

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miha Trtnik

**RAZVOJ APLIKACIJ SPLETNIH SOCIALNIH MREŽ
ZA IZVEDBO VIRUSNEGA MARKETINGA NA
PRIMERU FACEBOOK PLATFORME**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Ljubljana, 2009



Št. naloge: 01610/2009

Datum: 15.10.2009

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

Kandidat: **MIHA TRTNIK**

Naslov: **RAZVOJ APLIKACIJ SPLETNIH SOCIALNIH MREŽ ZA IZVEDBO
VIRUSNEGA MARKETINGA NA PRIMERU FACEBOOK PLATFORME
SOCIAL NETWORK SITES APPLICATION DEVELOPMENT FOR VIRAL
MARKETING NEEDS ON FACEBOOK PLATFORM EXAMPLE**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V okviru diplomskega dela predstavite spletne socialne mreže in možnosti njihove uporabe s poudarkom na Facebooku. Proučite možnost izvedbe takoimenovanega virusnega marketinga med uporabniki Facebooka. Predstavite tehnologije potrebne pri povezovanju aplikacij na Facebook platformo in možne integracijske točke za povezavo aplikacij s končnimi uporabniki. Razvijte prototip aplikacije, ki bo omogočala izvedbo nagradnih iger in vanjo vključite elemente, ki omogočajo lažje in hitrejše virusno širjenje.

V zaključku pojasnite prednosti in pomanjkljivosti Facebook platforme, ki jih boste spoznali med izvajanjem vaše aplikacije.

Mentor:

prof. dr. Marjan Krisper



Dekan:

prof. dr. Franc Solina

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miha Trtnik

**RAZVOJ APLIKACIJ SPLETNIH SOCIALNIH MREŽ
ZA IZVEDBO VIRUSNEGA MARKETINGA NA
PRIMERU FACEBOOK PLATFORME**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Marjan Krisper

Ljubljana, 2009

--- Original izdane teme diplomskega dela ---

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Miha Trtnik, z vpisno številko 63020168, izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja prof. dr. Marjana Krisperja, ter soglašam, da je diplomsko delo v elektronski obliki dostopno v zbirki »Dela FRI«.

V Ljubljani, dne 9. december 2009

Miha Trtnik

Zahvala

Za strokovno pomoč in nasvete pri pisanju diplomske naloge se zahvaljujem prof. Marjanu Krisperju in asistentu Dejanu Lavbiču.

Zahvaljujem se moji družini, ki me je tekom moje študijske poti podpirala.

Kazalo

Povzetek	1
Abstract.....	2
1. Uvod.....	3
2. Socialni graf	3
3. Virusni marketing.....	4
3.1 Opredelitev pojma virusni marketing	5
3.2 Izvedba virusnega marketinga	5
3.3 Elementi virusnega marketinga s primeri	6
3.3.1 Ponudba brezplačnih izdelkov ali storitev.....	6
3.3.2 Omogoča enostaven prenos na druge	6
3.3.3 Lahka razširitev	6
3.3.4 Izrablja pogosta obnašanja in motivacijo	7
3.3.5 Uporablja kombinirane vire komunikacije.....	7
3.3.6 Uporablja tuje vire	7
3.4 Prednosti virusnega marketinga.....	8
3.5 Slabosti virusnega marketinga	8
4. Definicija spletne socialne mreže.....	8
4.1.1 Identiteta	9
4.1.2 Prisotnost	9
4.1.3 Odnosi in razmerja	9
4.1.4 Pogovor.....	9
4.1.5 Skupnost	9
4.1.6 Ugled	9
4.1.7 Souporaba	10
5. Kratek pregled zgodovine spletnih socialnih mrež	11
5.1 Oglasne deske	11
5.2 IRC, ICQ in pogovorne aplikacije	11
5.3 Zgodnje spletne socialne mreže.....	11
5.3.1 SixDegrees.com.....	11
5.3.2 LiveJurnal	12
5.3.3 Ryze.com	12
5.3.4 Friendster	12

5.4	Sodobne spletne socialne mreže.....	13
5.4.1	MySpace.....	13
5.5	Facebook	14
6.	Razvojne možnosti	17
6.1	Facebook platforma.....	17
6.2	OpenSocial	19
7.	Facebook Platform.....	20
7.1	Princip delovanja klasične spletne aplikacije.....	20
7.2	Princip delovanja Facebook aplikacije.....	20
7.2.1	Klasična Facebook aplikacija z uporabo FBML.....	20
7.2.2	Facebook aplikacija z uporabo IFrame	21
7.3	Facebook API.....	21
7.3.1	REST – Representational State Transfer	22
7.3.2	Odgovori vmesnika API in lokalizacija.....	22
7.3.3	Primerjava FBML IFrame.....	22
7.4	JavaScript vmesnik API	22
7.5	FBML	23
7.6	XFBML	23
7.7	FBJS	24
7.8	FQL	24
8.	Anatomija Facebook aplikacije	25
8.1.1	Imenik aplikacij (Application directory).....	25
8.1.2	O aplikaciji (About page).....	25
8.1.3	Platno aplikacije (Canvas page).....	25
8.1.4	Profil	25
8.1.5	Nastavitve zasebnosti.....	27
8.1.6	Krmilnik novic (<i>Feed</i>)	27
8.1.7	Opozorila (<i>Alerts</i>)	27
8.1.8	Vabila (<i>Request</i>).....	27
8.1.9	Interakcija z uporabniki (Interacting with users).....	28
9.	Razvoj prototipa aplikacije.....	29
9.1	Ozadje.....	29
9.2	Opis aplikacije.....	30

