

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

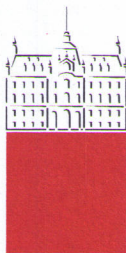
Janez Sorta

## **Integracija podatkov v socialnih omrežjih**

DIPLOMSKO DELO  
NA VISOKOŠOLSLEM STROKOVNEM ŠTUDIJU

Mentor: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2011



Št. naloge: 00162/2011

Datum: 02.09.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **JANEZ SORTA**

Naslov: **INTEGRACIJ PODATKOV V SOCIALNIH OMREŽJIH**  
**DATA INTEGRATION IN SOCIAL NETWORKS**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V zadnjih nekaj letih se je popularnost socialnih omrežij močno dvignila. Pri uporabi le teh se pojavi možnost integracije osebnih podatkov, ki jih želimo zbrati o določeni osebi. V okviru diplomskega dela najprej predstavite pojem socialnega omrežja in danes najbolj razširjena in uporabljena omrežja. Predstavite problem integracije podatkov iz različnih socialnih omrežij in pregled aplikacij, ki so na voljo za odpravljanje omenjenega problema. Izdelajte aplikacijo, ki za uporabnika združi vse razpoložljive osebne podatke iz socialnih omrežij Facebook, Twitter in LinkedIn. Predstavite vse aplikacijske vmesnike, ki jih boste v okviru izdelave aplikacije uporabili in arhitekturo REST na osnovi katere deluje večina aplikacijskih vmesnikov za socialna omrežja.

Mentor:

viš. pred. dr. Aljaž Zrnec



Dekan:

prof. dr. Nikolaj Zimic

# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani/-a     Janez Sorta,  
z vpisno številko        63070360,

sem avtor diplomskega dela z naslovom:

Integracija podatkov v socialnih omrežjih

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom (naziv, ime in priimek)  
viš. pred. dr. Aljaž Zrnec
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne \_\_\_\_\_

Podpis avtorja: \_\_\_\_\_

## **ZAHVALA**

Za mentorstvo in strokovno pomoč pri izdelavi diplomske naloge se zahvaljujem viš. pred. dr. Aljažu Zrnecu.

Posebna zahvala gre mojim staršem, ki so me spodbujali in podpirali v času študija.

## KAZALO VSEBINE

POVZETEK .....	1
ABSTRACT .....	2
1 UVOD .....	3
2 SOCIALNA OMREŽJA .....	4
2.1 Opredelitev socialnega omrežja .....	4
2.2 Predstavitev nekaterih socialnih omrežij .....	5
2.2.1 Tipi socialnih omrežij .....	5
2.2.2 Twitter .....	5
2.2.3 YouTube .....	6
2.2.4 MySpace .....	7
2.2.5 LinkedIn .....	7
2.2.6 Facebook .....	8
2.2.7 Google+ .....	9
3 PROBLEM INTEGRACIJE PODATKOV IZ RAZLIČNIH SOCIALNIH OMREŽIJ ...	11
3.1 Integracija podatkov in njen namen .....	11
3.2 Problemi pri integraciji podatkov .....	12
3.2.1 Razlike v formatih podatkov .....	12
3.2.2 Koncepti v socialnih omrežjih .....	13
3.3 Prijava v socialna omrežja .....	14
3.3.1 Google OpenSocial .....	14
3.3.2 OpenID .....	15
3.3.3 Facebook Connect .....	15
3.3.4 OAuth .....	16
3.4 Pregled programskih rešitev za integracijo podatkov v socialnih omrežjih .....	16
3.4.1 TweetDeck .....	17
3.4.2 Yoono .....	17
3.4.3 RockMelt .....	18
3.4.4 Ping.fm .....	18
3.4.5 Social Mention .....	18
4 INTEGRACIJA SOCIALNIH OMREŽIJ Z UPORABO PROGRAMSKIH VMESNIKOV .....	20
4.1 Spletni servis .....	20

4.2	(Spletni) API.....	21
4.3	REST .....	21
4.3.1	REST in http.....	22
4.3.2	HTTP metode .....	23
4.3.3	REST arhitektura.....	24
4.3.4	Varnost .....	26
5	PROGRAMSKA PODPORA ZA IZVEDBO INTEGRACIJE .....	27
5.1	Diagram primerov uporabe .....	28
5.2	Registracija aplikacije .....	30
5.3	Prijava v posamezno socialno omrežje .....	31
5.4	Prikaz osnovnih podatkov .....	34
5.5	Prikaz seznama prijateljev po posameznem socialnem omrežju.....	36
5.6	Prikaz osebnih podatkov o posameznem prijatelju .....	38
5.7	Združevanje skupnih prijateljev z več socialnih omrežij .....	40
5.8	Združevanje vseh razpoložljivih osebnih podatkov posameznega prijatelja z več socialnih omrežij .....	41
5.9	Iskalnik prijateljev .....	44
6	SKLEPNE UGOTOVITVE.....	46
	LITERATURA IN VIRI .....	48

## KAZALO SLIK

Slika 1: Uvodna stran Google+ socialnega omrežja .	10
Slika 2: Prijava s pomočjo OpenID računa	15
Slika 3: Uvodna stran izdelane spletne aplikacije	28
Slika 4: Diagram primerov uporabe	29
Slika 5: Potrditev dovoljenj ob Facebook prijavi	32
Slika 6: Nastavitev dovoljenj za dostop do podatkov Twitter aplikacije	33
Slika 7: Potrditev dostopa aplikacije do uporabniških podatkov na socialnem omrežju Twitter.....	33
Slika 8: Potrditev dostopa aplikacije do uporabniških podatkov na socialnem omrežju LinedIn	34
Slika 9: Uspešna prijava na vsa tri socialna omrežja z uporabo aplikacije	36
Slika 10: Seznam prijateljev po posameznem omrežju.....	38
Slika 11: Združeni prijatelji iz vseh treh socialnih omrežij v en seznam.....	40
Slika 12: Integrirani podatki iz socialnega omrežja Facebook in Twitter.....	43

## KAZALO TABEL

Tabela 1: Seznam dovoljenj za dostop do osebnih podatkov Facebook uporabnika	32
Tabela 2: Integracije podatkov po posameznih poljih.....	42
Tabela 3: Ocena zaupanja po posameznem socialnem omrežju	44

## SEZNAM KRATIC IN SIMBOLOV

<b>API</b>	( <i>angl. application programming interface</i> ) programski vmesnik
<b>CSS</b>	( <i>angl. cascading style sheets</i> ) programski jezik za oblikovanje spletnih strani
<b>DNS</b>	( <i>angl. domain name system/service/server</i> ) sistem domenskih imen
<b>FBJS</b>	( <i>Facebook Javascript</i> ) Facebookova alternativa Javascriptu
<b>FBML</b>	( <i>Facebook markup language</i> ) Facebookov označevalni jezik podoben HTML-ju
<b>FTP</b>	( <i>angl. File transfer protocol</i> ) protokol za prenos datotek
<b>FQL</b>	( <i>Facebook query language</i> ) Facebookov povpraševalni jezik
<b>HTML</b>	( <i>angl. Hyper Text Markup Language</i> ) označevalni jezik za izdelavo spletnih strani
<b>HTTP</b>	( <i>angl. HyperText Transfer Protocol</i> ) glavna metoda za prenos informacij na spletu
<b>HTTPS</b>	( <i>angl. HyperText Transfer Protocol Secure</i> ) je zavarovana različica http, ki uporablja SSL in TLS, da zaščiti promet pred vmesnimi opazovalci
<b>JSON</b>	( <i>angl. JavaScript Object Notation</i> ) enostaven format sporočil za izmenjavo podatkov
<b>PHP</b>	( <i>angl. Hypertext Preprocessor</i> ) programski jezik za strežniški razvoj dinamičnih spletnih vsebin
<b>REST</b>	( <i>angl. Representational state transfer</i> ) koncept programske arhitekture, namenjen porazdeljenim sistemom, kot je svetovni splet
<b>SOAP</b>	( <i>angl. Simple Object Access Protocol</i> ) protokol za izmenjavo podatkov v računalniških omrežjih z uporabo spletnih servisov

- SSL** (*angl. Secure Sockets Layer*) kriptografski protokol za zagotavljanje varnejše povezave med strežnikom in odjemalcem
- UDDI** (*angl. Universal Description, Discovery and Integration*) mehanizem za iskanje spletnih storitev z uporabo jezika XML
- URI** (*angl. Uniform Resource Identifier*) niz, ki identificira vir na svetovnem spletu
- URL** (*angl. Uniform Resource Locators*) niz, ki določa, kje na svetovnem spletu je vir dosegljiv in mehanizem za dostop do vira
- XML** (*angl. Extensible Markup Language*) razširljiv označevalni jezik za izmenjavo strukturiranih dokumentov v spletu
- WSDL** (*angl. Web Services Description Language*) jezik za opis spletnih storitev z uporabo XML-ja
- W3C** (*angl. World Wide Web Consortium*) mednarodni inštitut za spletne standarde

## POVZETEK

V zadnjih letih je moč zaslediti vse večjo popularnost najrazličnejših socialnih omrežij - naj gre za tipično deljenje trenutnih novic in fotografij ali morebiti za ozko usmerjeno področje socialnih omrežij. Pri uporabi le-teh se pojavlja težava razdrobljenosti osebnih podatkov. Tako sem v diplomski nalogi predstavil problem integracije oziroma združevanja podatkov iz več socialnih omrežij.

V teoretičnem delu sem predstavil nekatere najbolj osnovne tipe oziroma danes najbolj razširjena in uporabljena socialna omrežja. Največ pozornosti sem posvetil Facebooku, Twitterju in LinkedInu, ki so obravnavani tudi v spletni aplikaciji. Nisem pa mogel mimo novo nastalega Google+, ki naj bi predstavljal nov korak v razvoju socialnih omrežij.

Pregledal sem nekatere programske rešitve, ki so danes na voljo za integracijo podatkov v socialnih omrežjih. Predstavil sem tudi arhitekturo REST (*angl. Representational state transfer*), na kateri temeljijo programski vmesniki API (*angl. Application Programming Interface*). Ti so osnova za izvedbo programskih rešitev za integracijo podatkov. Tak pristop sem uporabil pri izdelavi spletne aplikacije v okviru praktičnega dela diplomske naloge. Na podlagi prej pridobljenih znanj in z uporabo programskih vmesnikov API sem izdelal spletno aplikacijo, ki združi razpoložljive osebne podatke uporabnika iz socialnih omrežij Facebook, Twitter in LinkedIn v smiselno celoto.

**Ključne besede:** socialno omrežje, API, REST, integracija podatkov

## **ABSTRACT**

In recent years various social networks are more and more popular. Whether it is about typically sharing news and photos or some special part of social network. There is a problem in fragmentation of personal data. In this diploma I presented problem of integration and aggregation of data from multiple social networks.

In the theoretical part I presented some of today most popular and widely used social networks. I gave most attention to Facebook, Twitter and LinkedIn, which are used in application too. I couldn't pass the new Google+, which should represent a new step in development of social networks.

I review to some today software solution for data integration in social networks. I also presented REST architecture, base for programming interfaces API. APIs are basic for my application in practical part of diploma. With my new skills and using APIs I made a web application that merge all personal data available from Facebook, Twitter and LinkedIn social networks.

**Keywords:** social network, API, REST, data integration

## 1 UVOD

Michael Durwin, raziskovalec socialnih medijev, je zapisal:

»Socialna omrežja se morajo zavedati, da ne morejo zadržati vseh uporabnikov samo na njihovem omrežju. Povprečen spletni uporabnik namreč uporablja minimalno tri socialna omrežja [2].«

Z vse večjo popularnostjo spletnih socialnih omrežij in temu podobnih storitev, ki omogočajo spletnemu uporabniku deljenje takojšnjih sporočil, spletnih dnevnikov (*angl. blog*), videov in slik, niti ni presenetljivo, da ima povprečen uporabnik več računov na takih servisih. Posledica tega je razpršenost podatkov, kar je motiviralo razvijalce, da z zbiranjem, pregledovanjem in primerjavo vsebin uporabnikov na različnih servisih olajšajo, predvsem pa pospešijo širjenje vsebin.

Aplikacija oziroma orodje, ki bi uporabnikom pomagalo pri upravljanju z več socialnimi omrežji, trenutno ni na voljo. Vsaj tako, ki bi se prijelo med uporabniki, ne. Razvijalci namreč še vedno niso podrobno definirali, kaj točno naj bi taka orodja počela, saj se socialna omrežja uporabljajo za različne namene. Če vzamemo za primer lastnika nekega podjetja, bo le-ta tipično želel dnevno prebirati novice njegovih poslovnih partnerjev, pregledovati sporočila, komentarje in spremljati pogovore drugih o produktih podjetja. Poleg naštetega bo želel odgovarjati na sporočila, posodabljati status v profilu, deliti povezave in fotografije. Vsekakor ne gre spregledati tudi povabil prijateljev na dogodke, spletne strani itd. Vse to na več socialnih omrežjih, kar se da hitro in učinkovito [2].

Arhitekturni stili in model mrežno orientirane programske arhitekture je naslov doktorske naloge Royja Fieldinga, sicer enega od glavnih avtorjev http specifikacij za verziji 1.0 in 1.1. V svojem delu je opredelil spletne arhitekturne principe, ki odgovorijo na vprašanja:

- Zakaj je splet tako razširjen in prevladujoč?
- Kaj določa merila spleta?
- Kako lahko arhitekture na spletu uporabljamo v lastnih aplikacijah?

Skupino omenjenih arhitekturnih principov je poimenoval arhitektura REST, ki je podlaga za delovanje programskih vmesnikov API, namenjenih za vgradnjo konceptov branja in pisanja iz socialnih omrežij na tretjih spletnih straneh in aplikacijah [1].

## 2 SOCIALNA OMREŽJA

### 2.1 Opredelitev socialnega omrežja

Socialna omrežja so aplikacije, spletne storitve, platforme ali strani, ki gradijo in odražajo socialne mreže ali socialne odnose med ljudmi s skupnimi interesi in/ali aktivnostmi. Te spletne aplikacije posameznikom omogočajo:

- ustvarjanje javnega ali delno javnega profila znotraj omejenega sistema,
- upravljanje seznama uporabnikov, s katerimi so povezani,
- prikazovanje in pretok njihovih seznamov povezav in povezav drugih znotraj sistema.

Strani spletnih socialnih omrežij omogočajo uporabnikom deljenje idej, aktivnosti, dogodkov in interesov znotraj svojih individualnih omrežij. Vse to pa temelji na trgih socialnih mrež, ki opredeljujejo kreativne industrije in nove medije, tako da integrirajo in preoblikujejo nove tehnologije v nove storitve. Glavno vodilo spletnih socialnih omrežij je dati posamezniku možnost ohranjanja stikov z ljudmi, ki jih morda sicer ne bi videl nikoli več. Po drugi strani pa lahko s pomočjo spletnih socialnih omrežij posameznik gradi novo obsežnejšo mrežo poznanstev in pridobiva nove prijatelje. Vsi ti spletno zgrajeni odnosi oziroma omrežja zajemajo veliko število uporabnikov interneta. Prav zato so učinkovit in na številnih področjih uporaben pripomoček, ki se ga lahko uporabi za pospeševanje prodaje, reklamiranje, povezovanje s poslovnimi partnerji, izobraževanje ipd. [3].

Za socialna omrežja so značilne določene tipične funkcije. Najpogosteje morajo uporabniki ustvariti svoj profil, ki vsebuje različne informacije o njih samih. Na svoj spletni profil lahko naložijo svoje slike, objavijo svoj spletni dnevnik, iščejo druge uporabnike s podobnimi interesi ter zbirajo in delijo seznam stikov. Dodatek uporabniških profilov pa je mesto, kjer se nahajajo komentarji prijateljev in drugih uporabnikov. Za varstvo uporabnikove zasebnosti imajo socialna omrežja navadno nastavitve, ki omogočajo uporabniku, da si sam nastavi, kdo lahko vidi njihov profil, vzpostavi kontakt z njimi, doda na seznam stikov ipd.

