koše. To znamená, že veškeré e-maily, které byly odeslané na adresu naší domény budou doručeny na jednu konkrétní e-mailovou adresu. Získáme tak teoreticky neomezené množství e-mailů k registraci na facebook.

Dále by se musel vyřešit problém s registrací z jedné IP adresy denně.

Toto omezení by šlo obejít použitím několika spolehlivých proxy serverů³⁰, přes které by byly falešné účty registrovány. Rychlost registrovaných účtů na facebooku denně by se odvíjela od počtu využívaných proxy serverů.

Pravděpodobně by se muselo vyřešit i opisování kontrolního CAPTCHA kódu, pokud facebook detekuje podezřelou robotickou činnost. Na internetu ale fungují služby, které umí za poplatek řešit opisování CAPTCHA obrázků³¹. Vše funguje automaticky pomocí OCR (Optical character recognition - optické rozpoznávání znaků) anebo lidským opisováním textů. Výhodou takovýchto služeb je, že si lze vše pronajmout a nemusíme se starat o nic jiného.

Pravděpodobně by bylo potřeba řešit i jiné problémy spojené s automatickým zakládáním facebookových profilů, které by se objevily při hlubším bádání nad problémem. Každopádně se ale určitě jedná o velmi zajímavé téma, které by šlo zpracovat více do hloubky. Protože by rozsah takového rozpracování pravděpodobně vydal na samostatnou diplomovou práci. Já jsem pro svůj experiment potřeboval pouze omezený počet facebookových účtů, takže jsem všechny účty zakládal ručně.

[todo] - etika = ovladani v pohode? - seznam uzivatelu = jen data

4.4 Strategie přidávání přátel k falešným účtům

Při vymýšlení celého experimentu bylo hlavním cílem simulovat reálné chování uživatelů na facebooku. Chtěl jsem také vyzkoušet různé strategie přidávání přátel. Na začátku experimentu jsem tedy rozdělil všechny účty do pěti kategorií podle vyplněnosti profilu. Každá kategorie měla známku od 1 do 5, přičemž známka 1 znamená, že profil je nevyplněný a známka 5 znamená, že profil je velmi vyplněný. Hlavní motivací pro takovéto rozdělení je, že více vyplněné facebookové profily by měly být pro ostatní návštěvníky věrohodnější.

Faktory ovlivňující vyplněnost profilu:

³⁰Proxy server - server, prostřednictvím kterého se lze připojit k internetu, výhodou je že není vidět naše IP adresa, ale IP adresa proxy serveru, seznam proxy serverů lze nalézt na internetu, například - <http://www.hidemyass.com/proxy-list/>

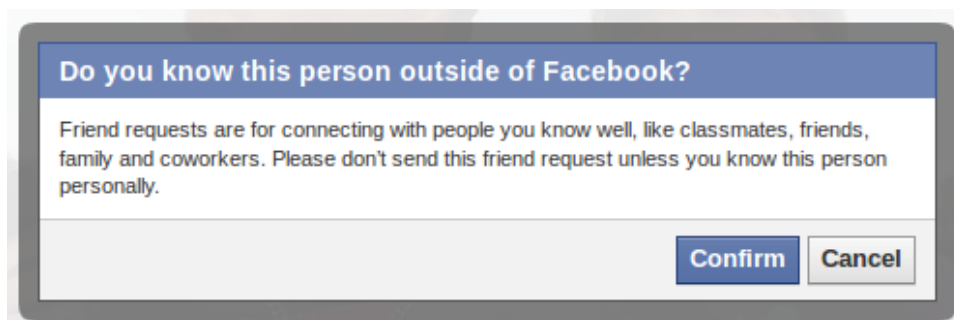
³¹<http://decaptchablog.com/decaptcher-services>

4. PRŮBĚH EXPERIMENTU

- profilový obrázek a
- foto - 5 skupin - 5 levelu - ucty se navzajem neznají

Chování uživatele + Metriky

Hlavním cílem interakce na facebooku byla věrohodnost pro ostatní facebookové uživatele. Důležité tedy bylo zautomatizovat přidávání přátel na facebooku. Ze začátku jsem po zaregistrování nového profilu plánoval přidávat pokaždé sedm přátel. Facebook ale při přidávání většího množství přátel (většinou okolo pěti, ale někdy i méně) zobrazil potvrzovací okno 4.2. Ustálil jsem tedy po tomto zjištění počet přidávaných přátel na pět za jedno přihlášení.