9.3	Problemska domena	30
9.4	Primeri uporabe - PU	30
9.4.1	PU Dodaj nagradno igro	31
9.4.2	PU Preglej nagradne igre	32
9.4.3	PU Žrebaj	32
9.4.4	PU Sodeluj	33
9.4.5	PU Priporoči prijatelju	33
9.5	Diagrami zaporedja	33
9.5.1	Dodaj nagradno igro	34
9.5.2	Preglej nagradne igre	35
9.5.3	Žrebaj	36
9.5.4	Sodeluj	37
9.5.5	Priporoči	37
9.6	Entitetni diagram podatkovne baze	38
10.	Arhitektura	38
10.1	MVC	38
10.2	Trinivojska arhitektura	39
10.3	Prilagoditev konceptov potrebam	41
10.3.1	Facebook API	41
10.3.2	Arhitektura	41
10.3.3	Zajem podatkov	42
10.4	Izbira programskega jezika	42
11.	Razvoj Facebook aplikacije v praksi	43
11.1	Nastavitve	44
11.1.1	Osnova (<i>Basic</i>)	44
11.1.2	Avtorizacija namestitve (<i>Authentication</i>)	45
11.1.3	Profili (<i>Profiles</i>)	45
11.1.4	Platno (<i>Canvas</i>)	46
11.1.5	Povezava (<i>Connect</i>)	47
11.1.6	Gradniki (<i>Widgets</i>)	48
11.1.7	Napredno (<i>Advanced</i>)	48
11.2	Inicializacija Facebook vmesnika	49
11.2.1	Nastavitve strežnika	49

11.3	Uradne knjižnice za programski jezik PHP	49
11.3.1	Inicializacija	50
11.4	Preverjanje uporabnika	50
11.4.1	Preverjanje prijavljenosti v sistem	50
11.4.2	Preverjanje, ali je uporabnik administrator	51
11.5	Dovoljenja aplikacije	51
11.5.1	Primer zapisa kode za preverjanje dovoljenj	51
11.5.2	Zaslonska maska	52
11.6	Izvedba primera uporabe Dodaj nagradno igro	52
11.6.1	Enovrstična polja s tekstom	53
11.6.2	Izbira iz nabora vrednosti	53
11.6.3	Vnos besedila	53
11.6.4	Vnos poljubnih elementov forme	53
11.6.5	Izbira datuma	53
11.6.6	Primer zapisa oznak za vnosno formo	55
11.6.7	Zaslonska maska	56
11.7	Izvedba primera uporabe – Preglej nagradne igre	57
11.7.1	Zaslonska maska	57
11.8	Izvedba primera uporabe – Sodeluj	57
11.8.1	Primer kode za objavo obvestila v založniku	58
11.8.2	Zaslonska maska	59
11.9	Izvedba primera uporabe Priporoči prijatelju	59
11.9.1	Primer zapisa oznak za formo priporoči prijatelju	60
11.9.2	Zaslonska maska	61
11.10	Izvedba primera uporabe Žrebaj	62
11.10.1	Primer zapisa kode za pošiljanje elektronske pošte	62
11.10.2	Zaslonska maska	63
12.	Zaključek	63
13.	SEZNAM UPORABLJENIH VIROV	65

Seznam slik

Slika 1 - Skica prikazuje socialni graf osebe A, ki je postavljena v središče. Osebe, ki so neposredno povezane, so zarisane malenkost manjše, osebe, oddaljene dva koraka, še manjše itd. [2]	4
Slika 2 - Hitro širjenje in prenos ideje na prijatelje.....	5
Slika 3 - Skica prikazuje gradnike spletnih socialnih mrež, spodnje skice pa prikazujejo poudarke posameznih gradnikov, ki jih imajo konkretni primeri spletnih socialnih mrež. Kot je razvidno, ima Facebook med primeri najbolj uravnoteženo zgradbo. [11]	10
Slika 4 - Primer oglasne deske	11
Slika 5 - Prikaz števila aktivnih uporabnikov Facebooka. Za aktivne uporabnike veljajo tisti registrirani uporabniki, ki so se vsaj enkrat v zadnjih tridesetih dneh prijavili na stran.	14
Slika 6 - Primerjava dnevnik prikazov strani friendster.com, facebook.com in myspace.com. Na Y osi so označeni odstotki prikazov strani glede na število prikazov strani celotnega svetovnega spleta, ki ga obravnava spletna storitev Alexa. Na X osi je izbran največji časovni razpon, ki ga je na Alexi še moč izbrati. Vidimo, da je šele na začetku 2009 Facebooku uspel usodni preboj, s katerim je iz prestola najbolj obiskane spletne skupnosti vrgel MySpace.	15
Slika 7 - Časovni trak nastajanja spletnih socialnih mrež, označene so le najpomembnejše – vseh je na tisoče. Točen čas nastanka je zelo težko določiti, saj vsaka stran ob navajanju podatkov upošteva subjektivne kriterije, kaj začetek sploh pomeni. Nekatere za začetek štejejo datum prve registracije domene, druge datum prijave prvega uporabnika na stran, tretje so obstajale že dolgo časa, preden so izpolnjevale pogoje za socialno mrežo. Podatki so pridobljeni iz vira označenega pod številko [10], spletne storitve Alexa.com, datume slovenskih ponudnikov pa sem izvedel s povpraševanjem preko elektronske pošte.....	16
Slika 8 - Standardni pristop k izvedbi spletnih aplikacij je postavitev aplikacije skupaj s podatkovno bazo na enoten strežnik, do katerega lahko uporabniki dostopajo preko svetovnega spleta s pomočjo unikatnega URL-ja. Facebookov model je malenkostno drugačen. Uporabnik do aplikacije dostopa preko Facebooka, čeprav je aplikacija dejansko nameščena na svojem strežniku. Facebook nastopa kot dvosmerni proksi med brskalnikom uporabnika in aplikacijo, pri čemer prestreza in spreminja podatkovni tok v realnem času (na primer spreminja vsa imena JS spremenljivk). S tem Facebook ohranja nadzor nad izvajanjem.....	18
Slika 9 - Skica trinivojske zasnove aplikacije. Predstavitveni nivoji so ločeni, logika v ozadju ostaja enaka. [24]	19
Slika 10 - Princip delovanja klasične spletne aplikacije.....	20
Slika 11 - Princip delovanja Facebook aplikacije z uporabo označevalnega jezika FBML in klici funkcij vmesnika API.....	21
Slika 12 - Princip delovanja Facebook aplikacije z uporabo IFrameov.....	21
Slika 13 - Princip delovanja Facebook aplikacije z asinhronskimi klici. [28].....	23
Slika 14 - Primer uporabnikovega profila.....	26
Slika 15 - Na spodnjem delu zaslona je menu aplikacij.....	26