Nekatera spletna omrežja imajo še dodatne funkcije, kot npr. možnost ustvarjanja skupin, ki imajo skupne interese in pripadnosti, možnost nalaganja in predvajanja video datotek ter možnost razpravljanja v forumih. Ne nazadnje je postalo popularno tudi mobilno socialno mreženje, ki dodatno poenostavi konstantno povezovanje uporabnikov spletnih omrežij, saj za to ne potrebujejo več računalnika [3].

Socialna omrežja so lahko poslovno obarvana (npr. LinkedIn), namenjena deljenju določenih skupnih interesov (npr. Myspace), namenjena študentskemu druženju (npr. Facebook na začetku) ali vsem (npr. Facebook danes). Kakšne tipe socialnih omrežij bomo zasledili, je odvisno od tega, kako se do njih dostopa (npr. prek mobilnega telefona) ter kakšne interaktivne storitve nam ponujajo. Večinoma je vsem spletnim socialnim omrežjem skupna

možnost dodajanja komentarjev uporabnikov, obstajajo pa tudi takšna, kjer lahko posameznik z nekaj znaki napiše krajši blog oziroma mikroblog (npr. Twitter).

Poleg vseh pozitivnih lastnosti socialnih omrežij je treba omeniti tudi nekatere slabosti le-teh. Najpogostejše in najbolj problematično je vprašanje zasebnosti. Uporabniki v svojih spletnih profilih namreč objavijo svoje osebne podatke, ki jih je možno na različne načine zlorabiti (npr. kraja identitete). Pojavijo se številni vprašljivo pristni profili, celo dvojni profili, lažni profili in informacije ipd. Problem je, da ko enkrat objavimo določene podatke na internetu, jih je skoraj nemogoče popolnoma izbrisati. Krivcev za ogrožanje uporabnikove zasebnosti je več. Uporabnik lahko ne nazadnje objavi svoje zasebne informacije, ki jih spletna stran nezadostno zavaruje in tako omogoči tretji strani, da prosto uporablja na socialnih omrežjih objavljene podatke v različne namene. Hote ali ne hote, naši podatki kmalu postanejo dostopni vsem poznanim in neznanim uporabnikom interneta [3].

## **2.2 Predstavitev nekaterih socialnih omrežij**

### *2.2.1 Tipi socialnih omrežij*

Poznamo več tipov socialnih omrežij. Glede na nekatere danes najbolj razširjene in uporabljane oziroma glede na tiste, ki bodo predstavljene v nadaljevanju, jih lahko razdelimo v štiri osnovne skupine:

- spletnik in mikro spletnik,
- avdio in video vsebine,
- poslovna omrežja,
- socialna omrežja [4].

### *2.2.2 Twitter*

Spletnik je spletni dnevnik, namenjen ljudem, ki pišejo članke in preko njih delijo svoje misli. Osebni spletniki so še vedno aktualni, vedno več pa je namenjenih določeni temi. Mikro spletnik je po načinu podoben spletniku, t.j. uporabniki sporočajo, kaj se dogaja v tistem trenutku. Namenjeni so za sporočanje osebnih kratkih novic ali misli in tudi za sporočanje aktualnih globalnih novic.

Twitter uvrščamo med mikro blog socialna omrežja. Osnovna uporabnost temelji na izmenjavi kratkih sporočil med uporabniki.

Svojo pot je začel leta 2006 in kmalu doživel popularnost svetovne razsežnosti. Veliko uporabnikov je namreč naveličanih ostalih socialnih omrežij, ki jim ponujajo množico neuporabnih in večkrat nadležnih funkcionalnosti, kot so razne igre, kvizi itd., medtem ko ima Twitter jasno določene cilje in ponujene funkcionalnosti.

Vse skupaj je začel Jack Dorsey, programer in spletni navdušenec, ki je idejo o spletni storitvi, s pomočjo katere bi uporabniki lahko hitro sporočali, kje so in kaj počnejo, zaupal ekipi ODEO-ja, ki se ukvarja z avdio in video vsebinami. Skupaj s članom ekipe ODEO-ja Bizzom Stoneom sta izdelala prototip Twitterja. Skupaj s še tretjim članom Evanom Williamsom so hitro privabili investitorje in ustanovili sklad, iz katerega se še danes investira razvoj Twitterja.

Twitter omogoča objavljanje kratkih sporočil dolžine 140 znakov, bolje poznanih pod imenom »tweet«. Takšna dolžina je skladna z SMS (*angl. Short Message Service*) sporočili. Že res, da so SMS sporočila dolga 160 znakov, toda prvih 20 je pri Twitterju rezerviranih za uporabniško ime. Urejanje samega profila je minimalno. Uporabnik lahko spreminja sliko ozadja, osebne fotografije, barve in še nekatere najbolj osnovne nastavitve.

Twitter ima milijon registriranih aplikacij in 750 tisoč razvijalcev, ki razvojno platformo uporabljajo dnevno. Twitter je v mesecu juliju 2011 predstavil svojo novo posodobljeno spletno stran za razvijalce, na kateri so se osredotočili predvsem na:

- razprave med razvijalci, kjer poleg tipičnih tem o samem razvoju aplikacij najdemo tudi kategorijo »Dev Teatime«, namenjeno bolj socialnim temam oziroma spoznavanju razvijalcev med seboj;
- blog razvijalcev vključuje vse vrste uporabnih informacij, ki olajšajo delo razvijalcev. Sem so vključena pomembna obvestila z novostmi o vmesnikih API, dogodki, vodiči za uporabo vmesnikov API, študije primerov itd.;
- najdemo še podrobno dokumentacijo, upravitelja uporabniških aplikacij in iskalnik [5].

### 2.2.3 YouTube

YouTube je ena najpriljubljenejših internetnih strani za izmenjavo videoposnetkov, kjer jih uporabniki lahko pregledujejo, komentirajo in ocenjujejo. Ustanovitelji spletne strani so Chad Hurley, Steve Chen in Jawed Karim, nekdanji zaposleni pri podjetju PayPal. V maju leta 2005 je bila stran postavljena na ogled javnosti, šest mesecev kasneje pa je pričela z delovanjem. Danes je njen lastnik podjetje Google, ki je moral ob prevzemu zagotoviti, da bo YouTube obdržal svojih 67 zaposlenih in deloval neodvisno od Googla.

Za predvajanje video posnetkov se uporablja Adobe flash predvajalnik, kot dodatek za spletne brskalnike, ki ga ima nameščenega velika večina uporabnikov. Na voljo je tudi različica, ki za predvajanje videa uporablja HTML5 (*angl. Hyper Text Markup Language*) standard. Vendar to velja le za tiste brskalnike, ki podpirajo omenjeni standard. YouTube omogoča poleg nalaganja videa z največjo ločljivostjo 4096x3072 pik tudi 3D posnetke.

Za razvijalce sta na voljo dva vmesnika API. Prvi je namenjen samemu predvajalniku. S pomočjo vmesnika se določa izgled video predvajalnika. Poznamo dve vrsti predvajalnikov.

Prvi je klasični, ki ga zasledimo na večini spletnih straneh in tudi na samem YouTubeu. Drugi v osnovi vsebuje samo video brez kontrolnih gumbov. Razvijalci lahko sami izdelajo svoj lasten video predvajalnik. Oba predvajalnika uporabljata vmesnik API, ki se ju upravlja preko JavaScript, ActionScript in HTML jezika.

Drugi vmesnik imenovan podatkovni vmesnik API, omogoča izvajanje operacij, ki so sicer po večini na voljo na sami spletni strani YouTubea. Omogočeno je iskanje in prikaz videov ter ogled podobnih vsebin. Prav tako se lahko uporabnik predstavi s pomočjo avtentikacije in nato nalaga videe, spreminja seznam predvajanja oziroma upravlja še z nekaterimi uporabniškimi funkcijami. Sam vmesnik deluje na podlagi http zahtev in odgovorov z uporabo jezika XML, imenovan tudi Google podatkovni protokol (*angl. Google Data Protocol*), katerega delovanje temelji na REST programski arhitekturi za branje in spreminjanje podatkov [6].

#### 2.2.4 *MySpace*

MySpace je leta 2003 ustanovil Tom Anderson. Glavni značilnosti sta bogata avdio in video vsebina, ki imata ključno vlogo pri razvoju strategije omrežja. Poudarek je na zabavnih vsebinah.

Ciljna publika omrežja so predvsem mladi, glasbeniki, umetniki in ostale javne osebnosti, ki si želijo osebne promocije, temu ustrezna pa je tudi vsebina na omrežju. Deluje v smeri medijske platforme za kulturo in zabavo, sorodno televiziji, in ga uporabniki, blagovne znamke ter znane osebnosti uporabljajo za lastno izražanje in promocijo.

Na voljo so standardne funkcionalnosti socialnih omrežij. Poleg le-teh je urejanje profila mogoče z uporabo HTML-ja in CSS-ja, ki se ga lahko uporablja znotraj nekaterih vnosnih polj. Med uporabniki se je izkazal za uporabnega predvsem za dodajanje videov in flash animacij. Za urejanje in izmenjavo glasbe je na voljo posebna kategorija MySpace Music - stran, ki omogoča kreiranje glasbenih seznamov in izmenjavo skladb.

Leta 2008 je MySpace postavil razvojno platformo, imenovano MySpace Developer Platform, ki razvijalcem omogoča pisanje lastnih MySpace funkcionalnosti. Osnovana je na OpenSocial platformi.

#### 2.2.5 *LinkedIn*

Mreženje je že dolgo časa prisotno tudi v strokovnih krogih. Omogoča, da se posameznik poveže z drugimi strokovnjaki z določenega področja. Najpopularnejše poslovno socialno omrežje je LinkedIn.

LinkedIn, ki ima danes preko 100 milijonov uporabnikov, je leta 2003 ustanovil Reid Hoffman. Je najbolj uveljavljeno poslovno socialno omrežje in je ponekod že postalo orodje za iskanje novega delovnega kadra. Omrežje omogoča širjenje poslovnih stikov, iskanje poslovnih partnerjev in delavcev. Osebni profili članov so nekoliko bolj podobni življenjepisu. Uporabniki lahko preko razširjene osebne mreže iščejo poslovne prijatelje, priložnosti za zaposlitev in posel ter pomoč za razrešitev poslovnih problemov. Omogočeno je vključevanje in kreiranje skupin ter profilov podjetij. Nekaterne funkcije iskanja delovnega kadra so plačljive. LinkedIn je vzpostavil jasen protokol potrjevanja vezi in zaupanja. S sprejemom kontakta uporabnik jamči za osebo, s katero sklepa kontakt, zato sklepanje vezi s tujci ni pogosto.

LinkedIn Recruiter je napredna, plačljiva storitev za podjetja in kadrovske agencije, ki pogosteje zaposlujejo. Je orodje, ki omogoča objavo zaposlitvenih oglasov in iskanje potencialnih zaposlenih (po nazivu, lokaciji, podjetju ali ključnih besedah) med vsemi uporabniki omrežja. Omogoča tudi grupiranje kandidatov, označevanje z zaznamki, opozorila o priključitvi novih uporabnikov v omrežje, pošiljanje sporočil. Prednosti orodja so:

- več priporočil potencialnih kandidatov za delo,
- nižji stroški iskanja kandidata,
- večja učinkovitost.

LinkedIn Answers je storitev, ki nudi pomoč pri iskanju odgovora na zastavljen problem ali zgolj vprašanje. Uporabnik zastavi poslovno orientirano vprašanje, na katero lahko ostali odgovorijo. Uporabnik, katerega odgovor ostali označijo kot najboljši med vsemi podanimi odgovori, si prisluži točko, te pa mu prinašajo večjo kredibilnost in ugled strokovnjakov.

LinkedIn razvijalcem ponuja dve vrsti vmesnikov API:

- JavaScript vmesnik API z bogato knjižnico za izdelavo dinamičnih aplikacij v spletnih brskalnikih. Interakcija poteka preko vtičnikov (*angl. plugin*), ki jih je mogoče najti na strani za razvijalce.
- REST API, kot so ga poimenovali pri LinkedIn, omogoča preprost dostop do podatkov o uporabnikih, podjetjih in povezavami med njimi. Podatki so v formatih XML in JSON.

### 2.2.6 Facebook

Socialna omrežja nam omogočajo, da ostajamo v stiku s prijatelji in sklepamo nova prijateljstva. Priljubljenost spletnih socialnih omrežij je v izjemnem porastu. Eno danes najpriljubljenejših, če ne kar najpriljubljenejše, je Facebook.

Zasnoval ga je Mark Zuckerberg, takrat študent 2. letnika univerze Harvard, leta 2004. Prvotno je bila spletna stran namenjena le študentom omenjene univerze, postopoma je postala dostopna ostalim institucijam, leta 2006 pa tudi globalni populaciji.

Več kot 750 milijonov ljudi na svetu uporablja Facebook in vsaj polovica uporabnikov ga obišče vsaj enkrat na dan, kjer v povprečju preživi 20 minut. Facebook je ogromen vir osebnih podatkov, kjer uporabniki večinoma uporabljajo svoja prava imena in elektronski naslov, se povezujejo s svojimi pravi prijatelji ter delijo svoje misli in novice.

Za razvijalce je na voljo Facebook platforma, ki vsebuje množico vmesnikov API in orodij za prenos ter obdelavo podatkov na zunanjih straneh. Razvijalci so se primorani naučiti tudi nekaterih jezikov, da lahko uporabljajo vmesnike API:

- FQL povpraševalni jezik za pridobivanje podatkov,
- FBML označevalni jezik, za katerega bi lahko rekli, da je okrnjena verzija HTML,
- FBJS skriptni jezik kot alternativa za JavaScript

Družabni ali socialni graf, kot ga je imenoval Zuckerberg, predstavlja prepletenost odnosov med uporabniki družabnih omrežij, tudi izven Facebooka. S tem naj bi pridobili še pametnejši splet bližje semantičnemu spletu. Podatki o uporabnikih so namreč razvijalcem na voljo preko vmesnikov API, s čimer so lahko aplikacije bolj prilagojene posameznikom. Na konkretnem primeru to pomeni, da bo lahko spletni radio obiskovalcu ponudil glasbo, ki mu je všeč, saj bo te podatke dobil iz Facebooka [7].

V tesni povezavi s socialnim grafom je open graph protokol, katerega cilj je še bolj integrirati Facebook v spletne strani in aplikacije. Z vključevanjem Open Graph značk na spletnem mestu to izenači s Facebook stranjo. Z značkami se določi, kako se bo stran pojavljala na Facebooku. Vzemimo za primer, da na spletno stran integriramo gumb »všeč mi je«. V praksi to pomeni, da se z uporabnikovim klikom na gumb vzpostavi povezava med spletno stranjo in uporabnikom. Tako se obravnavana spletna stran pojavi med uporabnikovimi dejavnostmi in interesi. Na spletno mesto je potrebno vključiti meta značke in gumb »všeč mi je«. Ključne značke so:

- *og:title* – naslov strani/objave in hkrati tudi besedilo, ki se izpiše, ko uporabnik s klikom na »všeč mi je« gumb deli vsebino z ostalimi;
- *og:type* – kategorija objekta (spletne strani), ki služi za kategorizacijo Facebook strani, ki so posledično ustvarjene;
- *og:image* – URL grafika, ki se prikaže med objavami. To je lahko določena standardna fotografija ali dinamično ustvarjen URL glede na objavo. Če ta ni določena, Facebook izbere fotografije, ki jih najde na strani, ali pa določi privzet Open graf logotip;
- *og:url* – URL spletne strani, ki je uporabljen kot stalni id v Open grafu;
- *og:site\_name*: ime spletnega mesta, ki se izpiše ob objavi;
- *fb:admins* / *fb:app\_id* – id uporabnika in/ali aplikacije, ki je administrator tega spletnega mesta in hkrati tudi administrator Facebook strani, ki je potem ustvarjena.