Obrázek 4.2: Potvrzení přidání osoby do přátel na facebooku

= získávání prvních přátel

= neznámí mezi sebou

Pro experiment jsem sdílení omezil na přidávání statusů na svou zeď. Jedná se totiž o hlavní možnost, jak dát na facebooku o sobě vědět. Frekvence přidávání statusů byla nastavena na jeden status za přihlášení.

Obsah vkládaných statusů byl obecný a neutrální. Jednalo se většinou o citáty a výroky známých i neznámých osobností na internetu, vtipy a neutrální a spontánní komentáře aktuálního dění (například o počasí). Jako zdroj pro statusy citátů sloužila webová stránka: citaty.net a jako zdroj vtipných statusů sloužily stránky: www.vtipy.net, vysmatej.cz a vtipy.peoplelovepeople.com. Snažil jsem se vyhýbat rasistickým a politickým vtipům, abych nepobouřil některého z přidávaných přátel na face-

booku. Spontánní texty statusů jsem vymýšlel sám nebo jsem se inspiroval statusy na zdi ostatních fiktivních účtů.

Po přihlášení k účtu, stažení informací a spuštění automatizovaných akcí jsem vždy ručně kontroloval co se na facebookovém účtu děje. Ve většině případů se u účtu zobrazovaly notifikace o potvrzení žádostí o přátelství a několik zpráv o tom, zda náš fiktivní účet zná osobu, které byla odeslána žádost o přátelství a občas i štouchnutí.

Facebookvé zprávy nebyly automaticky zpracovávány. Zpracovávání by totiž nepřinášelo přidanou hodnotu k našemu experimentu a zpracovávání textu by bylo složité, jak už bylo zmíněno dříve. Na žádné zprávy jsem nikdy neodpovídal. Teoreticky by bylo možné připravit univerzální odpověď na otázku, zda se dané facebookové účty znají. Například vymyšlení univerzálního místa, kde se mohly dané osoby setkat - škola, práce, apod., anebo přiznání, že se účty neznají). Pak by ale zcela určitě pokračovalo další dotazování pomocí facebookových zpráv, odkud se dané účty znají anebo proč byla odeslána žádost o přátelství na facebooku. Nejen, že by takovéto zpracovávání zabralo mnoho času a nebylo by možné plně automatizovat, ale někdy by mohlo takovéto chování být i kontraproduktivní.

Závěr

5.1 Vyhodnocení

- Grafy - Vyhodnocení

5.2 Možnosti pokračování experimentu

Pro naši aplikaci nebylo potřeba uchovávat texty komentářů u statusů, takže se tento problém nemusel řešit. Ukládá se pouze počet komentářů, který lze získat zpracováním zobrazeného textu. Ukládání textu komentářů by mohlo být užitečné, pokud by bylo možné zjistit citové zabarvení komentářů (negativní/pozitivní/neutrální). Pro takovéto analýzy jsou ale potřeba pokročilé algoritmy pro práci s textem a pro kvalitní rozpoznání by bylo potřeba delší úseky textu. Takovéto analýzy se využívají pro sledování nálady ve společnosti (například při volbách apod.) anebo pro sledování spokojenosti s určitým produktem nebo brandem (značkou) na sociálních sítích.

Hlavní cíl experimentu ale byl odlišný a počet facebookových účtů, které byly do experimentu zahrnuty nebyl dostatečně velký (hlavně počet facebookových přátel účtů), aby měly výsledky vypovídající hodnotu. Každopádně by se při větším množství účtů zahrnutých do experimentu mohlo podařit získat vypovídající masu lidí, kde by mohly být výsledky zajímavé. Zatím jsou takováto data získávána pouze z veřejně dostupných zdrojů anebo ze stránek na facebooku a takováto data mohou být zkreslená (omezením vzorku jen na určitou skupinu uživatelů). Jedná se o jeden ze směrů, kterým by se dal experiment dále rozšířit a pokračovat v něm v

budoucnu.