Slika 16 - Zavihek <i>Info</i>	27
Slika 17 - Diagram primerov uporabe.....	31
Slika 18 - Diagram zaporedja za primer uporabe <i>Dodaj nagradno igro</i>	34
Slika 19 - Diagram zaporedja za primer uporabe <i>Poglej nagradne igre</i>	35
Slika 20 - Diagram zaporedja za primer uporabe <i>Žrebaj</i>	36
Slika 21 - Diagram zaporedja za primer uporabe <i>Sodeluj</i>	37
Slika 22 - Diagram zaporedja za primer uporabe <i>Priporoči</i>	37
Slika 23 - Entiteni diagram.....	38
Slika 24 - Prikaz arhitekture MVC. [34].....	39
Slika 25 - Prikaz trinivojske arhitekture.....	40
Slika 26 – Skica arhitekture MVC kot del trinivojske arhitekture. [38].....	41
Slika 27 - Desno zgoraj se nahaja kontrolna plošča s seznamom naših aplikacij.....	43
Slika 28 - Vzpostavljanje nove aplikacije.....	43
Slika 29 - Nadzorna plošča aplikacije.....	44
Slika 30 - Zaslonska maska - Na levi zahteva za dovoljenje pošiljanje elektronske pošte, na desni zahteva dovoljenja za objavo obvestil.....	52
Slika 31 - Uporabnik lahko dovoljenja kadarkoli nastavi (ukine) pod nastavitvami dovoljenj posamezne aplikacije.....	52
Slika 32 - Vnosno polje s pisanjem vnaprej.....	53
Slika 33 - Facebook že uporablja koledarček za izbiro datumov, ki pa ni na voljo preko oznak jezika FBML.....	54
Slika 34 - Koledarček ki ga generira moja koda.....	55
Slika 35 - Vnos nagradne igre.....	56
Slika 36 - Pregled nagradnih iger.....	57
Slika 37 - Uporabnik odgovarja na nagradno vprašanje.....	59
Slika 38 - Ko odgovori, se med novicami na njegovem profilu prikaže obvestilo o udeležbi.....	59
Slika 39 - Izbiranje prijateljev za pošiljanje vabil.....	61
Slika 40 - Preden sistem pošlje vabilo, lahko uporabnik pregleda sporočilo ali doda lastno.....	61
Slika 41 - Elektronska pošta, ki jo prejme uporabnik ob žrebu.....	63

Seznam tabel

Tabela 1 – Seznam direktnih povezav za izvedbo specifičnih akcij.....	28
Tabela 2 - Seznam podatkov o uporabniku, ki jih smemo hraniti neomejeno.....	42
Tabela 3 - Nastavitve: Osnova.....	45
Tabela 4 - Nastavitve: Avtorizacija namestitve.....	45
Tabela 5 - Nastavitve: Platno.....	46
Tabela 6 - Nastavitve: Povezava.....	48
Tabela 7 - Nastavitve: Gradniki.....	48
Tabela 8 - Nastavitve: Napredno.....	49

Seznam uporabljenih kratic in simbolov

XML eXtensible Markup Language	Razširljiv označevalni jezik.
HTML HyperText Markup Language	Označevalni jezik za določitev strukture dokumentov, ki se prenašajo po spletu in pregledujejo s spletnimi brskalniki.
URL Uniform Resource Locator	Internetni naslov, na katerem se nahaja vsebina.
JSON JavaScript Object Notation	Format izmenjave podatkov.
API	Aplikacijski programski vmesnik.
WIKI	Spletna stran, ki jo lahko obiskovalci z uporabo enostavnih pravil oblikovanja hitro spreminjajo in urejajo.
OpenSocial	Platforma za razvoj aplikacij, namenjenih spletnim socialnim mrežam, alternativa Facebook platformi.
IFrame	Oznaka jezika HTML, ki omogoča, da je spletna stran razdeljena na segmente. Vsak segment lahko prikazuje svoj dokument.
REST Respresentational State Transfer	Koncept programske arhitekture, namenjen porazdeljenim sistemom, kot je svetovni splet. Opisuje, kako naj bodo definirani in naslovljeni porazdeljeni viri s poudarkom na enostavnosti izmenjave informacij in razširljivosti.
Tag	Oznaka.
FBML FaceBook Markup Language	Označevalni jezik, ki ga uporablja Facebook.
JS JavaScript	Skriptni jezik, ki omogoča programski dostop do objektov v odjemalcu. Integriran v brskalnik odjemalca.
FBJS FaceBook JavaScript	Facebookova alternativa JS.

XFAML	FBML, ki omogoča asinhrono klice funkcij vmesnika API Facebook platforme.
FQL Facebook Query Language	Facebookov povpraševalni jezik.
PU Primer Uporabe	Tipičen primer uporabe aplikacije.
MVC Model-View-Controller	Tip arhitekture.
PHP Hypertext Preprocessor	Splošnonamenski skriptni jezik, uporabljen tipično pri razvoju spletnih aplikacij.
SSM	Spletna Socialna Mreža

Povzetek

Facebook je trenutno največji, a le eden med mnogimi primeri spletnih socialnih mrež. V tem trenutku ima prek tristo petdeset milijonov aktivnih uporabnikov. Masa ljudi oglaševalcem ponuja nove pristope in načine prenosa svojih sporočil uporabnikom.

V svoji diplomski nalogi sem opredelil kako so uporabniki na spletnih socialnih mrežah med seboj povezani, kako so se mreže razvijale in kaj spletna socialna mreža sploh je. Skozi oči potencialnega oglaševalca sem preučil možnosti izvedbe virusnega marketinga med uporabniki.