### 2.2.7 Google+

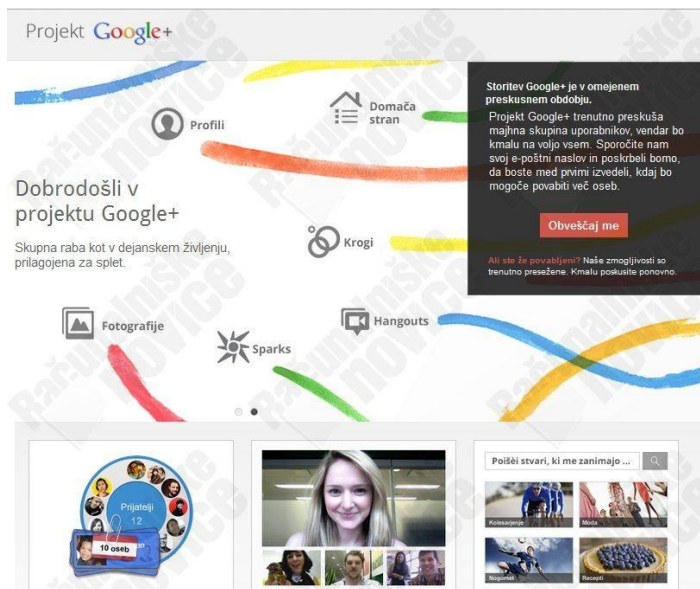
Danes Googlovo socialno omrežje Google+ uporablja že 20 milijonov spletnih uporabnikov. Tu je zanimivo predvsem dejstvo, da je to uspelo le preko zasebnih povabil (socialno omrežje

namreč še ni prosto dostopno) v pičlih treh tednih. Socialno omrežje je sicer na voljo od 28. junija letos. Pričakovano ga uporablja največ Američanov, in sicer pet milijonov. S poltretjim milijonom uporabnikov jim sledijo Indijci.

V storitev Google+ so vključeni elektronski poštni predal Gmail, spletni iskalnik Google, mobilna platforma Android, spletni brskalnik Chrome in še nekaj novih in nadvse uporabnih spletnih storitev. Obsega štiri ključne storitve, in sicer: +Circles, +Sparks, +Hangouts in +Mobile.

Ključna komponenta +Circles (krogi) omogoča razlikovanje med prijatelji. Prijatelje razdelimo v različne skupine glede na njihovo vlogo (na primer posel, družina, sorodniki in podobno). Uporabnik s tem pridobi možnost nadzora nad tem, kakšne vsebine bo delil s kom. Medtem ko pri socialnih omrežjih uporabniki stremijo k temu, da bi imeli čim več prijateljev oziroma stikov, to pri +Circles ne bo mogoče, vsaj v takšni meri ne. To posledično pomeni bolj brezskrbno deljenje misli ter fotografij, saj zasebnost ne bo ogrožena.

Storitev +Sparks (ideje) temelji na osnovi že izvedenih iskanj in interesov. Podjetje Google nam tu v realnem času posreduje izbrane vsebine in posodobitve neposredno iz svetovnega spleta. Gre za nekakšno mešanico priljubljenih storitev Google News in Google Reader. Naslednja v vrsti storitev je +Hangouts (video srečanja), ki omogoča spletno združevanje prijateljev. Tu lahko prijatelji klepetajo v živo in to celo preko videokonferenc (do 10 prijateljev). Storitev +Mobile (mobilni) ne koristi le komunikacijskim zmožnostim mobilnih naprav (predvsem tistih, opremljenih z mobilnim operacijskim sistemom Android), ampak nudi polno podporo še za sistem GPS, digitalno kamero in sporočila.



Slika 1: Uvodna stran Google+ socialnega omrežja.

### 3 PROBLEM INTEGRACIJE PODATKOV IZ RAZLIČNIH SOCIALNIH OMREŽIJ

#### 3.1 Integracija podatkov in njen namen

Dandanes smo deležni vse večje prisotnosti socialnih omrežij. Kot posledica tega je moč na številnih spletnih straneh zaslediti komponente različnih socialnih omrežij. Naslednji korak v razvoju ne bi bil nastanek novega Facebooka, temveč uveljavitev orodja, ki bo upravljalo s podatki, ki jih vsa ta socialna omrežja nudijo. Z razvojem socialnega spleta je uporabnik prisiljen k uporabi vse več socialnih omrežij in hitro se zgodi, da postane preobremenjen. Razvijalci poskušajo najti način za upravljanje z vsemi temi podatki preko namiznih aplikacij, brskalnikov ali spletnih strani. Glavna značilnost teh aplikacij je, da nas obveščajo o spremembah, objavah v realnem času. Pri razvoju je treba posebno pozornost nameniti:

- zbiranju podatkov,
- filtriranju podatkov,
- ažurnosti podatkov [11].

Integracija podatkov je proces, ki zagotavlja enoten pogled na podatke, razdeljene po različnih virih. Glavni cilj integracije podatkov je omogočiti enostavno in hitro pridobivanje informacij, ko je na voljo več virov. Proces integracije podatkov igra ključno vlogo na področju skladiščenja podatkov (*angl. data warehouse*) in podatkovnem rudarjenju (*angl. data mining*). To postaja vedno bolj pomembno, vse odkar količina podatkov, ki jih uporabnik lahko objavi v socialnih omrežjih, ne predstavlja več omejitve. Da se podatkovno rudarjenje lahko izvaja, mora obstajati enoten pogled na podatke iz različnih socialnih omrežij. Na tem mestu postane integracija podatkov nujno potrebna.

Semantični splet, imenovan po Timu Bernersu, ima kot sestavni del spleta tretje generacije pomembno vlogo pri integraciji spletno orientiranih podatkov. Semantično orientirane spletne strani namreč že vsebujejo mehanizme, ki predstavljajo podatke v obliki, primerni za nadaljnjo obdelavo oziroma integracijo.

Osnovni namen integracije podatkov je uporabniku olajšati upravljanje s socialnimi omrežji, število katerih se dnevno povečuje, njihova uporaba pa je vedno bolj pomembna. Prav različne usmerjenosti socialnih omrežij so privedle do problema integracije podatkov. Te lahko temeljijo na:

- osebnih odnosih (ohranjanje stikov),
- skupnih interesih (osebnih, socialnih, na podlagi vsebin - knjige, filmi itd.),
- profesionalnih odnosih (strokovni stiki, iskanje zaposlitve).

Uporabniki so tako prisiljeni uporabiti več socialnih omrežij, da bi s tem preprečili prepletanje osebnih in poklicnih interesov. Sezname stikov na Facebooku in LinkedInu se tako pri večini uporabnikov bistveno razlikujejo.

Osnovni namen integracije podatkov je torej pridobivanje in obdelava podatkov, pridobljenih iz več socialnih omrežij. Še bolj natančno lahko aplikacije podatke pridobivajo, objavljajo

oziroma počnejo oboje. Še preden nastopi obdelava podatkov, se mora uporabnik prijaviti v posamezno socialno omrežje. Namen je poenotiti uporabniške podatke za prijavo, tako da bi obstajalo eno uporabniško ime in geslo za prijavo v vsa socialna omrežja. Le-to se zdi težko uresničljivo, saj vsako socialno omrežje teži k svojemu načinu prijave, npr. Facebook Connect.

V integracijo podatkov je vključeno:

- spremljanje novih objav in ostalih vsebin,
- dialog med uporabniki znotraj aplikacij,
- deljenje vsebin z ostalimi uporabniki,
- analiza podatkov.

Na tem mestu lahko integracijo podatkov v socialnih omrežjih razdelimo po tipu ciljnih uporabnikov. Tu gre lahko za domače ali poslovno orientirane uporabnike. Vzemimo za primer poslovno orientiranega uporabnika, ki tipično želi objaviti informacijo o novem izdelku na več socialnih omrežjih in spremljati odzive uporabnikov. Vse to želi narediti hitro in učinkovito. Domači uporabnik pa bo tipično želel prebirati in komentirati objave prijateljev iz več socialnih omrežij naenkrat.

Eden pomembnejših ciljev integracije podatkov je analiza in obdelava pridobljenih podatkov. Namen je torej pridobljene podatke analizirati in jih ponuditi v za uporabnika primerni obliki. To sem želel pokazati v praktičnem delu diplomske naloge, kjer sem pridobljene podatke analiziral, združil v celoto in jih kot rezultat posredoval uporabniku aplikacije.

## 3.2 Problemi pri integraciji podatkov

### 3.2.1 Razlike v formatih podatkov

Raznolikost v formatih podatkov je glavni problem v integraciji podatkov. V splošnem jo lahko razdelimo v tri širše kategorije:

- *Strukturna/shematična raznolikost* se pojavi zaradi razlik v strukturi različnih podatkovnih baz. Večina aplikacij potrebuje hiter dostop do podatkov, za kar je potrebna optimizacija strukture podatkovne baze. Posledica tega so strukturne in shematične raznolikosti med viri podatkov.
- *Sintaktična raznolikost* je najbolj pogost problem, ki se danes pojavlja. Pojavi se zaradi razlik v protokolih, jezikih/vmesnikih (SQL, ODBC), podatkovnih modelih (relacijski, objektno usmerjeni) itd.
- *Semantična raznolikost* nastane zaradi različnih razlag pomena osnovnih podatkov.

Na podlagi raznolikosti formatov poznamo tri glavne kategorije podatkovnih modelov:

- *Strukturirani podatkovni model* vsebuje dobro organizirane podatke z uporabo struktur, kot so na primer tabele. Shema za predstavitev informacij je dobro definirana. Glavna prednost tega modela je preprost način izvajanja poizvedb v sistemu. Primeri

strukturiranih podatkov so ER model (*angl. entity-relationship model*), UML (*angl. Unified Modeling Language*), relacijski model (*angl. relational model*) itd.

- *Pol strukturirani podatkovni model* ne vsebuje nobene razmejitev med podatki in njeno shemo. Aplikacija določa število atributov v shemi, ki se dejansko uporabljajo. Glavna prednost takega sistema je zagotavljanje prilagodljivosti formatov podatkov za izmenjavo med različnimi vrstami podatkovnih baz in enostavno spreminjanje sheme. Izvajanje poizvedb na takšnem sistemu ni tako preprosto kot v strukturiranem modelu. Za primere polovično strukturiranega podatkovnega modela veljajo TCP/IP paketi, e-mail sporočila, stisnjene datoteke itd.
- *Nestrukturirani podatkovni model* vsebuje vse vrste podatkov, ki nimajo določene strukture. Tak primer so slike in ostale grafike, digitalni dokumenti itd. Vsi nestrukturirani podatki so tudi pol strukturirani, ker čeprav vsebujejo množico bitov brez strukture kot take, datoteka kot celota vsebuje strukturo, ki jo je moč prebrati in spreminjati.

### 3.2.2 Koncepti v socialnih omrežjih

Spletni servisi zagotavljajo delovanje programskih vmesnikov API na oddaljenih sistemih. Z njihovo uporabo poteka programska integracija podatkov iz raznovrstnih informacijskih sistemov. Tehnični izziv integracije se torej rešuje z uporabo spletnih servisov s standardizacijo struktur za izmenjavo podatkov. Komunikacija med sistemi poteka z izmenjavo sporočil, največkrat v XML obliki z uporabo komunikacijskega protokola SOAP ali programske arhitekture REST. XML omogoča večjo fleksibilnost vseh vrst podatkov, ki se obdelujejo.

Poznamo različne modele za združevanje podatkov z uporabo spletnih servisov:

- *vsebinsko združevanje* zbira vsebino, ki se nanaša na določeno temo, iz različnih virov in s tem zagotavlja boljšo povezanost med več viri podatkov;
- *primerjalno združevanje* na podlagi določenih meril uporabnikov primerja rezultate različnih področij in s tem zagotavlja optimalne rezultate;
- *združevanje na podlagi razmerij* zagotavlja skupno točko med uporabnikom in različnimi viri informacij, s katerimi je uporabnik povezan.

Izzivi:

- zagotoviti uporabniku varnost pri uporabi spletnih storitev,
- kakovostni, natančni, dosledni in pravilni podatki, ki jih spletni servisi proizvedejo,
- črpanje pomena besed iz teksta je največji izziv za računalniške programe. Semantični splet predstavlja mehanizem za objavljanje informacij na način, ki omogoča lažje razumevanje in interpretacijo s strani računalnika [8].

### 3.3 Prijava v socialna omrežja

Problem integracije podatkov iz različnih socialnih omrežij je proces zbiranja podatkov iz več socialnih omrežij na eno lokacijo oziroma združitev več uporabniških profilov v enega. Integracija je izvedena s pomočjo programskih vmesnikov API. Da lahko vmesnik API dostopa do uporabniških podatkov, mora uporabnik odobriti dostop do storitev, tako da se prijavi s pripadajočim uporabniškim imenom in geslom, sinhroniziranim iz določenega socialnega omrežja. Na spletno mesto se doda povezava za prijavo, ki uporabnika preusmeri na socialno omrežje. Nato se bodisi prijavi ali registrira in odobri aplikaciji dostop do njegovih podatkov. Na koncu je preusmerjen nazaj na prvotno spletno mesto. Tako lahko začne aplikacija uporabljati oziroma obdelovati njegove uporabniške podatke. Za tak način prijave večinoma skrbi avtentikacijski protokol, ki je prvi korak v integraciji socialnih omrežij. Glavni namen je uporaba zgolj enega uporabniškega imena in gesla za prijavo na več socialnih omrežij. Tako se uporabnikom na različne spletne strani ni treba prijavljati z različnimi uporabniškimi računi, temveč zadostuje enotno uporabniško ime in geslo. Glavna predstavnika na tem področju sta OpenId in Facebook Connect, z njimi tesno povezan je avtentikacijski protokol OAuth. Še korak dlje gre Google OpenSocial, katerega namen je enoten programski model za uporabo API-jev.

#### 3.3.1 Google OpenSocial

OpenSocial je razvojna platforma, ki je nastala v sodelovanju z Myspacom in je namenjena socialnim omrežjem. V splošnem je zbirka programskih vmesnikov API za izdelavo socialnih aplikacij na tretjih straneh. Deluje na osnovi JavaScript API-jev in REST arhitekture oziroma njenih protokolov za interakcijo med strežniki.

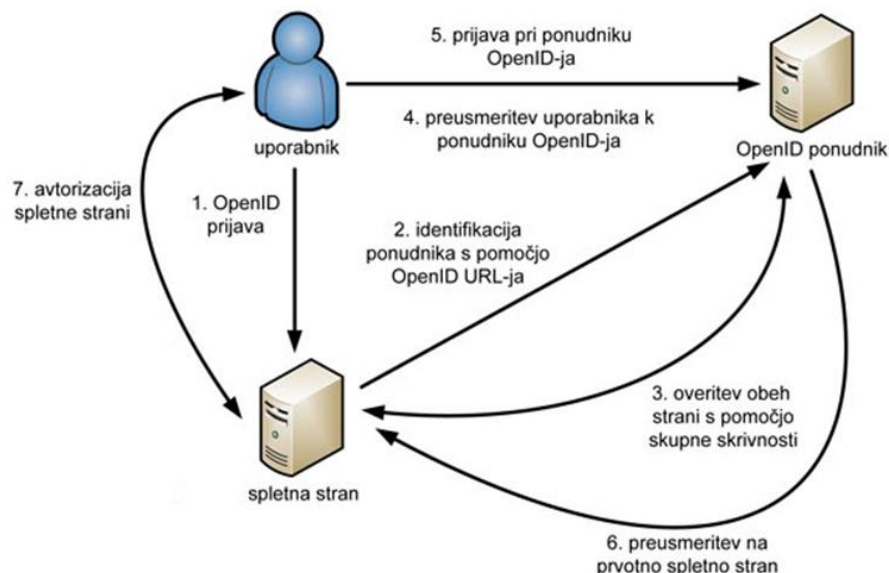
Glavno vodilo je enoten programski model, ki je v pomoč razvijalcem in spletnim mestom. Razvijalci se morajo tako naučiti vmesnik API enkrat, da ga lahko uporabljajo na vseh spletnih straneh, ki OpenSocial podpirajo. Ker je lahko OpenSocial implementiran na katero koli spletno stran, je lažje priti do ciljnih uporabnikov. Obstaja tudi večja možnost sodelovanja razvijalcev med seboj, glede na to da so API-ji standardizirani.

Aplikacije se lahko prikazujejo z uporabo Googlovih dodatkov (*angl. gadget*) in se izvajajo tako na odjemalcu kot na strežniku. Potek dogodkov krmili aplikacija sama, tako da komunicira direktno z brskalnikom. V aplikacijah na strani odjemalca komunikacija poteka z uporabo XML dokumentov in uradnih JavaScript knjižnic. Prav uporaba uveljavljenih standardov HTML, JavaScript in XML daje prednost pri razvoju, saj se ni potrebno učiti novih programskih jezikov [9].