Kromě získávání nálady ve společnosti by bylo možné i ovlivňovat ostatní profily. Vkládáním pozitivních statusů o určitém příspěvku by mohlo sloužit jako zajímavá reklama. Lidé jsou odmítaví ke klasickým formám reklamy (letáky, televizní reklamy, apod.), ale pokud by se jednalo o pozitivně laděné sdělení, které by nevypadalo na první pohled jako reklama, mohlo by se jednat o zajímavou možnost ovlivnit mínění facebookových uživatelů. Teoreticky by nemuselo být sdělení pozitivní, někdy by mohla stačit zmínka na facebooku a uživatelé by se díky sociální aktivitě (sdílení, dávání like a komentování) postarali o šíření mezi další uživatele internetu. Reklama by mohla být i negativní. V dnešní době už existují služby pro placené blogové články na míru (Například `plac1a.cz`). Podobně by to mohlo fungovat i u facebooku.

Facebook na podzim roku 2012 zavedl možnost placení za zviditelnění statusů. Na facebooku se nyní nezobrazují všechny statusy od vašich přátel, ale jen vybrané statusy, které facebook považuje pro vás za nejhodnotnější a nejkvalitnější. Vše jde nastavit a ovlivnit skupinami do kterých si přátele zařadíte (algoritmy pro zobrazování příspěvků nejsou veřejné). Nikdy ale nemáte jistotu, že váš příspěvek uvidí všichni (nebo určití) přátelé. Za poplatek \$6.99 (přibližně 140 Kč - duben 2013) pak uživatelé facebooku mohou promovat své příspěvky, aby se zobrazily většímu počtu kamarádů. Takto lze promovat příspěvky i u facebookových stránek, kde tato funkce dává větší smysl.

[TODO] zdroje/citace??? :)

Slovník použitých pojmů a zkratek

HTML - Hypertext markup language - zdrojový formát vzhledu webových stránek, který je zpracováván internetovým prohlížečem (<http://en.wikipedia.org/wiki/HTML>).

Like skupiny na facebooku - Like = Líbí se mi - způsob připojení se ke skupně (Group) nebo stránce (Page) na facebooku

Like statusu - Like = Líbí se mi - způsob vyjádření, že se uživateli líbí status na facebooku, bez nutnosti status komentovat. Jedná se o modré tlačítko, na kterém je znázorněn palec nahoru.

Postování statusu - Vkládání příspěvku na svou zeď na facebooku.

Přidat se do vztahu na facebooku -

Přítel na facebooku - Jedá se o virtuální kamarádství mezi dvěma lidmi, které funguje tak, že oba zúčastnění musí přátelství potvrdit. Prakticky vše probíhá tak, že můžeme komukoliv na facebooku zaslat pozvánku k přátelství, dotyčný pak může pozvánku přijmout anebo odmítnout.

Profil na facebooku

(internetový) prohlížeč = browser - Program pro zpracování a zobrazování internetových stránek. Zpracovává takzvaný HTML kód webové stránky.

A. SLOVNÍK POUŽITÝCH POJMŮ A ZKRATEK

REST - Representational state transfer - Architektura rozhraní pro komunikaci s webovými službami (Web services), postavený nad HTTP protokolem
(https://en.wikipedia.org/wiki/Representational_state_transfer).

Status - Příspěvek na svou facebookovou zeď od uživatelů (Facebook nabádá k odpovědi na otázku: „What is on your mind?“ = „Co se vám honí hlavou?“).

Stránka na facebooku

Štouchnout na facebooku (Poke) -

Zeď na facebooku - V terminologii facebooku se zeď používá k označení seznamu příspěvků svých a svých kamarádů.

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	exe	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
	thesis.pdf	text práce ve formátu PDF
	thesis.ps	text práce ve formátu PS