V drugem delu sem na kratko predstavil tehnologije, ki se uporabljajo pri povezovanju aplikacij s Facebook platformo. Raziskal sem tudi možne integracijske točke, preko katerih se posamezna aplikacija lahko poveže s končnim uporabnikom.

Razvil sem prototip aplikacije, ki omogoča izvedbo nagradnih iger. Aplikacija omogoča dodajanje nagradne igre z enim nagradnim vprašanjem in več možnimi odgovori, med katerimi je en pravilen. Uporabniki Facebooka lahko v nagradni igri sodelujejo z odgovorom na vprašanje.

V aplikacijo sem vključil elemente, ki omogočajo lažje in hitrejše virusno širjenje. Takšni elementi so na primer avtomatično pošiljanje sporočil, priporočil prijateljem, pošiljanje elektronske pošte itd.

V zaključku pojasnim prednosti in pomanjkljivosti Facebook platforme, ki sem jih spoznal med izvajanjem svoje aplikacije.

Ključne besede: virusni marketing, spletna socialna mreža, Facebook platforma, FBML, FBJS, XFBML, FQL

Abstract

Facebook is currently the largest social network site out there, but certainly not the only one. At this time Facebook has over three hundred and fifty million active users. The mass of people offers advertisers new approaches and methods of delivering their messages to users.

In my bachelor's thesis I begin by defining how users of social network sites are connected, how these networks evolved and what a social network site really is. I examine the feasibility of viral marketing through the eyes of potential advertiser.

In the second section I briefly introduced technologies used to connect applications with the Facebook platform. I researched possible integration points through which application can be linked to end-users.

I developed an application prototype that allows raffles to be played. Application enables you to add a raffle with one question and multiple-choice answers, of which one is correct. Facebook users can participate in the raffle if they correctly answer to the provided question.

I have included elements that allow easier and faster viral spread. Such elements, for example, automatically send messages and recommendations to friends, send e-mails etc.

In conclusion I explain the advantages and disadvantages of Facebook Platform that I have learned during my work.

Key words: viral marketing, social network site, Facebook Platform, FBML, FBJS, XFBML, FQL

1. Uvod

Internet lahko brez pomislekov uvrstimo med glavne iznajdbe dvajsetega stoletja. Za marsikoga je v zadnjem desetletju ali dveh postal glavni in nepogrešljivi vir pridobivanja informacij, komuniciranja, nakupovanja, učenja, druženja in zabave. Njegov vpliv sega tako v osebne kot v poslovne odnose.

Količina dostopnih informacij in spremenjene razmere so začele vplivati tudi na načine, kako podjetja gradijo odnose s svojimi strankami, ter na delo tržnikov, katerih oglaševalski postopki se počasi, a vztrajno selijo s klasičnih medijev in za večjo konkurenčnost iščejo vedno nove oblike oglaševanja. Med množico različnih pristopov se je uveljavil pojem »virusni marketing«, ki kot osnovo oglaševanja uporablja obstoječe socialne mreže.

Oglaševalci tovrstne metode uporabljajo čedalje bolj pogosto. Večina teh oglaševanj, če se omejim zgolj na svetovni splet, je osnovana na obstoječih in prosto dostopnih storitvah. V manjšini so oglaševalci, ki orodja virusnega marketinga izdelajo sami. Prva skupina internetno ozaveščenih oglaševalcev za promocijo uporablja sezname elektronskih naslovov in oblikovane skupine v okviru spletnih socialnih mrež. Drugi, ki imajo na voljo več sredstev, na svojih spletnih straneh prirejajo različne nagradne igre in tekmovanja, ki se lahko povezujejo tudi z drugimi mediji, kot sta na primer televizija ali radio.

Zanimivo je, da le malo oglaševalcev uporablja in pozna možnosti, ki jih ponujajo spletne socialne mreže, na katerih so potrošniki praktično oddaljeni le za miškin klik.

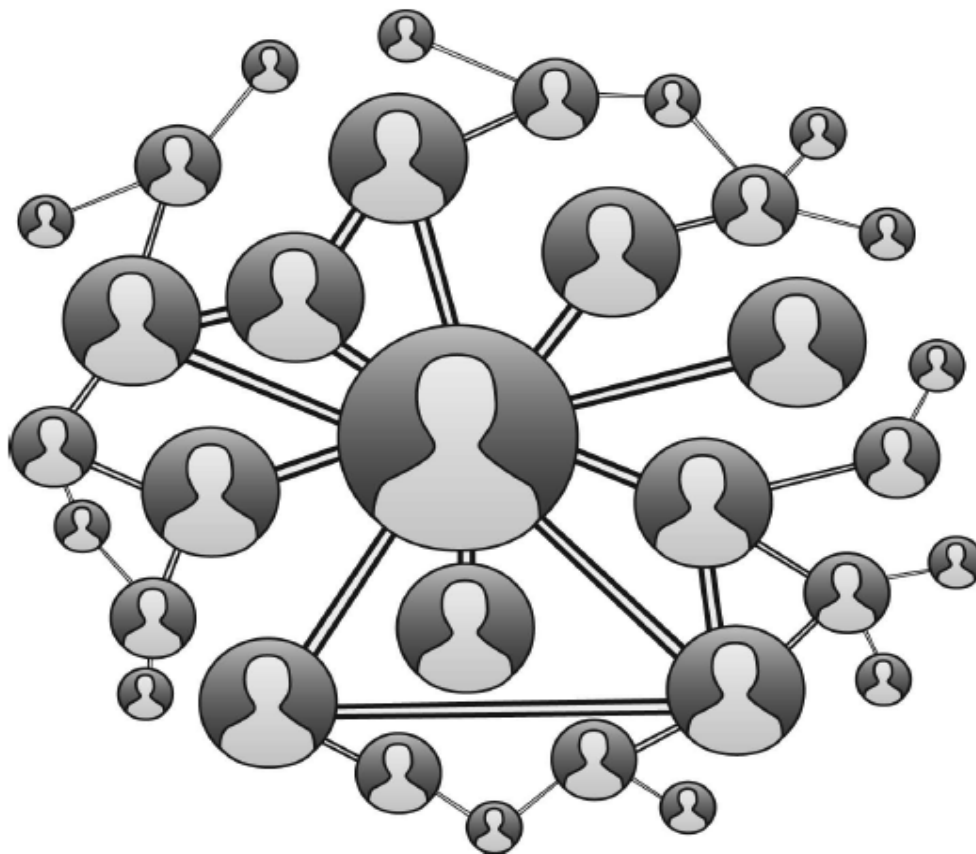
Facebook je trenutno največji, a le eden med mnogimi primeri spletnih socialnih mrež. V tem trenutku ima prek tristo petdeset milijonov aktivnih uporabnikov. Najpomembnejše za oglaševalce pa je, da jim ponuja orodja in načine, kako tem uporabnikom prenesti svoje sporočilo.