### 3.3.2 OpenID

OpenID je odprtokodni projekt, kjer je mogoče uporabniški račun pridobiti prek ponudnikov, kot so MyOpenID.com ali LiveJournal.com. Na voljo je tudi slovenski ponudnik mojID.com.

Uporabnik se prijavi na spletno stran, ki podpira OpenID, tako da vnese svoj OpenID v obliki URL-ja: mojid.com/diploma (1). Spletna stran ob pomoči OpenID v obliki URL-ja identificira ponudnika OpenID prek protokola YADIS, v našem primeru mojid.com (2). Obe strani ustvarita skupno skrivnost (ob pomoči Diffie-Hellmanovega protokola za izmenjavo ključev) in tako overita druga drugo. Zagotovita, da sta spletna stran in ponudnik OpenID res tista, za katera se predstavljata (3). Spletna stran uporabnika preusmeri k ponudniku OpenID in tam se ta prijavi (4), (5) ter avtorizira spletno stran, da pridobi od ponudnika uporabniško ime in morebitne druge podatke (el. naslov, ime, priimek, naslov, ...). Na koncu ponudnik uporabnika preusmeri na prvotno spletno stran (6), (7) [10].



Slika 2: Prijava s pomočjo OpenID računa.

### 3.3.3 Facebook Connect

Facebook Connect omogoča enotno prijavo na zunanje spletne strani s pomočjo že obstoječega Facebook računa. Ob prijavi se lahko z uporabnikovim dovoljenjem posredujejo tudi drugi uporabniški podatki. Te nastavitve dovoljenj lahko kadarkoli spremeni v svojem Facebook računu. S tem ima dobro kontrolo nad tem, kaj lahko aplikacija počne z njegovimi podatki oziroma kdo jih lahko vidi in kdo ne. Prav tako lahko vidi, kateri prijatelji že uporabljajo Facebook Connect na posamezni spletni strani.

### 3.3.4 OAuth

OAuth je avtentikacijski protokol, ki ga uporabljajo nekatera pomembnejša socialna omrežja. Predvsem od aprila 2010, ko ga je začel uporabljati Facebook, kot socialno omrežje z največ uporabniki, se je pokazala možnost, da bi OAuth postal glavni avtentikacijski protokol.

Delovanje OAuth protokola je najlažje pojasniti na resničnem primeru, zato bom predpostavil, da si želi uporabnik Janez na svojo spletno stran (diploma.si) prenesti in prikazati sliko profila, ki jo uporablja na socialnem omrežju Facebook. Veljajo naslednje predpostavke:

- Janez je uporabnik (*angl. user*).
- Ponudnika storitev predstavlja spletna stran diploma.si (*angl. service provider*).
- Slika profila, ki jo želimo prikazati na spletni strani diploma.si oziroma prenesti s strani facebook.com, se imenuje zaščiten vir (*angl. protected resources*).
- Spletna stran facebook.com predstavlja potrošnika (*angl. consumer*).
- S tem ko spletna stran diploma.si ponuja možnost prenosa slike iz facebook.com, jo lahko poimenujemo potrošnik-razvijalec (*angl. consumer developer*). Predhodno mora registrirati aplikacijo na spletni strani facebook.com in s tem pridobi potrošniški ključ (*angl. consumer key*) in potrošniško skrivnost (*angl. consumer secret*). Potrebuje ju namreč za dostop do OAuth podprtega vmesnika API spletne strani facebook.com.

Potek dogodkov je naslednji:

- Preden je uporabnik Janez preusmerjen na spletno stran facebook.com, da bi na njej vnesel svoje registracijske podatke, se na spletno stran facebook.com pošlje zahteva po povpraševalnem žetonu (*angl. request token*).
- Spletna stran diploma.si ob pridobitvi povpraševalnega žetona preusmeri Janeza na spletno stran facebook.com, kjer se na osnovi povpraševalnega žetona preveri avtentičnost povezave.
- Ko Janez vnese pravilne podatke za prijavo in nato še potrdi dostop do njegovih uporabniških podatkov, v tem primeru do slike profila, se povpraševalni žeton označi kot uporabniško avtoriziran in preusmeri Janeza skupaj z avtoriziranim povpraševalnim žetonom nazaj na diploma.si.
- Medtem ko Janez čaka na prenos slike profila diploma.si, uporabi avtoriziran povpraševalni žeton za zamenjavo z žetonom za dostop (*angl. access token*). Povpraševalni žetoni so namreč namenjeni pridobivanju uporabniških dovoljenj, žetoni za dostop pa omogočajo dostop do zaščitenih virov.
- Spletna stran diploma.si torej na facebook.com pošlje zahtevo, v kateri se zamenja povpraševalne žetone z žetoni za dostop in tako dovoli uporabo slike profila.

## 3.4 Pregled programskih rešitev za integracijo podatkov v socialnih omrežjih

Programi za integracijo podatkov v socialnih omrežjih se največkrat soočajo s težavo velike količine podatkov in z malo časa, ki ga imajo na voljo za obdelavo. Že sami klici po večini

katerihkoli vmesnikov API so časovno dokaj potratni. Presenetljivo je dejstvo, da obstaja že kar veliko število rešitev za razrešitev obravnavanega problema. Na voljo so v obliki spletnih strani, namiznih aplikacij itd. Vseeno nekega zares dobrega in prevladujočega orodja na trgu trenutno ni. Obstaja več vrst rešitev, ki so vsaka zase usmerjene in jih bom predstavil v nadaljevanju. Njihove splošne lastnosti oziroma usmerjenosti so po zaporedju naslednje:

- en uporabnik in večje število socialnih omrežij,
- več različnih uporabnikov, prijavljenih v isto socialno omrežje naenkrat,
- integracija podatkov socialnih omrežij v obliki spletnega brskalnika,
- posodabljanje stanj na več socialnih omrežjih skupaj,
- iskanje in analiza po socialnih omrežjih.

### 3.4.1 *TweetDeck*

Kot že samo ime pove, je TweetDeck v prvi vrsti namenjen za Twitter. Poleg združuje še Facebook, LinkedIn, Google Buzz, Foursquare in MySpace socialna omrežja.

Samo delovanje temelji na Twitter vmesnikih API, ki omogočajo uporabnikom pošiljanje in prejemanje sporočil ter ogled profilov. Deluje na večini najbolj uporabljenih operacijskih sistemov: Windows, mac OS in Linux. Obstajajo tudi verzije za iPhone, iPad in Android. Vse od maja 2011 je TweetDeck v lasti Twitterja, za kar je moral odšteti 30 milijonov evrov. Uporabniki lahko svoj prikaz razdelijo na stolpce in jim določijo poljubno vsebino. V enem stolpcu lahko na primer spremljajo posodobitve statusov z več socialnih omrežij, ustrezno integrirane in časovno razporejene. Prav tako lahko svoje prijatelje razporejajo v skupine, kar se je izkazalo za zelo uporabno. Orodje postane zares uporabno šele pri večjem številu sledilcev, saj urejeno pregledovanje objav šele takrat zares pride do izraza [12].

### 3.4.2 *Yoono*

Yoono velja za enega izmed najbolj priljubljenih tovrstnih programov, ki ga je moč brezplačno prenesti iz uradne spletne strani. Obstajajo različice za mobilni telefon, za spletni brskalnik ali v obliki namizne aplikacije.

Podpira dodajanje računov iz Facebooka, Twitterja, YouTubea, Flicka, MySpaca, Linkedlna, Foursquara, FriendFeeda, Yammerja, AIM-a, Yahoo! Messengerja in Google talka.

Vse najnovejše dogajanje iz različnih servisov se lahko spremlja v enem stolpcu. Po želji pa se lahko dogajanje na posameznem servisu ogleda tudi v svojem ločenem stolpcu. Zelo dobrodošla funkcionalnost programa je tudi ta, da lahko uporabnik sam nastavi, katere vsebine iz posameznega servisa naj se prikazujejo. Tako se lahko brez težav znebi nezanimivega izpisovanja najnovejših podvigov prijateljev pri igranju Facebookovih iger.

Program Yoono ima pred nekaterimi ostalimi konkurenčnimi programi tudi eno zelo dobrodošlo prednost. Omogoča namreč, da se lahko vanj vpišemo z več računi iz istega servisa.

### 3.4.3 *RockMelt*

RockMelt je socialni spletni brskalnik, podprt s strani Marca Andreessena, ustanovitelja včasih popularnega brskalnika Netscape. Posebnega poudarka sta deležni socialni omrežji Facebook in Twitter. Brskalnik je na voljo brezplačno, pred prvo uporabo je potrebno vzpostaviti povezavo s Facebook računom. Zasnovan je na osnovi Googlovega odprtokodnega projekta Chromium. Na levi strani se nahaja seznam kontaktov. Z njimi je možna komunikacija, ne da bi odprli nov zavihek ali dostopali do dejanske strani socialnega omrežja. Uporabnik lahko spreminja status, prikazno sliko, spremlja statuse prijateljev in ustvarja lastne sezname stikov. Na desni strani brskalnika se v stolpcu nahaja seznam podprtih socialnih omrežij, ki obvešča o morebitnih novih objavah in spremembah. Brskalnik vsebuje tudi Google iskalnik. Rezultati iskanja se prikazujejo v majhnem in dobro urejenem padajočem seznamu, tako da ni potrebe po odpiranju v novem zavihku.

### 3.4.4 *Ping.fm*

Spletno stran ping.fm bi lahko poimenovali kar posodobitelj statusov oziroma stanj. Omogoča namreč posodobitev statusa na vseh prijavljenih socialnih omrežjih naenkrat. Trenutno jih je že preko trideset. Statuse se lahko posodablja preko omenjene spletne strani ali preko dodatka, ki se integrira v brskalnik Firefox [12].

### 3.4.5 *Social Mention*

Social mention je orodje za iskanje in analizo po socialnih omrežjih v realnem času. Na podlagi iskanega niza pregleda preko 140 socialnih omrežij in združi rezultate na eno lokacijo. Namenjeno je predvsem v poslovne namene, za uporabno se izkaže tudi pri iskanju novic in podatkov o posamezni osebi. Samo iskanje bi lahko razdelili na osnovno in napredno. V osnovnem iskanju lahko iščemo glede na tip socialnega omrežja (blog, mikroblog, video itd.) ali po točno določeni platformi (Facebook, Twitter itd.). Napredno iskanje omogoča iskanje z več besedami ali celotnimi frazami, za kar se uporablja narekovaje. Iz rezultatov iskanja lahko predhodno odstranimo nezaželene besede in celo uporabnike. Podprte so še nekatere omejitve iskanja po lokaciji, času in jeziku. Rezultati iskanja so lahko razporejeni po datumu in omejeni po času. Za vsak rezultat iskanja je prikazana platforma z največ rezultati, na kateri je bil ustvarjen, kdaj in kdo je objavo ustvaril, kdo je največkrat komentiral. Vidimo lahko, koliko različnih oseb je komentiralo določeno temo in koliko odgovorov je bila deležna. Rezultati iskanja so lahko razporejeni glede na število komentarjev

ter ali so komentarji pozitivno oziroma negativno ocenjeni. Social mention ponuja možnost kreiranja opozoril za določene teme ali fraze. Po predhodno opisanih metodah uporabnik določi parametre iskanja in nato prejema obvestila o novih objavah na svoj e-mail naslov [13].

## 4 INTEGRACIJA SOCIALNIH OMREŽIJ Z UPORABO PROGRAMSKIH VMESNIKOV

### 4.1 Spletni servis

Spletni servis je po standardu W3C (*angl. World Wide Web Consortium*) formalno opredeljen kot programski sistem, ki ga identificira ustrezen URI (*angl. Uniform Resource Identifier*) ter katerega vmesnik in povezave so opredeljeni v jeziku XML. Opredelitev spletnega servisa lahko druga programska okolja odkrijejo s posebnim razpoznavnim servisom UDDI (*angl. Universal Description, Discovery and Integration*). Takšni sistemi oziroma uporabniki lahko uporabljajo spletni servis na način, ki je podan v njegovi opredelitvi in z uporabo na XML temelječih sporočil, ki jih posreduje mrežni protokol http. Takšen na XML temelječ jezik za označevanje omogoča funkcionalnost in veliko izrazno moč posredovanih sporočil [16].

Spletni servisi so aplikacije, ki se izvajajo na različnih strojnih platformah, operacijskih sistemih in so napisane v različnih programskih jezikih, da dinamično izmenjujejo informacije med seboj. Vse to je mogoče, ker pošiljanje in prejemanje informacij poteka v konsistentnih formatih, navadno XML. Spletni servisi znajo sestaviti XML, ki predstavlja servisno zahtevo aplikacije. Drugače povedano, lahko razčlenijo XML zahtevo, ki prihaja od neke aplikacije in glede na svojo funkcionalnost vrnejo odgovor na zahtevo.

Spletni servis pomeni spremembo oziroma nadgradnjo aplikacije ali uporabniškega programa v spletno aplikacijo, ki lahko ponudi ustrezen storitev drugim uporabnikom spleta ali medmrežja. Takšna storitev mora biti varna, kakovostna in ustrezno vodena oziroma upravljana. Spletni servis je dejansko vmesna programska oprema, ki zagotavlja medopravilnost ter služi za povezovanje, komunikacijo in izmenjavo podatkov. Lahko bi rekli, da imajo spletni servisi na medmrežju podobno vlogo kot vmesna oprema v lokalnih omrežjih.

Osnovne značilnosti spletnih servisov so:

- *ponovna uporaba aplikacij* – obstajajo mnoge funkcionalnosti, ki se v aplikacijah pogosto uporabljajo, na primer menjava valut, napoved vremena, prevodi med jeziki itd. Zato se lahko takšne dejavnosti tipizirajo in nato večkrat uporabijo;
- *povezovanje obstoječe programske opreme* – spletni servisi lahko pomagajo pri zagotavljanju raznih stopenj medopravilnosti, tako da omogočajo povezavo in izmenjavo podatkov med različnimi spletnimi aplikacijami, bazami podatkov itd.;
- komunikacija preko odprtih protokolov (TCP/IP, http itd.);
- najdejo se z imenikom UDDI in iskalnim servisom ter z uporabo ustreznega opisa WSDL (*angl. Web Services Description Language*).

## 4.2 (Spletni) API

Vmesnik API je skupek podatkovnih struktur, protokolov in orodij za dostop do spletno orientiranih programskih aplikacij. Vsebuje elemente za lažji razvoj aplikacij. Programski vmesnik je bodisi odvisen od programskega jezika bodisi ne.

Vmesniki API lahko bistveno prihranijo čas pri samem razvoju. Z izdelavo aplikacij ni potrebno začeti od začetka. S strani spletnih razvijalcev je spletni vmesnik API tipično definiran kot množica zahtev http in odgovorov, navadno izraženih s pomočjo sporočil na osnovi XML ali JSON notacije.

Vmesnik API določa zakonitosti in postopke, ki so potrebni, da lahko aplikacije komunicirajo med seboj. Specificirajo vrednosti, ki jih aplikacije potrebujejo za dokončanje opravil, transakcij in drugih procesov. Aplikaciji, ki kliče vmesnik API, za dokončanje opravila ni potrebno podrobno poznati, kako sam vmesnik API deluje.

Za vmesnike API bi lahko rekli, da se tiho izvajajo v ozadju. Za uporabnike programske opreme in spletne obiskovalce so popolnoma nevidni. Njihova osnovna funkcija je ustvariti povezavo med aplikacijami tako, da se skladno izvajajo in da uporabnik dobi zahtevane funkcionalnosti in informacije.

Za primer vzemimo nakup izdelka v spletni trgovini. Ko uporabnik kupuje produkt in vnese podatke o kreditni kartici, spletna stran uporabi vmesnik API za pošiljanje podatkov o kreditni kartici neki drugi aplikaciji. Nato se preveri, ali so vneseni podatki pravilni in opravi se postopek plačila. Prvotna spletna aplikacija prejme odgovor o uspešnem ali neuspešnem postopku plačila.

Spletni vmesnik API spada pod okrilje spletnih servisov. Največkrat jih srečamo v povezavi s tehnologijami Web 2.0. Poudarek ni več na do sedaj prevladujočih servisih SOAP, temveč se nagiba na arhitekturo REST. Spletni vmesnik API omogoča uporabo več spletnih servisov v eni aplikaciji, imenovani mashup [17].