V svoji diplomski nalogi bom skušal ugotoviti potrebe oglaševalcev in opredeliti sredstva, ki jim jih za doseg njihovih ciljev lahko ponudimo računalničarji z razvojem aplikacij za spletne socialne mreže.

Preveril bom tudi zahtevnost razvoja spletnih aplikacij za spletne socialne mreže in zrelost razvojnega okolja. Zasnoval bom aplikacijo za izvedbo nagradnih iger ter razvil enostaven prototip.

2. Socialni graf

Družbo si lahko predstavljamo kot ogromen graf, kjer smo ljudje vozlišča, povezave pa predstavljajo razmerja med nami. Dva človeka sta lahko povezana na različne načine; kot starš-otrok, sodelavec-sodelavec, fant-dekle itd. Socialni graf vsakega posameznika je drugačen, edinstven, lahko bi ga primerjali celo s prstnim odtisom. [1]



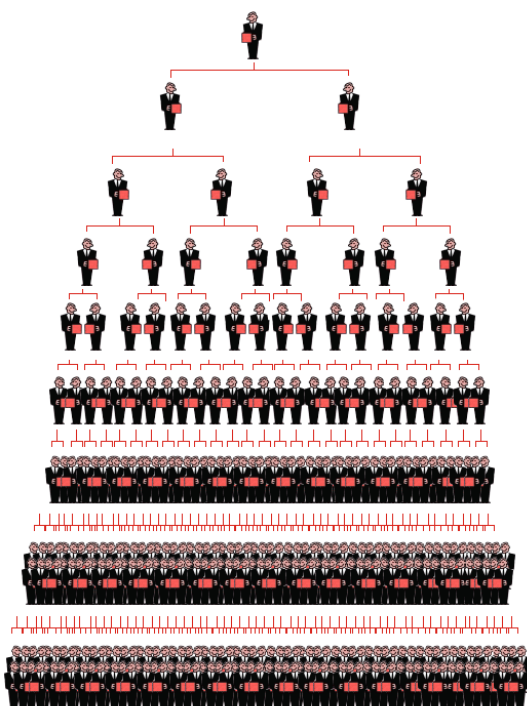
Slika 1 - Skica prikazuje socialni graf osebe A, ki je postavljena v središče. Osebe, ki so neposredno povezane, so zarisane malenkost manjše, osebe, oddaljene dva koraka, še manjše itd. [2]

Teorija grafov in mreženja sprva deluje zgolj kot obrobna tema za teoretike. V resnici imajo mreže ogromno vrednost, kar lahko ponazorimo na primeru telefonov: en sam telefon je popolnoma neuporaben, uporabnost raste z vsakim dodatnim telefonom v mreži, ker skupno število lastnikov telefonov predstavlja število ljudi, ki jih lahko pokličemo. Metcalfeov zakon pravi, da je vrednost mreže enaka kvadratu števila vseh uporabnikov v sistemu. [3]

Večjo dodano vrednost ponuja mrežna storitev posamezniku, pogosteje se bo vanjo vključeval in uporabljal njene storitve. Aktivno vključeni posamezniki, ki storitvi prostovoljno namenjajo svojo pozornost, pa v današnji ekonomiji pomenijo zagotovilo za neposreden ali posreden dobiček prek marketinga.

3. Virusni marketing

Značilnosti virusov so, da se lahko prenašajo s človeka na človeka in hitro zajamejo širše populacije, v primernem okolju in z ugodnimi pogoji se lahko razmnožujejo eksponentno. Pri tem pa ne prenašalec in ne žrtve nimajo nadzora nad okužbo. Ko je človek enkrat z virusom okužen, praviloma nanj razvije imunost.



Slika 2 - Hitro širjenje in prenos ideje na prijatelje.

Virusni marketing se ni pojavil zaradi interneta. Njegovi začetki segajo že v pradavnino, saj v najbolj primitivni obliki temelji na širjenju sporočila od ust do ust. Je pa res, da je internet povzročil njegov preporod in uveljavitev. Slovenci lahko idejo virusnega marketinga lepo ponazorimo s svojim pregovorom, ki pravi, da »dober glas seže v deveto vas«. Še bolj pa ustreza navedek Victorja Hugoja, ki pravi, da »z nobeno vojsko ni mogoče ustaviti ideje, katere čas je napočil«.

3.1 Opredelitev pojma virusni marketing

Izraz »virusni marketing« sta leta 1996 skovala Steve Jurvetson in Tim Draper, da bi opisala marketinško strategijo storitve Hotmail. Vsakemu elektronskemu sporočilu je bil na koncu dodan pripis v obliki pošiljateljevega priporočila »Zagotovi si svojo osebno, brezplačno elektronsko pošto s Hotmailom na <http://www.hotmail.com>«. Hotmail si je s to strategijo v sedmih mesecih pridobil več kot 10 milijonov uporabnikov. [4]

Virusni marketing je ideja, ki se širi, in ideja, ki pomaga tržiti, medtem ko se širi. Je posebna oblika ideavirusa ali modne ideje, ki se prenaša prek dela populacije in ki uči, spreminja in vpliva na vsakega, ki se ga dotakne. Je strategija, ki omogoča in vzpodbuja ljudi, da posredujejo marketinško sporočilo in sodelujejo pri govoricah ter s tem ustvarijo osnovo za eksponentno rast izpostavljenosti in vpliva sporočila. Odvisen je predvsem od visoke stopnje posredovanja sporočil.

3.2 Izvedba virusnega marketinga

Ob pravilni izvedbi marketinške akcije se oglasno sporočilo med uporabniki širi hitro in nevsiljivo prek medsebojne komunikacije. Podjetje lahko marketinški virus ustvari z določeno zanimivo in uporabno informacijo, brezplačno elektronsko knjigo, partnerskimi programi, elektronskimi priporočili, zanimivimi video posnetki, interaktivnimi igrkami, slikami,

Opis velja tako za resnične viruse kot za računalniške. V obeh primerih je prizvok izjemno negativen. Iste značilnosti veljajo tudi za virusni marketing, le da negativni prizvok tu ni povsem upravičen. Virusni marketing namreč poudarja princip, da naj podjetja prenehajo s trženjem do ljudi in ustvarijo okolje, da bodo ljudje tržili drug drugemu.

Zgornji opis velja tako za resnične viruse kot računalniške viruse. V obeh primerih je prizvok izjemno negativen. Iste značilnosti veljajo tudi za virusni marketing, le da negativni prizvok tu ni povsem upravičen. Poudarja namreč princip, da morajo podjetja prenehati s trženjem ljudem in začeti ustvarjati okolje, v katerem bodo ljudje tržili drug drugemu.

tekstovnimi sporočili itd. Osnovna oblika virusnega marketinga ni določljiva in se stalno spreminja.

Za izvedbo uspešnega virusnega marketinga je cilj oglaševalcev izbrati posameznike z visokim potencialom socialne mreže in jih okužiti z virusnim sporočilom, za katerega so ti posamezniki dojemljivi. Tako okuženi posamezniki postanejo prenašalci sporočila, ki ga prenesejo na svoje prijatelje in znance v socialni mreži. Le-ti ponovno postanejo prenašalci ...

3.3 Elementi virusnega marketinga s primeri

Nekatere strategije virusnega marketinga delujejo bolje kot druge. Lastnosti tipičnih strategij lahko strnemo v šest točk. Ni nujno, da je za izvedbo virusnega marketinga pokritih vseh šest točk, v osnovi pa velja, da več točk, kot jih izvedba pokriva, večja je možnost uspeha. [5]

3.3.1 Ponudba brezplačnih izdelkov ali storitev

»Brezplačno« je beseda, ki je poleg besede »novo« najpogosteje uporabljena beseda v oglaševanju. Virusni marketing uporablja princip zakasnjene nagrade. Podjetje dobička ne bo vpisalo zdaj ali jutri, vendar bo ob dovolj velikem zanimanju ustvarilo bazo odjemalcev, ki zagotavlja dobiček v prihodnosti. Podari nekaj, prodaj nekaj.

Dairyalea, podružnica prehrambenega giganta Kraft, je za promocijo svojih prigrizkov za otroke pripravila spletno igro Cheesymovies. Uporabniki so lahko v njej z različnimi pripomočki in zvočnimi učinki posneli svoje kratke risanke. Ob koncu oglaševanja so tekmovalci glasovali za najboljši izdelek. Priredili so celo virtualno podelitev oskarjev.

Campaign Magazine je akcijo uvrstil med najboljših deset spletnih oglaševalskih akcij 2002. Povprečen gost je na strani preživel od trideset minut do ene ure. [6]

3.3.2 Omogoča enostaven prenos na druge

Virusi se širijo le, ko so za to izpolnjeni pogoji. Primerni mediji so elektronska pošta, spletne strani, slike, video ... Za prenos sporočila mora zadoščati pritisk ali dva na miškin gumb, potek prenosa ne sme biti zapleten, idealno se naj vključi v obstoječo komunikacijo prenašalcev.

Najbolj enostaven primer, kjer uporabnik virus širi celo brez dodatnega dela, je v Hotmail.com. Uporabniki njihovo storitev uporabljajo zaradi storitve same, torej zaradi brezplačnega pošiljanja elektronske pošte. Ob vsakem poslanem sporočilu posredujejo tudi reklamo za spletni servis. Nobenega dodatnega napora za uporabnika, velika prednost za ponudnika.

3.3.3 Lahka razširitev

S hitrim širjenjem uporabnikov lahko nastanejo težave. Če ponujamo storitev, moramo razmišljati o prihodnosti. Ali so naše zmogljivosti dovolj velike, da bomo sposobni storitev ali brezplačni izdelek zagotoviti tisočem okuženih potrošnikov?

Socialna spletna mreža Friendster je v prvih treh mesecih obratovanja dosegla tri milijone uporabnikov. Ker nihče ni načrtoval tako hitre širitve, po tehnični strani niso bili ustrezno

pripravljeni in prihajalo je do rednih izpadov. Uporabniki so si zato začeli izmenjevati elektronske naslove in komunicirati po ustaljenih poteh mimo spletne strani.

3.3.4 Izrablja pogosta obnašanja in motivacijo

Abraham Maslow je leta 1943 v svoji motivacijski teoriji predstavil hierarhijo človekovih ciljev in osnovnih potreb, ki se stopnjujejo od osnovnih fizičnih potreb, potreb po varnosti, ljubezni in pripadnosti, spoštovanju in samouresničitvi. [7]

Maslowa teorija je med prvimi, ki se jih oglaševalci naučijo, in velja v vseh tipih oglaševanja. Virusni marketing ni izjema.