## 4.3 REST

REST (*angl. Representational state transfer*) bi lahko poimenovali kar arhitektura izgradnje spletnih aplikacij, ki je pri svojem izvajanju neodvisen od platforme in od programskega jezika. Ideja arhitekture je nadomestiti obstoječe kompleksne mehanizme za komunikacijo med napravami, kot je na primer SOAP, z bolj preprostejšimi zgolj z uporabo protokola http. Lahko bi rekli, da kar sam svetovni splet (WWW) temelji na arhitekturi REST. V definicijo so vključeni pojmi:

- *Naslovljivi viri*

- Ključni odvzem informacij in podatkov poteka s pomočjo virov. Vsak vir mora biti naslovljiv preko URI-ja.
- *Enoten in omejen vmesnik*
  - Uporablja se majhno skupino dobro definiranih metod za manipulacijo in upravljanje z viri.
- *Predstavitvena usmerjenost*
  - Viri, usmerjeni z uporabo URI, so lahko v različnih formatih. Različne platforme potrebujejo različne formate. Za primer, brskalniki potrebujejo HTML, JavaScript pa JSON.
- *Komunikacijska neodvisnost*
  - Strežnik si ne zapomni vsake odjemalčeve zahteve. Vse informacije za izvršitev zahteve so vključene v samo zahtevo. Taki tipi aplikacij so lažji za upravljanje.

REST spletne storitve se osredotočajo na izpostavljanje podatkov. Naslov URI poda, katere podatke želimo, da nam spletna storitev vrne ali sprejme. Vrnjen ali sprejet dokument se smatra za trenutno stanje tega vira. Za interakcijo s spletnim strežnikom, na katerem teče storitev, se v praksi uporablja protokol http s svojimi ukazi. To pa zagotovi podporo na skoraj vseh platformah, ki so sposobne uporabljati svetovni splet. Nabor operacij, ki omogočajo pridobitev, pošiljanje, posodabljanje in brisanje podatkov, so v primeru uporabe protokola http kar njegovi ukazi GET, PUT, POST, DELETE [16].

Uporaba REST spletnih storitev je smotna, če imamo potrebo po spletnem vmesniku in če želimo do storitev dostopati iz najrazličnejših naprav – od mobilnih naprav do večjih strežnikov. Vse naprave, ki imajo dostop do svetovnega spleta, imajo namreč tudi dostop do osnovnega nabora metod http.

Ena izmed pomanjkljivosti arhitekture REST je po mnenju nekaterih odsotnost jezika WSDL, s katerim bi opisali seznam in strukturo spletnih storitev. Opazka pa ni povsem na mestu, saj jezik WSDL verzije 2.0 že omogoča opis REST spletnih storitev. Težava je v tem, da je trenutno v splošni uporabi predvsem prva verzija jezika WSDL. Še vedno pa lahko za opis naših virov v XML jeziku uporabimo vse pripadajoče standarde, npr. tehnologiji DTD in shemo XML [14].

#### 4.3.1 *REST in http*

Kljub temu da REST ni protokol, se ga največkrat omenja v povezavi s protokolom http. V grobem bi lahko rekli, da je to mrežni aplikacijski protokol za izvajanje zahtev in odgovorov. Pri uporabnikih gre pri učenju REST arhitekture bolj za poglobljanje znanja iz protokola http kot pa za način razvoja aplikacij. Pri tehnologijah, različnih od REST, kot je na primer SOAP in drugi spletni servisi, se http uporablja zgolj za transportni protokol in tako se pokaže samo majhen del njegovih zmogljivosti. Marsikdo bi celo rekel, da je njegov glavni razlog uporabe

zgolj za prehod skozi požarne zidove. Http je namreč bogat aplikacijski protokol, ki ponuja uporabne in zmogljive funkcionalnosti za razvijalce. V osnovi je namenjen za uporabo v spletu, posebej za spletne brskalnike.

Primer zahteve http:

```
GET /index.html HTTP/1.1
Host: diploma.si
User-Agent: Mozilla/5.0
Accept: text/html
Accept-Language: en-us,en;
Accept-Encoding: gzip
```

Na zgornjem primeru je pomen posameznih vrstic naslednji:

- brskalnik pošlje zahtevo na naslov <http://diploma.si/index.html>,
- GET je metoda, ki jo pošiljamo strežniku,
- index.html je objekt, ki nas zanima,
- http/1.1 je verzija protokola,
- host, userAgent, accept, accept-language in accept-encoding so sporočila glav (*angl. header*).

Odgovor strežnika je dokaj podoben. Vsebuje verzijo protokola http, ki ga uporabljamo, statusno kodo, kratko sporočilo, ki pojasnjuje statusno kodo, skupino spremenljivk, ki določajo glave in opsijsko sporočilo. Odgovor na prejšnjo zahtevo http je naslednji:

```
HTTP/1.1 200 OK
Content-Type: text/html
<head>
  <title>Primer http odgovora</title>
</head>
<body>
  <p>Vsebina http odgovora je...
```

Statusna koda 200 v sporočilu sporoča, da je vse v redu oziroma da ni prišlo do napak. Pomeni, da se je zahteva uspešno izvedla in da je uporabnik prejel informacije, ki so ga zanimale. Sicer ima http daljši seznam statusnih kod. Naj še omenim statusno kodo 500, ki sporoča, da je prišlo do napake [15].

#### 4.3.2 HTTP metode

Http ima majhno, omejeno število metod. Vsaka ima svoj učinek in namen. Naj omenim tiste, ki se jih uporablja pri vmesnikih API socialnih omrežij:

- *GET* – je operacija, namenjena branju. Uporablja se za pridobivanje specifičnih informacij od strežnika. Ne glede kolikokrat izvedemo operacijo, bo rezultat vedno

isti. Uvrščamo jo med varne operacije. Klicanje metode GET namreč ne spremeni stanja strežnika.

- *PUT* – metoda pošlje zahtevo, da strežnik shrani vsebino sporočila na zahtevano lokacijo, predvideno s strani zahteve http. Navadno je metoda realizirana kot operacija vstavljanja ali posodabljanja. Pri klicanju metode je identifikacija vira, ki ga posodabljam ali ustvarjam, znana. Posledično je pošiljanje sporočila PUT z istimi parametri več kot enkrat brez efekta. Za primerjavo bi lahko vzeli MS Word dokument, kjer je, ne glede kolikokrat pritisnemo gumb za shranjevanje, datoteka, v katero se dokument shrani, vedno ista.
- *DELETE* – se uporablja za brisanje virov.
- *POST* – velja za eno izmed manj varnih operacij protokola http. Uporablja se za vstavljanje in posodabljanje virov.
- *HEAD* – ima iste funkcionalnosti kot GET s to razliko, da odgovor na zahtevo vrne le statusno kodo in glave, povezane z zahtevo.

### 4.3.3 REST arhitektura

Vzemimo za primer preprosto situacijo, kjer poznamo uporabnikov id in želimo dobiti več podatkov o njem. Najprej si pogledjmo, kako bi sestavili zahtevo z uporabo SOAP:

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:body pb="http://www.diploma.si/uporabnik/">
    <pb:GetUserDetails>
      <pb:UserID>12345</pb:UserID>
    </pb:GetUserDetails>
  </soap:Body>
</soap:Envelope>
```

REST spletne storitve temeljijo na virih (*angl. resources*). Osnovni koncept REST modela so viri s svojimi identifikatorji. Identifikatorji so npr. spletne strani, pa tudi sporočila novičarskih skupin ali datoteke na FTP strežnikih. Edini pogoj za obstoj identifikatorja je, da ga lahko opišemo kot edinstveni identifikator.

Naslavljanje je način, s katerim je vsak objekt in vir v sistemu dosegljiv preko unikatnega identifikatorja. REST naslavljanje upravlja s pomočjo URI-jev. Vsakokrat, ko naredimo zahtevo po informacijah v brskalniku, pišemo URI. Vsaka zahteva http mora vsebovati URI objekta, o katerem iščemo ali objavljamo informacije. Format URI je podan v nadaljevanju:

```
scheme://host:port/path?queryString#fragment
```

- Scheme je protokol, ki ga uporabljamo za komunikacijo. Kot sem že omenil, je za REST to http ali https.

- Host je ime DNS ali IP naslov. Po potrebi se naprej doda port, ki je numerična vrednost. Host in port predstavljata lokacijo vira v omrežju.
- Path predstavlja lokacijo na strežniku.
- Po znaku ? sledijo parametri poizvedbe. V kolikor jih je več, so ločeni z znakom &.
- Zadnji del URI-ja je fragment, ki je ločen z znakom #. Največkrat se uporablja za pozicioniranje na določeno mesto v dokumentu, ki ga poizvedujemo.

Praktičen REST primer z uporabo URI poizvedbe:

```
http://www.diploma.si/uporabnik/podrobnosti/12345
```

URL naslov se pošlje strežniku z uporabo zahteve GET, od koder sledi odgovor http. Ena izmed prednosti REST je, da lahko zahteve neposredno testiramo zgolj z uporabo spletnega brskalnika.

V prejšnjem primeru sem predstavil preprost primer zahteve REST, z enim parametrom. Uporabimo lahko tudi bolj kompleksne zahteve, z uporabo več parametrov. V večini primerov jih podamo preko http metode GET.

```
http://www.diploma.si/uporabnik/podrobnosti?ime=janez&priimek=sorta
```

V kolikor gre za še daljše parametre, uporabimo http metodo POST. Kot pravilo naj bi veljalo, da se metoda GET uporablja zgolj za branje, POST poleg branja združuje še kreiranje in posodabljanje.

Praktičen primer odgovora strežnika v XML formatu:

```
<user>
  <id>1117556503</id>
  <name>Janez Sorta</name>
  <first_name>Janez</first_name>
  <last_name>Sorta</last_name>
  <link>http://www.facebook.com/profile.php?id=1117556503</link>
  <birthday>12/02/1988</birthday>
  <gender>male</gender>
  <timezone>2</timezone>
  <locale>sl_SI</locale>
  <verified>true</verified>
  <updated_time>2011-09-02T12:41:45+0000</updated_time>
  <type>user</type>
</user>
```

Seveda so lahko uporabljeni tudi nekateri drugi formati. Za razliko od SOAP, REST ni omejen zgolj na XML. Med preostalimi jeziki za odgovor prevladuje predvsem JSON. Vsak ima svoje prednosti in slabosti. XML velja za lahko razširljivega in varnega. JSON pa je priljubljen predvsem zaradi preprostega razčlenjevanja v JavaScript [14].

#### 4.3.4 Varnost

Ustrezno lahko implementiramo varnost REST spletnih storitev. Uporabimo lahko več vrst avtentikacije in avtorizacije, npr. Basic access authorization ali avtentikacijo s certifikatom. Varno povezavo z REST spletnimi storitvami vzpostavljamo preko protokola https.

Avtentikacija temelji na potrjevanju in identifikaciji odjemalcev, ki želijo dostopati do servisa. Največkrat gre za preverjanje uporabnika z uporabniškim imenom in geslom, z nekaterimi standardiziranimi protokoli, ki se uporabljajo pri avtentikaciji.

Ko je uporabnik enkrat avtenticiran, bo želel uporabljati REST servis. Avtorizacija, ki ni del kakršnega koli internetnega protokola, preverja, ali lahko posamezen uporabnik dostopa in pošilja zahteve na določen URI. Na primer, za eno skupino uporabnikov želimo omogočiti pisanje (PUT/POST/DELETE), za drugo pa ne.

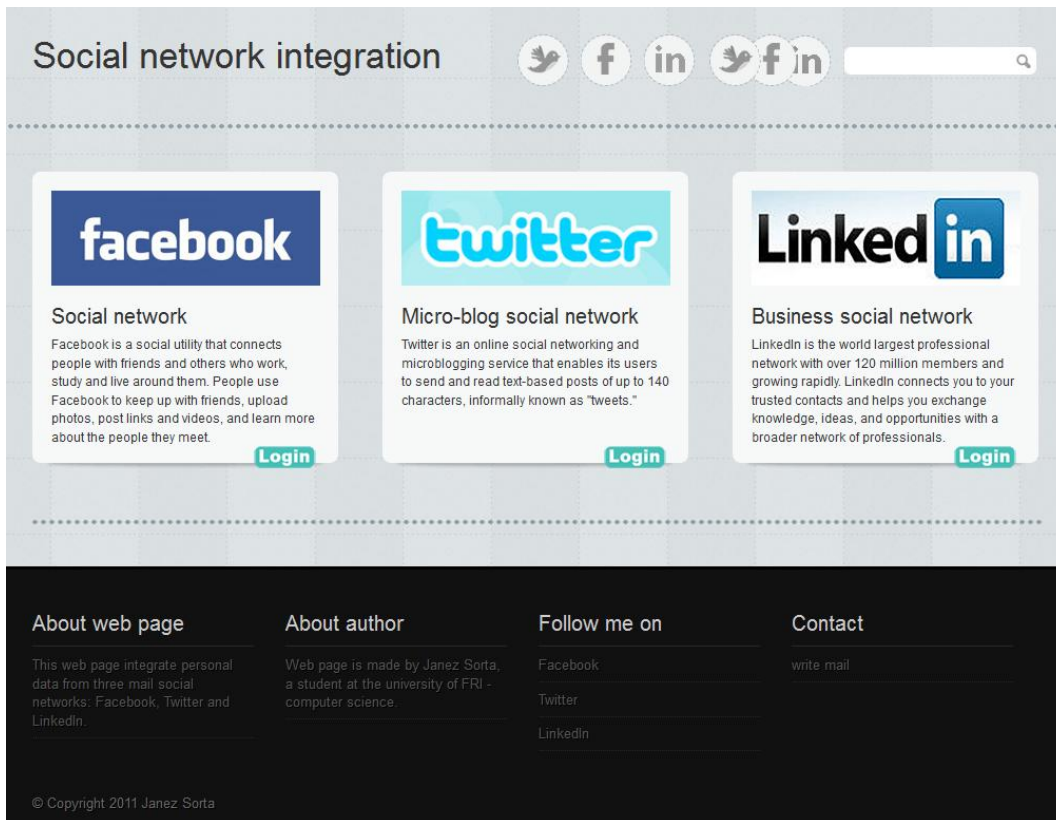
Medtem ko uporabnik komunicira s spletnim servisom, je mogoče prestreči pakete ter brati zahteve in odgovore, v kolikor http povezava ni zavarovana. Podatki so lahko varovani z enkripcijskimi servisi, kot je na primer SSL.

## 5 PROGRAMSKA PODPORA ZA IZVEDBO INTEGRACIJE

V okviru praktičnega dela diplomske naloge sem izdelal spletno aplikacijo, katere glavni namen je integracija osebnih podatkov uporabnikov treh vodilnih socialnih omrežij Facebook, Twitter in LinkedIn. Uporabnik izbere določeno storitev, vpiše prijavnne podatke zanjo in aplikacija nove podatke ustrezno združi s podatki, ki jih že ima na voljo. Na primer, če ugotovi, da je uporabnik z isto osebo prijatelj na Facebooku in LinkedInu, združi podatke o njej in jih prikaže. Glede na obsežnost podatkov, ki jih ponujajo socialna omrežja, sem se v okviru diplomske naloge v aplikaciji odločil obravnavati le osebne podatke uporabnikov. Komunikacija in pridobivanje podatkov poteka s pomočjo programskih vmesnikov API, ki jih bom posamezno predstavil v nadaljevanju. Predhodno sem si zastavil potek razvoja aplikacije, ki sem ga razdelil na več delov:

- *Prijava v posamezno socialno omrežje.* Na podlagi avtentikacijskega protokola OAuth se uporabnik prijavi na vsa tri obravnavana socialna omrežja.
- *Prikaz seznama prijateljev po posameznem socialnem omrežju.* Po uspešni prijavi lahko uporabnik pregleduje seznam prijateljev, sledilcev oziroma kontaktov.
- *Prikaz osebnih podatkov o posameznem prijatelju.* Izpišejo se vsi osebni podatki o posamezniku, katere se le-ta odloči deliti z ostalimi.
- *Združevanje skupnih prijateljev z več socialnih omrežij.* Predhodno mora biti uporabnik prijavljen v vsaj dve socialni omrežji. Skupni prijatelji se združijo in prikažejo v enotnem seznamu.
- *Združevanje vseh razpoložljivih osebnih podatkov posameznega prijatelja z več socialnih omrežij.* Spet mora biti uporabnik prijavljen v vsaj dve socialni omrežji. Vsi razpoložljivi osebni podatki se po določenih zakonitostih, ki bodo predstavljena v nadaljevanju, združijo v smiselno celoto.
- *Iskalnik prijateljev.* Iskanje po predhodno sestavljenem seznamu.

Uvodna stran je prikazana na sliki 3, sicer je na svetovnem spletu dostopna preko naslova URL: <http://www.janezsorta.si>.

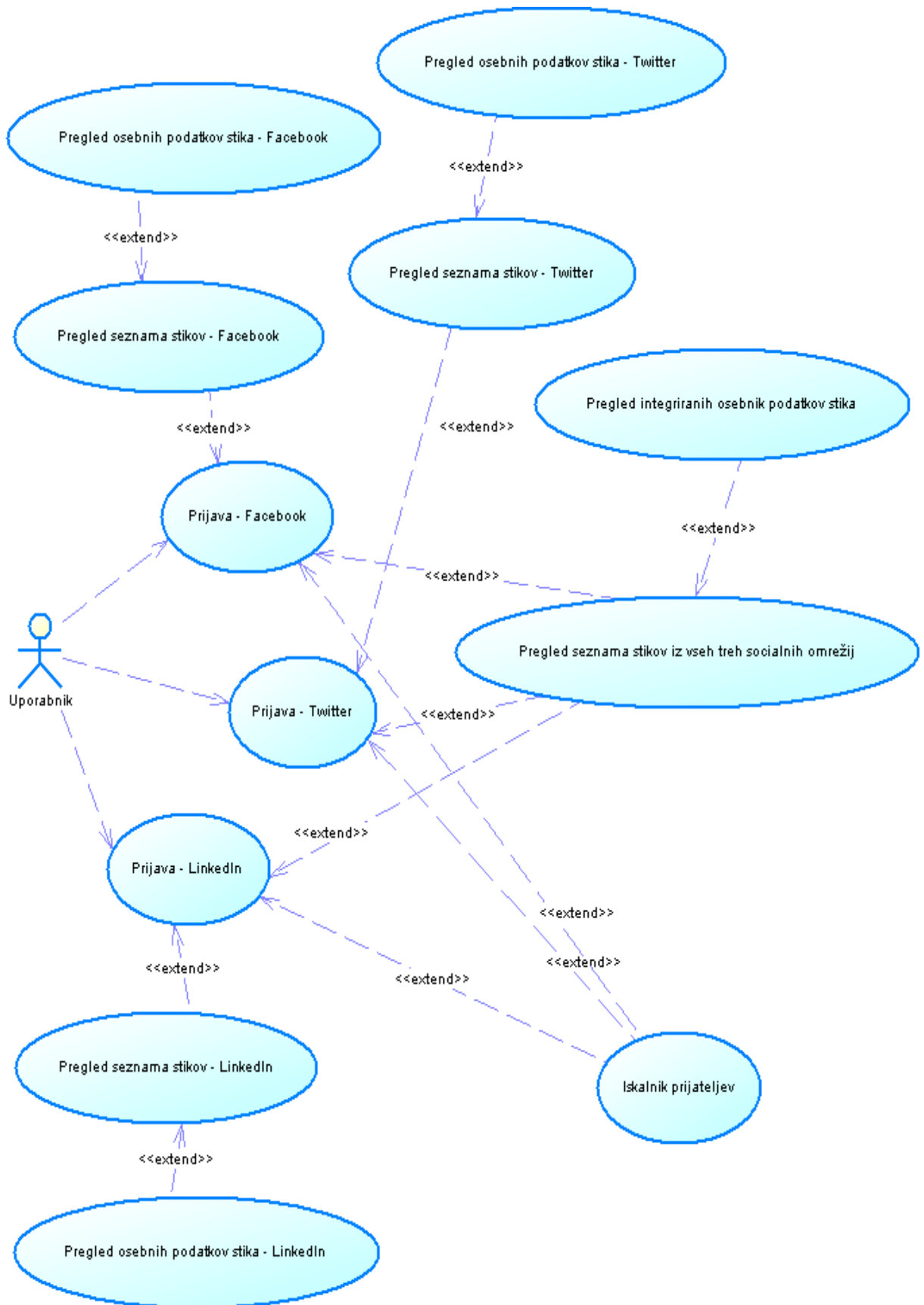


Slika 3: Uvodna stran izdelane spletne aplikacije.

Za lažjo navigacijo po spletni aplikaciji in samo uporabo sem se odločil vključiti že izdelano predlogo, ki sicer za samo delovanje nima bistvenega pomena. Predloga je na voljo na URL naslovu <http://www.site5.com/wordpress-themes/> in je izdelana z uporabo označevalnega jezika HTML ter stilskih predlog CSS, v skladu z oblikovnimi standardi spleta druge generacije. Glede na moje potrebe sem bil primoran predlogo nekoliko spremeniti. Za razvoj spletne aplikacije sem uporabil skriptni programski jezik PHP, saj se je ta glede na moje predhodno programersko znanje in obsežno dokumentacijo izkazal kot najboljša izbira. Predvsem sem se želel izogniti uporabi FBML, FQL, FBJS programskih jezikov pri razvoju aplikacije, ki vključuje Facebook. Za dinamično poživitev spletne strani sem uporabil skriptni programski jezik JavaScript oziroma na njem temelječe ogrodje jQuery. Podatkovne baze nisem uporabil, saj se podatki ob vsaki osvežitvi spletne strani ažurirajo iz posameznega socialnega omrežja.

## 5.1 Diagram primerov uporabe

Za potrebe zajemanja in predstavitve uporabniških zahtev sem izdelal diagram primerov uporabe, ki služi kot osnova pri definiranju zahtev, določanju funkcionalnosti objektov in interakciji med objekti in uporabniki. Za potrebe aplikacije sem definirala uporabnika, ki predstavlja akterja v diagramu primerov uporabe. Predpogoj za uporabo funkcij integracije podatkov je prijava v posamezno socialno omrežje.



Slika 4: Diagram primerov uporabe.

## 5.2 Registracija aplikacije

Preden sem lahko začel z izdelavo spletne aplikacije, sem se moral predhodno registrirati kot uporabnik na posamezno socialno omrežje in nato registrirati še aplikacijo. S tem sem pridobil potrošniški ključ in potrošniško skrivnost, ki ju bom v nadaljevanju potreboval pri postopku prijave z uporabniškim imenom in geslom, imenovanim avtentikacija. Poleg osnovnih podatkov sem moral pri vseh treh registracijah izpolniti še dve bistveni vnosni polji:

- URL naslov spletne strani, s katere bom dostopal do vmesnikov API;
- URL naslov, na katerega naj bo uporabnik preusmerjen po uspešni avtentikaciji.

### *Facebook*

Registracijo aplikacije sem opravil na spletni strani Facebook platforme, ki se nahaja na naslovu <http://www.developers.facebook.com/>. Poleg izpolnjevanja nekaterih tipičnih vnosnih polj sem moral vnesti še svojo obstoječo telefonsko številko, na katero sem nato prejel potrditveno kodo, s katero sem lahko nadaljeval in zaključil postopek registriranja aplikacije. Poleg vnosa telefonske številke Facebook ponuja možnost registriranja aplikacije še s številko kreditne kartice.

Po uspešni registraciji aplikacije je na voljo uporabniška podstran za pregledovanje in spreminjanje nastavitvev aplikacije, statistiko uporabe itd. Spletna stran Facebook platforme poleg upravljanja z aplikacijo ponuja obsežno dokumentacijo, blog razvijalcev, status delovanja platforme in podporo uporabnikom. Za posebej uporabnega se je izkazal graf API raziskovalec (*ang. graph API Explorer*), kjer je s pomočjo vnosne vrstice možna hitra in enostavna uporaba graf API vmesnika. Prav s tem vmesnikom sem kasneje pridobival uporabniške podatke.

### *Twitter*

Registracija aplikacije poteka na Twitter platformi, ki se nahaja na spletnem naslovu <https://dev.twitter.com>. Po uspešni registraciji aplikacije je na voljo uporabniška podstran za upravljanje z nastavitvami aplikacije, ki vsebuje le najbolj osnovna vnosna polja, kar mi je bilo posebej všeč. Hitro sem ugotovil, da Facebook ponuja obsežno zbirko nastavitvev, ki jih povprečen razvijalec nikoli ne potrebuje.

Spletna stran Twitter platforme poleg upravljanja z aplikacijo ponuja še dokumentacijo, blog razvijalcev, podroben status delovanja platforme za vsak vmesnik API posebej in podporo uporabnikom. Zame najpomembnejša je bila vsekakor konzola za razvijalce, kjer sem lahko hitro in enostavno izvajal klice API.

### *LinkedIn*

Registracijo aplikacije sem opravil na LinkedIn platformi na spletnem naslovu <https://developer.linkedin.com/>. Po uspešni registraciji je na voljo uporabniška podstran za upravljanje z nastavitvami aplikacije, ki ponuja več informacij kot Twitter, a še vedno mnogo manj kot Facebook. Vse skupaj je združeno na eni daljši podstrani, kar se mi s strani

LinkedIna ne zdi posebej posrečeno. Tako lahko na eni strani najdemo bistvene podatke o aplikaciji, kot so ime in naslov URL, zraven so še podatki o namenu uporabe in programska orodja, ki jih bomo uporabljali pri izdelavi in so za samo aplikacijo popolnoma nepomembni.

Spletna stran LinkedIn platforme poleg upravljanja z aplikacijo ponuja še podrobno dokumentacijo, blog razvijalcev, forum in zame najpomembnejšo konzolo za razvijalce za izvajanje klicev API. Od vseh treh platform sem prav za LinkedIn potreboval največ časa, da sem se privadil nanjo.

#### *Izvajanje klicev*

Prijava v posamezno socialno omrežje in izvajanje klicev API je izvedeno s pomočjo PHP knjižnice *libcurl*, ki podpira protokole http, https, https certifikate, http metode in avtentikacijo s pomočjo uporabniškega imena in gesla. Naštel sem samo bistvene oziroma tiste, ki sem jih kasneje uporabil v mojem primeru.

### 5.3 Prijava v posamezno socialno omrežje

Za prijavo v posamezno socialno omrežje sem uporabil OAuth avtentikacijski protokol, ki ga uporabljajo vsa tri omrežja. Postopek prijave je v vseh treh primerih isti. Uporabnik izbere povezavo za prijavo, od koder je preusmerjen na spletno stran socialnega omrežja. Vpiše prijavne podatke, potrdi dovoljenje za dostop do uporabniških podatkov in je na koncu preusmerjen nazaj na prvotno spletno stran. Ob prijavi se posredujejo tudi informacije o sami aplikaciji. Sestavil sem konfiguracijsko datoteko, v kateri so shranjeni potrošniški ključji, potrošniške skrivnosti in lokacija na strežniku, ki vsebuje PHP knjižnice.

```
//facebook configuration
$config['fb_api'] = '189052694478407';
$config['fb_secret'] = 'f44863f664126e0b0411c8ecd7c15701';
$config['facebook_library_path'] = 'facebook/facebook.php';
```

#### *Facebook*

Za prikaz zelenih podatkov mora aplikacija pridobiti nekatera dovoljenja od uporabnika za dostop do teh podatkov. Teh dovoljenj je kar nekaj in se jih poda preko parametra *scope* pri zahtevi za avtentikacijo. Dostop se potrdi pri prijavi v socialno omrežje, kot je prikazano na sliki 5. Dovoljenja, ki jih potrebujemo za potrebe izdelave aplikacije, so prikazana v tabeli 1.

Ime dovoljenja	Opis dovoljenja
friends_about_me	Dostop do uporabniškega dela profila »o meni«
friends_birthday	Dostop do datuma rojstnega dneva z letom rojstva
friends_education_history	Dostop do podatkov o preteklem in morebiti sedanjem šolanju

friends_hometown	Dostop do domačega mesta uporabnika
friends_likes	Dostop do seznama strani, ki so uporabniku všeč
friends_location	Dostop do lokacije, na kateri se uporabnik nahaja oziroma trenutno živi
friends_relationships	Dostop do stana uporabnika
friends_religion_politics	Dostop do uporabnikove religijske in politične usmeritve
friends_work_history	Dostop do podatkov o preteklih in sedanjih delovnih razmerij
e-mail	Dostop do e-mail naslova

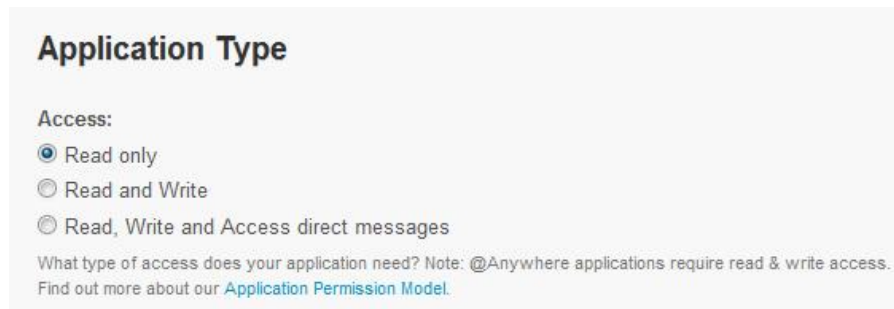
Tabela 1: Seznam dovoljenj za dostop do osebnih podatkov Facebook uporabnika.



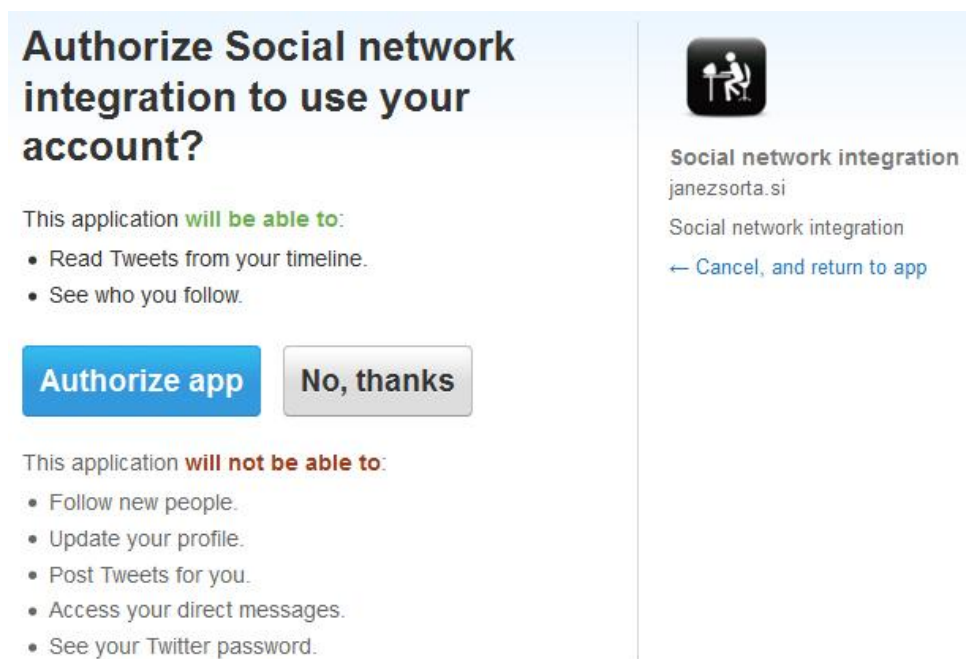
Slika 5: Potrditev dovoljenj ob Facebook prijavi.

### Twitter

Nastavitev dovoljenj pri Twitter aplikaciji poteka preko kategorije z nastavitvami na straneh platforme. Na voljo so tri vrste dovoljenj, ki so prikazana na sliki 6. Za potrebe moje aplikacije sem izbral dovoljenje samo za branje, saj bom podatke le pregledoval oziroma prebiral. Sicer si lahko uporabnik pri postopku prijave natančneje prebere, na kakšen način bo aplikacija uporabljala njegove podatke, in sicer na način, ki je prikazan na sliki 7.



Slika 6: Nastavitev dovoljenj za dostop do podatkov Twitter aplikacije.

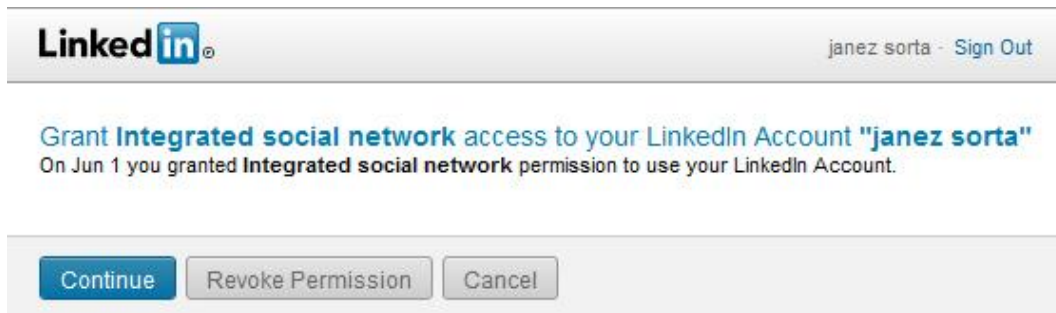


Slika 7: Potrditev dostopa aplikacije do uporabniških podatkov na socialnem omrežju Twitter.

### *LinkedIn*

LinkedIn že predvideva nastavitve pravic. V mojem primeru ima vsako polje določene pravice, ki povedo, komu je vidno in komu ne. Obstajajo tri možnosti:

- vidno vsem prijavljenim uporabnikom LinkedIn,
- vidno samo uporabniku,
- vidno uporabniku in ostalim njegovim stikom.



Slika 8: Potrditev dostopa aplikacije do uporabniških podatkov na socialnem omrežju LinkedIn.

## 5.4 Prikaz osnovnih podatkov

Ko se uporabnik uspešno prijavi v posamezna socialna omrežja, lahko spremlja stanje prijave. Prikažejo se osnovne informacije iz uporabniškega profila posameznega socialnega omrežja. Za prijavljenega uporabnika se prikaže uporabniško ime, prikazna slika in povezava do profila na socialnem omrežju.

### *Facebook*

Za prikaz osnovnih informacij o uporabniku sem uporabil graf vmesnik API, delovanje katerega temelji na dejstvu, da ima vsak objekt svoj unikatni id. Facebook poziva razvijalce, naj uporabljajo graf vmesnik API in opustijo uporabo REST vmesnika API. Trenutno je slednji še na voljo za uporabo, saj ponuja nekatere funkcionalnosti, ki jih še niso uspeli prenesti na graf vmesnik API. Omenjeni vmesnik sem v nadaljevanju uporabil pri vseh pridobivanjih podatkov s Facebooka. Klici se izvajajo na podlagi GET zahtev http.