Google je z virusnim marketingom izstrelil Gmail, v zadnjem času pa se ravno prek virusnega marketinga širi njegova nova storitev Google Wave. Najprej je bila na voljo le nekaj razvijalcem, ki so se udeležili Googlove I/O konference maja 2009. Septembra so nabor uporabnikov razširili in poslali sto tisoč povabil v omrežje. Vsak povabljeni uporabnik je imel na voljo še dvajset ali trideset povabil, ki jih je lahko razdelil po lastni presoji. Hkrati so objavili video posnetke, kaj vse lahko naredimo z novim produktom. Kdor je prišel do uporabniškega računa, je imel nekaj več kot drugi, bil je v prednosti in v središču pozornosti, saj so se njegovi prijatelji in znanci potegovali za povabila, ki jih je imel. Da se je na ta način ustvarila prava tekmovalnost, dokazujejo tudi ponudbe na spletni tržnici e-Bay, kjer so povabila prodajali po okoli sto ameriških dolarjev, rekordna ponudba pa je bila kar pet tisoč dolarjev. [8]

3.3.5 Uporablja kombinirane vire komunikacije

Dobra ideja se širi po različnih medijih. Priporočljiva je kombinacija marketinga na spletnih socialnih mrežah, povezava mrež (na primer Facebooka in Youtuba) ter oglaševanja s klasičnimi pristopi.

Leta 2007 je nogometni klub Sporting Portugal oglaševal prodajo vstopnic. Na svoji spletni strani je objavil video posnetek, ki so si ga obiskovalci z navdušenjem pošiljali drug drugemu. Preden se je posnetek zavrtel, je moral obiskovalec vpisati svojo telefonsko številko. Posnetek je prikazoval igralce, ki čakajo v slačilnici, medtem ko njihov trener telefonira obiskovalcu in razloži, da enostavno ne morejo pričeti sezone, dokler uporabnik ne kupi sezonske vstopnice. Odlična integracija ter sinhronizacija videa in telefonskega klica sta obiskovalce presenetila in navdušila. V 24 urah so zabeležili prek dvesto tisoč ogledov in telefonskih klicev. [9]

3.3.6 Uporablja tuje vire

Strategije virusnega marketinga se mnogokrat nanašajo na uporabo tujih virov, torej po nekaj prvih okužbah se virusna ideja širi sama in ne zahteva stroškov vzdrževanja.

Takšne vrste so na primer video posnetki, pri katerih podjetje vloži začetna sredstva za izdelavo posnetka in nekaj malega za sejanje, stroške za posredovanje sporočila pa nosijo potrošniki sami.

3.4 Prednosti virusnega marketinga

Prednosti virusnega marketinga so predvsem nizki stroški. Kot je že omenjeno, po začetni investiciji v sporočilo in začetno sejanje postane občinstvo brezplačen medijski kanal. Virusna oglaševanja so nevsiljiva in se potrošnikom zdijo privlačna. Potekajo na podlagi osebnih priporočil znancev, kar povečuje kredibilnost in zanimivost sporočila. Dosegajo lahko težko dosegljiva občinstva z možnostjo selektivnega ciljanja. V določenih primerih je trajnost lahko neomejena, saj se ideja prenese tudi v medije in predvsem zapisi v blogih in spletnih straneh ostanejo za vedno. Pomembno je, da virusni marketing ne izključuje drugih marketinških aktivnosti in obstoječih sporočil, čeprav lahko deluje samostojno. [4]

3.5 Slabosti virusnega marketinga

Virusni marketing ima žal tudi svoje negativne lastnosti. Nadzorovanje oglaševalskega sporočila je oteženo in v večini primerov omejeno zgolj na začetek virusnega oglaševanja. Možno je spreminjanje sporočila v teku širjenja. Sporočilo lahko tudi uide izven okolja, za katerega je bilo namenjeno, in se nenadzorovano razširi, kar posledično lahko predstavlja grožnjo lokalnim marketinškim strategijam. Kombinacija spreminjanja sporočila, nenadzorovano širjenje in malo nepremišljenosti v začetku lahko hitro privede do negativnih občutkov. Negativne občutke lahko sproža tudi pretirano prejetje sporočil, saj je ideja sveža, nova in zanimiva le prvič. [4]

4. Definicija spletne socialne mreže

Spletno socialno mrežo (SSM) lahko opišemo kot spletno storitev, ki v osnovi posameznikom omogoča [10]:

- izvedbo bolj ali manj javnega profila znotraj sistema;
- se povezati z drugimi uporabniki (narediti svojo socialno mrežo), pri čemer je bolj v ospredju prenos že obstoječih znanstev (povezav) iz realnega življenja na splet kot spoznavanje novih;
- pregledovati seznam povezav drugih uporabnikov v sistemu.

Za pridružitve SSM se mora uporabnik registrirati tako, da odgovori na obrazec z vprašanji, ki so tipično o naslovu elektronske pošte, spolu, starosti, naslovu bivanja, interesnih področjih itd. Zbrani podatki se prikažejo v profilu uporabnika, ki je njegov virtualni odsev. Podatke v profilu lahko kasneje praviloma poljubno urejamo ali dodajamo.

Mreža po registraciji uporabnika pozove, naj poišče svoje prijatelje, pri čemer so oznake povezav lahko različne.

Večina SSM strani omogoča tudi puščanje sporočil na prijateljevem profilu (komentarji) in zasebna sporočila, podobna sistemu elektronske pošte. V drugih uporabnih funkcijah se SSM strani razlikujejo.

Omenil sem že Metcalfeov zakon, ki govori o vrednosti mreže v odvisnosti od števila uporabnikov. Vendar mora mreža uporabnikom že v osnovi ponuditi nekaj, s čimer lahko

zadovoljijo svoje potrebe. Glede na sklope zadovoljevanja socioloških potreb bi lahko zgradbo spletnih socialnih mrež razčlenili na identiteto, prisotnost, odnose in razmerja, pogovor, skupnost, ugled in souporabo. [11]

4.1.1 Identiteta

Obstaja način identifikacije ljudi v sistemu. Facebook za identiteto uporablja ime in priimek uporabnikov. Čeprav lahko uporabnik ime spremeni, le-to ostane shranjeno v bazi. Kljub spremembi se ob iskanju po prvotnem imenu še vedno izpiše med zadetki, čeprav z novim imenom. Poleg tega se uporabnik lahko identificira tudi z uporabniškim imenom, ki je pravzaprav skrajšan URL naslov do njegovega profila (v obliki www.facebook.com/uporabniskoime).