```
https://graph.facebook.com/me
```

Zahteva vrne nekatere osnovne podatke o prijavljenem uporabniku, kar določa »me« na koncu zahteve. Namesto tega je lahko vnesen tudi id uporabnika. Podatki so v JSON formatu. Iz odgovora sem uporabil attribute:

- *id*: id uporabnika,
- *name*: ime in priimek uporabnika,
- *link*: povezava do uporabniškega profila na Facebooku.

Kot sem že omenil, sem od tu dobil uporabniško ime in povezavo do profila. Ob tem sem izkoristil še ponujeni id uporabnika, na podlagi katerega sem dobil še prikazno sliko. Zahtevo sem priredil tako, da sem jo uporabil znotraj HTML značke `img`, namenjene prikazu slik.

```

```

### Twitter

Pri izvajanju klicev API na Twitter socialnem omrežju sem uporabil za to pripravljene REST vmesnike API, ki so razdeljeni v več skupin. Klic se izvrši s pomočjo http zahteve GET, odgovori pa so v XML formatu. Struktura je enolično določena.

```
http://api.twitter.com/verzija/kategorija_vmesnika_API/vmesnik_api.format_odgovora?parametri
```

Za prikaz osnovnih informacij o uporabniku sem uporabil `verify_credentials` vmesnik API, ki vrne vse podatke o prijavljenem uporabniku. Sicer bi si v mojem primeru želel rešitev, s katero bi dobil le najbolj osnovne podatke o uporabniku, vendar je Twitter zaenkrat ne ponuja.

```
http://api.twitter.com/1/account/verify_credentials.xml
```

Poleg tega, da sem dobil osnovne podatke o uporabniku, sem z uporabljenim vmesnikom dodatno preverjal, če je bila prijava z avtentikacijo uspešna. Kot odgovor sem tako dobil še statusno kodo 200, v kolikor ni prišlo do napak, oziroma 401 s pojasnilom napake.

### LinkedIn

Za izvajanje klicev API sem v naslednjem primeru in tako tudi v nadaljevanju uporabljal REST API vmesnike. LinkedIn sicer ponuja še JavaScript vmesnike. Sintaksa je enotno določena in se med vmesniki bistveno ne razlikuje.

```
http://api.linkedin.com/v1/vmesnik_api/parametri
```

Podobno kot pri Facebooku pomeni »me« pri klicanju vmesnikov, tu enako velja za znak »~«, torej da gre za trenutno prijavljenega uporabnika. Izvrševanje klicev API temelji na GET zahtevah http, odgovori so v XML formatu.

```
http://api.linkedin.com/v1/people/~
```

Z uporabljenim vmesnikom sem dobil vse podatke, ki sem jih potreboval. Kot že omenjeno, znak ~ na koncu zahteve pomeni, da gre za trenutno prijavljenega uporabnika, `people` pa predstavlja vmesnik API za pridobivanje podatkov o posameznikih. Sicer se omenjeni vmesnik API lahko uporablja na dva načina:

- standardni vrne celoten profil uporabnika. Prikaz posebnih vsebin je odvisen od nastavitve zasebnosti lastnika profila, razmerje med lastnikom profila in tistim, ki do profila dostopa;
- javni vrne polja, ki jih lastnik profila označi kot javne.

Po uspešni prijavi v vsa tri socialna omrežja je uporabnik pripravljen na uporabo funkcij, ki integrirajo osebne podatke njegovih prijateljev. Stanje prijave lahko spremlja preko zaslonske maske, prikazane na sliki 9.



Slika 9: Uspešna prijava na vsa tri socialna omrežja z uporabo aplikacije.

## 5.5 Prikaz seznama prijateljev po posameznem socialnem omrežju

Ko je uporabnik uspešno opravil vsaj eno prijavo, si lahko ogleda seznam prijateljev za posamezno socialno omrežje. Kar me je tu zanimalo, so bili imena, priimki in id-ji uporabnikovih prijateljev. Ime in priimek sem potreboval zgolj za prikaz, id pa sem podal kot parameter v URL naslovu, ki predstavlja povezavo do podrobnosti o uporabniku.

### Facebook

```
https://graph.facebook.com /me/friends
```

```
{
  "data": [
    {
      "name": "Matej Lavrenčič",
      "id": "523279419"
    },
    {
      "name": "Matej Vidmar",
      "id": "524189361"
    },
    {
      "name": "Polona Strancar",
      "id": "530653097"
    },
    ...
  ]
}
```

S klicem zgornjega graf vmesnika API sem dobil imena, priimke in id-je uporabnikovih prijateljev. Friends na koncu zahteve pomeni, da iščem prijatelje navedenega uporabnika.

## Twitter

```
https://api.twitter.com/1/friends/ids.xml
```

```
<ids>
  <id>14538601</id>
  <id>14131261</id>
  <id>277177552</id>
  ...
</ids>
```

Vmesnik API vrne seznam id-jev uporabnikovih prijateljev, kar v mojem primeru ni bilo dovolj. Potreboval sem še imena in priimke, ki sem jih dobil z naslednjim klicem *show* vmesnika API.

```
https://api.twitter.com/1/users/show.xml?user_id=id_uporabnika
```

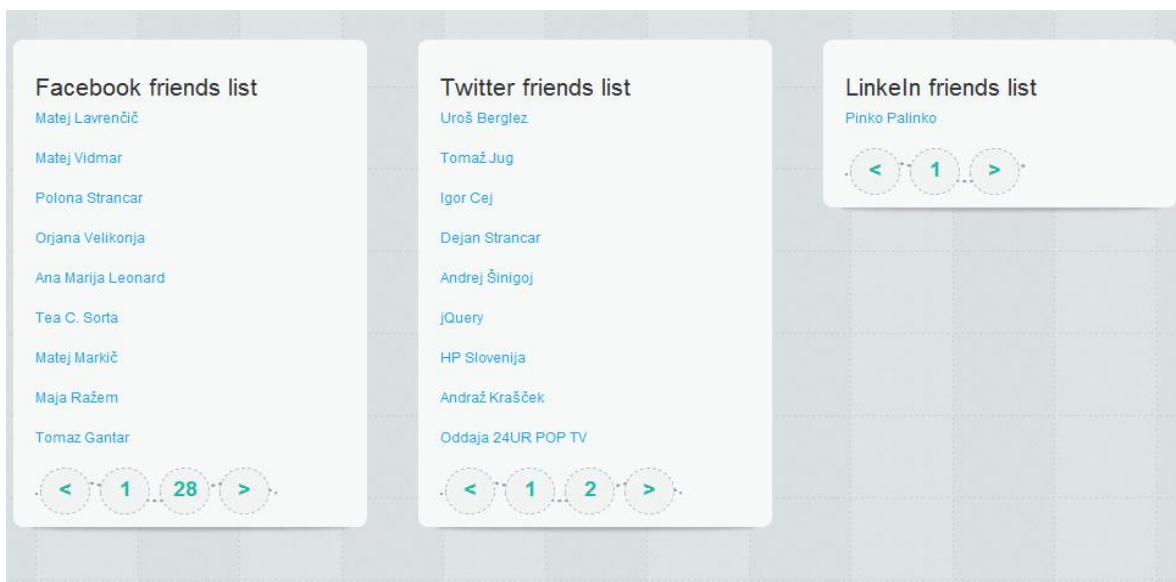
Dobil sem kar vse podatke o uporabniku, z določenim id-jem, kar mi je predstavljalo le dodatno delo in upočasnjevalo aplikacijo. Iz XML odgovora sem nato uporabil le ime in priimek uporabnikovega prijatelja.

## LinkedIn

```
http://api.linkedin.com/v1/people/~connections:(id,first-name,last-name)
```

```
<connections total="1">
  <person>
    <id>acwx36eFUa</id>
    <first-name>Pinko</first-name>
    <last-name>Palinko</last-name>
  </person>
</connections>
```

Z uporabo *connections* vmesnika API povezave in s podanimi ustreznimi parametri sem prišel do podatkov, ki sem jih potreboval pri sestavljanju seznama LinkedIn uporabnikov. Pri uporabi omenjenega vmesnika in prav tako kateregakoli drugega med osebnimi podatki ni mogoče dostopati do e-mail naslova. Prav tako velja pri dostopanju do podatkov o prijateljih pravilo prve globine, kar pomeni, da lahko uporabnik dostopa le do podatkov svojih prijateljev, do podatkov prijateljevih prijateljev pa več ne.



Slika 10: Seznam prijateljev po posameznem omrežju.

## 5.6 Prikaz osebnih podatkov o posameznem prijatelju

Za posamezno socialno omrežje sem nato prikazal vse osebne podatke o uporabniku, ki jih določeno socialno omrežje ponuja, ločeno za vsako posebej.

### Facebook

```
https://graph.facebook.com/Id_uporabnika?
fields=name,gender,locale,languages,bio,birthday,education,email,hometown,location,political,relationship_status,religion,work
```

Z uporabo parametra fields sem dobil le podatke, ki so me zanimali in imajo naslednji pomen:

- name: polno uporabniško ime
- gender: spol
- locale: kratica države
- languages: jeziki, ki jih uporabnik govori
- bio: biografija
- birthday: datum rojstva
- education: podatki o izobrazbi
- email: e-mail naslov
- hometown: domače mesto
- location: trenutni naslov
- political: politična usmeritev
- relationship\_status: stan
- religion: religija
- work: podatki o delu

## Twitter

```
https://api.twitter.com/1/users/show.xml?user_id=id_tw
```

Preko parametra sem podal id uporabnika in dobil kopico podatkov, od katerih sem uporabil le polno uporabniško ime, lokacijo in kratek opis uporabnika. Sicer Twitter ne ponuja posebej veliko osebnih podatkov, omogoča pa oblikovanje uporabniškega profila. Z uporabo prej omenjenega vmesnika API sem prišel do barv v profilu uporabnika in s pomočjo CSS ukazov trenutni prikaz še bolj približal tistemu na Twitterju.

```
<style type="text/css">
table.user_data tr td {
    border: 1px solid #<?=$tw_data->profile_sidebar_border_color?>;
    color: #<?=$tw_data->profile_text_color?>;
    background-color: #<?=$tw_data->profile_sidebar_fill_color?>;
}
a.back {
    color: #<?=$tw_data->profile_link_color?>
}
</style>
```

## LinkedIn

```
http://api.linkedin.com/v1/people/id=id_li:(parametri)
```

LinkedIn ponuja zajetno število osebnih podatkov. Uporabniški profil naj bi predstavljal kar neke vrste življenjepis. Med parametri klika *people* vmesnika API sem navedel polja, ki me zanimajo:

- id: id uporabnika
- first-name: ime
- last-name: priimek
- headline: navadno trenutno delovno mesto v podjetju
- location:(name): naslov prebivališča
- industry: industrijska dejavnost, v kateri uporabnik dela
- summary: daljši opis uporabnikove profesionalne kariere
- specialties: krajši opis, kjer uporabnik našteje njegove posebne lastnosti
- positions: podatki o sedanjih in preteklih zaposlitvah
- associations: kratek opis uporabnikovih združenj, v katerih sodeluje
- publications: zbirka publikacij, ki so last uporabnika oziroma je sodeloval pri nastanku le-teh
- patents: zbirka patentov, ki so v lasti uporabnika
- languages: jeziki, ki jih uporabnik govori
- skills: zbirka posebnih znanj, veščin uporabnika
- certifications: zbirka uporabnikovih certifikatov
- educations: podatki o šolanju
- phone-numbers: seznam telefonskih števil

- date-of-birth: datum rojstva
- main-address: naslov trenutnega prebivališča
- picture-url: url do slike profila

## 5.7 Združevanje skupnih prijateljev z več socialnih omrežij

Na tem mestu želim prikazati seznam prijateljev iz vseh treh socialnih omrežij. Za pridobitev posameznega seznama sem uporabil vmesnike API in metode iz poglavja Prikaz seznama prijateljev po posameznem socialnem omrežju. Nato sem sestavil tabelo, kjer bom hranil uporabniška imena in id-je uporabnikov po posameznem socialnem omrežju.

```
$users = Array( 'name'=>array(),
               'id_fac'=>array(),
               'id_tw'=>array(),
               'id_li'=>array()
             );
```

Sprehodil sem se čez vse tri sezname in v tabelo *users* vstavljaj uporabniška imena in id-je po posameznem socialnem omrežju. V kolikor sem ugotovil, da gre za skupnega prijatelja, sem moral posebej paziti na pozicijo, na katero vstavljaj id.



Slika 11: Združeni prijatelji iz vseh treh socialnih omrežij v en seznam.

## 5.8 Združevanje vseh razpoložljivih osebnih podatkov posameznega prijatelja z več socialnih omrežij

V kolikor nek atribut ne obstaja na več socialnih omrežjih, sem se držal pravila, da izpišem tistega, kjer podatek obstaja. Kaj kmalu sem naletel na problem, kateri podatek izpisati, v kolikor obstaja na več socialnih omrežjih, oziroma komu bolj zaupati. V nasprotnem primeru sem se združevanja in določanja prioritet lotil na v nadaljevanju opisane načine. Za dostop do podatkov sem uporabil vmesnike API, opisane v poglavju Prikaz osebnih podatkov o posameznem prijatelju.

Atribut	Dostopno na socialnih omrežjih	Način združevanja in izpisa
Ime in priimek	Facebook Twitter LinkedIn	Brez omenjenega parametra uporabniški profil ne more obstajati, zato posebna izbira ponudnika ni potrebna.
Spol	Facebook	Dostopno le iz enega omrežja.
Oznaka države	Facebook	Dostopno le iz enega omrežja.
Jeziki	Facebook LinkedIn	Izpis poteka iz navedenih socialnih omrežij, pri čemer je treba paziti, da ne prihaja do podvajanj.
Naslov	Facebook LinkedIn	Pri izpisu ima prednost LinkedIn.
Biografija	Facebook Twitter LinkedIn	Biografija v več različicah obstaja na vseh treh socialnih omrežjih. Na LinkedInu sta to atributa specialties (krajši opis z uporabnikovimi posebnimi lastnostmi) in summary (daljši opis uporabnikove profesionalne kariere), na Twitterju description (kratek opis uporabnika) in na Facebooku bio (biografija). Vse attribute sem združil v celoto, tako da sem izpisal enega pod drugim.
Rojstni datum	Facebook LinkedIn	Prioriteto pri izpisu ima LinkedIn.
Izobrazba	Facebook LinkedIn	Na LinkedInu so shranjeni podatki o imenu ustanove, področju izobraževanja, začetku

		in koncu izobraževanja, pridobljeni stopnji izobrazbe, dejavnostih, ki jih je oseba opravljala v času študija, in o opombah. Poleg posamezne ustanove sem s Facebooka dodal razrede in sošolce po posameznih letih izobraževanja.
Domače mesto	Facebook	Dostopno le iz enega omrežja.
Trenutni naslov	Facebook Twitter LinkedIn	Po prioriteti si sledijo LinkedIn, Twitter, Facebook.
Zakonski stan	Facebook	Dostopno le iz enega omrežja.
Delovna razmerja	Facebook LinkedIn	Večina podatkov prihaja s strani LinkedIn. Od tu dobim ime podjetja, začetni in morebiten končen datum delovanja v podjetju in nekatere informacije o podjetju. Poleg naštetega sem s Facebooka dodal podatke o morebitnih projektih pri posameznem podjetju.
Industrija	LinkedIn	Dostopno le iz enega omrežja.
Združenja	LinkedIn	Dostopno le iz enega omrežja.
Odlikovanja	LinkedIn	Dostopno le iz enega omrežja.
Publikacije	LinkedIn	Dostopno le iz enega omrežja.
Patenti	LinkedIn	Dostopno le iz enega omrežja.
Posebna znanja	LinkedIn	Dostopno le iz enega omrežja.
Certifikati	LinkedIn	Dostopno le iz enega omrežja.
Telefonske številke	Facebook LinkedIn	V kolikor se telefonske številke ne podvajajo, se izpišejo v seznamu.

Tabela 2: Integracija podatkov po posameznih poljih.

Integrated user data from Facebook and Twitter		
Name		
Gender	male	
Locale	sl	
Languages	English Hrvatski Deutsch Italiano	
Description	Real-time enterprise ; Social media ; Mobile & web user experience ; Web design ; Content management ; Entrepreneurship ; Franchising ; House music & Wine :)	
Birthday	12/11	
Education	School name	srednja ekonomska in trgovska šola
	Start date	1995
	Type	High School
	School name	Gea College
	Start date	1999
	Concentration	Podjetništvo
	Type	College
	Hometown	Solkan
Location	Solkan	
Work	Title	Editor d.o.o.

Back

Slika 12: Integrirani podatki iz socialnega omrežja Facebook in Twitter.

Pri integraciji podatkov se je večkrat zgodil primer, da je obstajal različen podatek na več socialnih omrežjih. Na prvem socialnem omrežju je pisalo, da je oseba iz Nove Gorice, na drugem pa iz Ajdovščine. Pojavil se je problem zaupanja v pridobljene podatke. Analiziral sem nekatere osnovne podatke, ki jih proizvedejo uporabniki in ostale osnovne funkcionalnosti na vseh treh obravnavanih socialnih omrežjih, da bi tako dobil najbolj zaupanja vredno socialno omrežje. Tudi sam že več časa uporabljam vsa tri obravnavana socialna omrežja, na katerih imam več stikov, tako da sem na podlagi le-teh določil stopnjo zaupanja. Podrobneje sem pregledal profile več naključno izbranih in primerjal podatke med seboj. Odločal sem se na podlagi naslednjih kategorij oziroma vprašanj, ki sem si jih zastavil:

- Ali imajo vsi osebni podatki, ki jih kot take smatra socialno omrežje, vsebino, ki bi lahko bila uporabna za aplikacijo. Za primer naj navedem sicer dobro urejen zavihek posameznega uporabnika »podatki« na socialnem omrežju Facebook, ki med drugim vsebuje dejavnosti in interese. Ti so večinoma neuporabne povezave do različnih oglasnih podstrani in iger oziroma povezave, ki se v uporabnikov profil dodajo s klikom na gumb »všeč mi je«. Kot nasprotje Facebooku se profili uporabnikov na LinkedInu skušajo približati življenjepisu, tako da so takšne neprimerne povezave znotraj osebnih podatkov prava redkost, kar povečuje zaupanje v pravilnost podatkov v profilu.
- Posebno pozornost pri pregledovanju osebnih podatkov sem dal prikazni sliki. Kot najbolj primerna bi bila slika, na kateri je uporabnik, oziroma slika, podobna tistim v osebnih dokumentih.

- Postopek vzpostavljanja novih stikov. Kot najpreprostejšega se je v tej kategoriji izkazal Twitter. Uporabnik lahko sledi oziroma spremlja drugega uporabnika zgolj s klikom na gumb sledi (*angl. follow*). Nasprotje temu je LinkedIn, kjer je postopek vzpostavitve stika strožje oziroma bolj dosledno določen. Kot osnova je določitev razmerja s stikom, s katerim želimo vzpostaviti povezavo (prijatelj, sodelavec, delodajalec itd.). Tak način vzpostavljanja stikov oziroma sprejetje novega stika v skupino temelji na večjem zaupanju med uporabniki.
- Pomen in vsebina sporočil, ki jih uporabniki delijo s svojimi stiki. Pogovori na LinkedInu večkrat potekajo na določeno strokovno temo. Uporabniki si s primernimi odgovori lahko pridobijo naklonjenost morebitnih delodajalcev, torej se zavedajo pomembnosti svojih objavljenih sporočil in s tem se poveča stopnja zaupanja v vsebino objavljenih sporočil.
- Zaupanje socialnih omrežij med seboj se kaže na vsebovanosti konceptov drugih socialnih omrežij. LinkedIn tako vsebuje funkcijo oziroma vmesnik API za povezavo s Twitter računom. Twitter uporabnikom med drugim omogoča objavo sporočil na Facebooku.

V tabeli 3 sem predstavil rezultate stopnje zaupanja podatkov po posameznem socialnem omrežju. Posamezen kriterij sem ocenil z ocenami od ena do tri. Kot najbolj vrednega zaupanja se je izkazal LinkedIn.

<b>Kriterij</b>	<b>Facebook</b>	<b>Twitter</b>	<b>LinkedIn</b>
Osebni podatki	2	2	3
Nastavitve zasebnosti	2	1	3
Prikazna slika	1	1	2
Ostale multimedijske vsebine	2	2	2
Vzpostavitev novega stika	2	1	3
Vsebina objav uporabnika	1	2	3
Vsebina odgovorov na objave	1	2	3
<b>Skupaj</b>	11	11	19

Tabela 3: Ocena zaupanja po posameznem socialnem omrežju.

## 5.9 Iskalnik prijateljev

Stike sem iskal iz seznama obstoječih prijateljev. Preizkusil sem tudi možnost iskanja po celotnih bazah uporabnikov, kar se je izkazalo za neuporabno. Facebook večinoma ponuja le osnovne podatke o uporabniku, medtem ko LinkedIn v večini primerov vrne le ime in priimek. Dobljene uporabnike sem združil po načinu iz poglavja Združevanje skupnih prijateljev z več socialnih omrežij in jim dodal povezave za izpis integriranih podatkov po

načinu, opisanem v poglavju Združevanje vseh razpoložljivih osebnih podatkov posameznega prijatelja z več socialnih omrežij.

### *Facebook*

```
https://graph.facebook.com/search?q=iskani_niz&type=user
```

Graf vmesnik API omogoča tudi iskanje. Preko parametra *q* je podan iskani niz oziroma zahteva, preko parametra *type* pa tip rezultata, ki ga iščemo.

### *Twitter*

```
https://api.twitter.com/1/users/search.xml?q=iskani_niz
```

*Search* vmesnik API za iskanje v ozadju skriva isto programsko kodo za iskanje kot tisti na samem socialnem omrežju. Število rezultatov je omejeno na 1000. Parameter *users* sporoča, da gre za iskanje po uporabnikih, *q* pa vsebuje iskani niz.

### *LinkedIn*

```
http://api.linkedin.com/v1/people-search?iskani_niz
```

Iskanje uporabnikov po LinkedIn socialnem omrežju poteka z uporabo *people-search* vmesnika API, kateremu se kot parameter poda iskani niz.

## 6 SKLEPNE UGOTOVITVE

V diplomski nalogi sem najprej predstavil nekatera socialna omrežja. Razdelil sem jih po njihovem osnovnem namenu oziroma vrsti. Po pregledu številnih me je najprej presenetilo veliko število obstoječih socialnih omrežij, ki so v uporabi. Temu bi pripisal dejstvo, da obstaja več manjših, izdelanih za ozko usmerjene skupine ljudi. Z malo pretiravanja bi skoraj lahko rekel, da tako kot ima lahko danes vsakdo svojo spletno stran, ima lahko tudi socialno omrežje. Hitro sem ugotovil, da ima vsako področje svojega predstavnika, ki bistveno odstopa od ostalih. V času pisanja diplomske naloge je luč sveta ugledalo dolgo pričakovano socialno omrežje podjetja Google, in sicer Google+, ki naj bi predstavljalo nov mejnik v zgodovini socialnih omrežij. Po začetnem velikem navalu uporabnikov se je uporaba Google+ nekoliko stabilizirala. Tudi napoved, da bo Google+ prehitel oziroma nasledil Facebook, se za sedaj zdi še zelo daleč.

Nato sem se posvetil samemu problemu integracije podatkov v socialnih omrežjih. Ugotovil sem, da obstaja kar nekaj programskih rešitev za omenjen problem. Našel sem jih v obliki namiznih aplikacij, spletnih strani, spletnih brskalnikov in dodatkov za spletne brskalnike. Med vsemi ni predstavnika, ki bi izstopal, kot na primer izstopa Facebook od ostalih socialnih omrežij. Pri večini programskih rešitev gre zgolj za izpisovanje novih objav iz socialnih omrežij. Orodje, ki bi znalo še kaj več od tega, vsaj za sedaj ni na voljo.

Nato sem predstavil programsko arhitekturo REST, na kateri temelji izvajanje vmesnikov API, ki so bili osnova za izdelavo aplikacije. Arhitekturo sem primerjal s komunikacijskim protokolom SOAP, katerega osnovni namen je prav tako izmenjava sporočil v omrežjih.

Zadnji del diplomske naloge obravnava izdelavo aplikacije za integracijo osebnih podatkov iz socialnih omrežij Facebook, Twitter in LinkedIn, z uporabo programskih vmesnikov API. Uporabnik se mora najprej prijaviti v posamezna socialna omrežja, nato lahko vidi osebne podatke stikov iz prej omenjenih socialnih omrežij na enem mestu.

Prve težave sem imel z delovanjem vseh treh PHP knjižnic za prijavo in izvajanje klicev API. Problem je bil, ker so se vse knjižnice izvajale na isti lokaciji, oziroma bolje rečeno na isti spletni strani in je tako prihajalo do konfliktov pri delovanju. Prikaz osebnih podatkov iz vseh treh socialnih omrežij naenkrat je zahteval več klicev API, kar je močno upočasnjevalo aplikacijo. Posamezni klici API so časovno dokaj potratni. Pri integraciji podatkov se je problem pojavil, ko je bilo isto polje vneseno na več socialnih omrežjih. Na primer, na Facebooku je bilo zapisano, da je uporabnik iz Ljubljane, na LinkedInu pa da je iz Vrhnike. Tako sem bil postavljen pred vprašanje, kateremu socialnemu omrežju bolj zaupati. Glede na to, da velja LinkedIn za poslovno in profesionalno socialno omrežje, sem se odločil zaupati prav njemu. Pri izdelavi iskalnika sem želel izvesti iskanje po vseh uporabnikih posameznega socialnega omrežja in ne le po stikih uporabnika. Kaj kmalu sem ugotovil, da na ta način preko klicev API v večini primerov dobim le uporabniško ime, zato sem to idejo opustil. Še ena težava, na katero sem naletel med testiranjem, je bilo preseženo maksimalno število

poizvedb oziroma klicev API na določeno časovno obdobje. Tipično je število klicev API omejeno na uro oziroma na dan.

V mojem primeru gre za novo izdelano aplikacijo, tako da jo je gotovo moč še izboljšati. Predvsem menim, da bi lahko prihranil čas pri izvajanju klicev API. Glede na moje predhodno znanje sem se odločil uporabiti PHP programski jezik. V vsaj nekaterih primerih bi lahko odzivnost zmanjšal z uporabo Javascript vmesnikov API in posledično tehnologije AJAX.

**LITERATURA IN VIRI**

- [1] Bill Burke, *RESTful Java with JAX-RS*, Sebastopol, California, O'Reilly, nov. 2009, pogl. 1.
- [2] (2009) Manage Your Social Networks. Dostopno na: <http://www.entrepreneur.com/article/201136>
- [3] (2011) Spletno socialno omrežje. Dostopno na: [http://sl.wikipedia.org/wiki/Spletno\\_socialno\\_omre%C5%BEje](http://sl.wikipedia.org/wiki/Spletno_socialno_omre%C5%BEje)
- [4] (2009) Types of Social Networking Sites. Dostopno na: <http://ezinearticles.com/?Types-of-Social-Networking-Sites&id=4019476>
- [5] (2011) The New Twitter Developer Site. Dostopno na: <https://dev.twitter.com/blog/new-twitter-developer-site>
- [6] (2011) YouTube APIs and Tools. Dostopno na: [http://code.google.com/intl/sl-SI/apis/youtube/getting\\_started.html](http://code.google.com/intl/sl-SI/apis/youtube/getting_started.html)
- [7] (2011) Open Graph protocol. Dostopno na: <http://developers.facebook.com/docs/opengraph/>
- [8] (2008) Data Integration in Social Networks. Dostopno na: [http://www-users.cs.umn.edu/~phalak/Data\\_Integration\\_in\\_Social\\_Networks.pdf](http://www-users.cs.umn.edu/~phalak/Data_Integration_in_Social_Networks.pdf)
- [9] (2011) OpenSocial. Dostopno na: <http://code.google.com/intl/sl-SI/apis/opensocial/>
- [10] (2008) Mobilna identiteta – OpenID. Dostopno na: <http://www.monitor.si/clanek/mobilna-identiteta-openid/>
- [11] (2010) Professional-Strength Social Media Aggregators. Dostopno na: <http://socialmediatoday.com/index.php?q=SMC/203911>
- [12] (2010) The Best Social Aggregators. Dostopno na: <http://www.axzm.com/blog/the-best-social-aggregators>
- [13] (2010) Social mention. Dostopno na: <http://www.slideshare.net/aashmore/social-mention>
- [14] (2008) Learn REST: A Tutorial. Dostopno na: <http://rest.elkstein.org/2008/02/what-is-rest.html>

- [15] (2010) Key Principles of REST. Dostopno na:  
<http://dev.anyframejava.org/docs.en/anyframe/plugin/springrest/1.0.2/reference/html/ch03.html>
- [16] Leonard Richardson, Sam Ruby, *RESTful Web Services*, Sebastopol, California, O'Reilly, maj 2007, pogl. 3.
- [17] Anthony T. Velte, Toby J. Velte, Robert Elsenpeter, *Cloud Computing: A Practical Approach*, McGraw-Hill, 2009, pogl. 6.
- [18] Mark D. Hawker, *The Developer's Guide to Social Programming*, Crawfordsville, Indiana, Pearson Education, avg. 2010, pogl. 1, 2.
- [19] Gavin Bell, *Building Social Web Applications*, Sebastopol, California, O'Reilly, sep. 2009, pogl. 1, 10, 17
- [20] Jesse Feiler, *How to Do Everything with Web 2.0 Mashups*, McGraw-Hill, 2007, pogl. 2.