4.1.2 Prisotnost

Facebook nima neposrednega načina za ugotavljanje prisotnosti, kot ga imajo nekatere druge SSM strani, na katerih je prisotnost označena na uporabnikovem profilu. Trenutno prisotnost vseeno lahko ugotovimo, če se sprehodimo čez seznam prijateljev, prijavljenih v klepet, ki je sestavni del Facebooka. Seveda obstaja možnost, da uporabnika ni na seznamu, kar pomeni dvoje – ali res ni prisoten ali pa je med možnostmi izbral, da se njegovo ime ne prikazuje.

Uporabniki posredno prisotnost sporočajo tudi s pošiljanjem prispevkov, komentiranjem prijateljevih prispevkov in posodabljanjem statusa.

4.1.3 Odnosi in razmerja

Uporabnik doda drugega uporabnika med prijatelje. Pri Facebooku morata zahtevo za prijateljstvo potrditi oba. Včasih je bilo treba ob dodajanju novega prijatelja določiti tudi vrsto zveze, zdaj je to opuščeno.

4.1.4 Pogovor

Omogočena je komunikacija med uporabniki. Facebook omogoča pisanje na uporabnikov profil (zid), za katerega se uporablja izraz komentiranje in je javno vidno vsem drugim, ki imajo dostop do profila. Poleg komentiranja je na voljo še pošiljanje zasebnih sporočil, ki deluje na podobnem principu kot elektronska pošta. Aprila 2008 so razvijalci Facebooka dodali še klepet, ki omogoča pogovor ena-na-ena v realnem času. [12]

4.1.5 Skupnost

Vsak uporabnik lahko ustvari skupino in vanjo povabi svoje prijatelje. Stran skupine poleg komentiranja in nalaganja fotografij podpira tudi enostaven sistem forumov za izmenjavo mnenj.

Možna je izvedba posebnih strani za oboževalce, na katerih se lahko predstavljajo zvezdniki, različne organizacije, podjetja, izdelki ipd. Ko uporabnik postane oboževalec nečesa, poudari nekaj o sebi in se z nečim poenoti.

4.1.6 Ugled

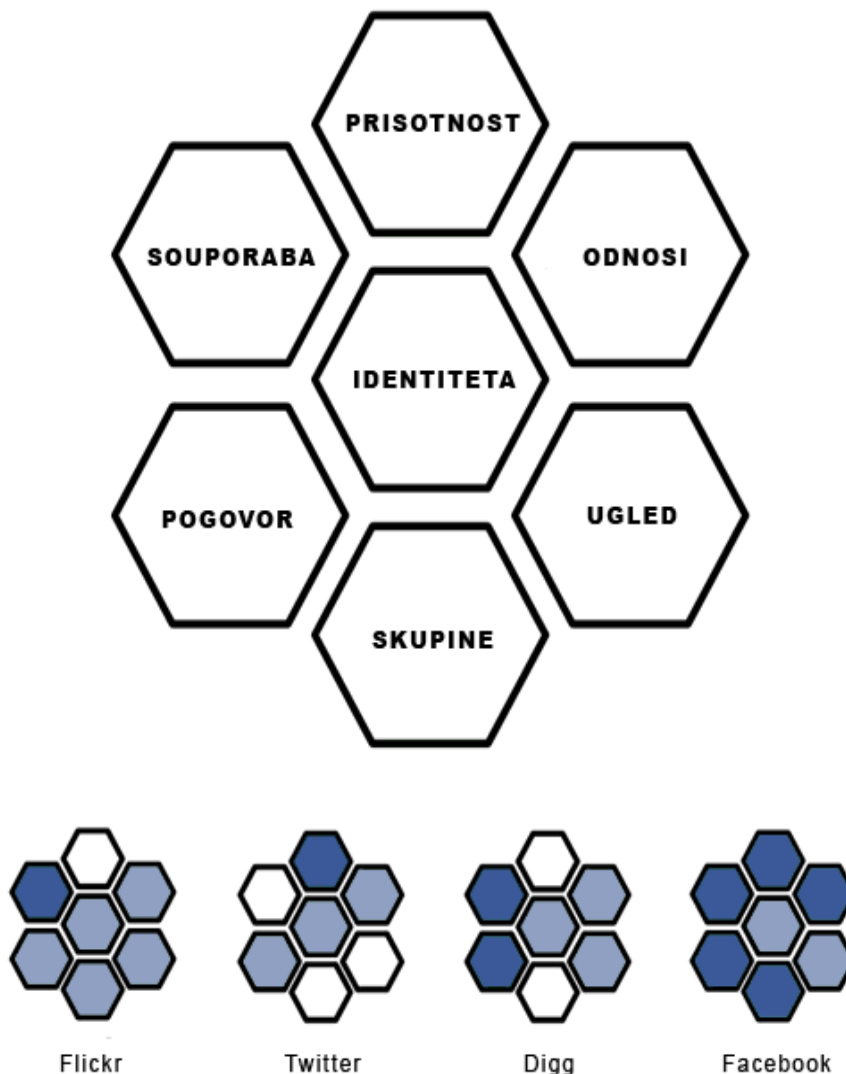
Obstaja način merjenja statusa uporabnikov – kdo je dober uporabnik, komu lahko zaupamo itd. Facebook celovitega merjenja uporabnika nima, ker ga niti ne potrebuje. Celovito

merjenje bi prišlo v poštev, če bi mrežo gradili z nepoznanimi uporabniki – takrat bi informacija o ugledu prišla prav.

Namesto dodajanja k ugledu uporabnika lahko posamezne prispevke nagrajujemo s klikom na povezavo »Všeč mi je«.

4.1.7 Souporaba

Prek Facebooka lahko organiziramo foto albume in vanje nalagamo fotografije, označujemo ljudi na njih in mesto, kjer je bila fotografija posneta. Fotografije delimo z drugimi uporabniki. Kdo vse si jih lahko ogleda, je odvisno od nastavitvev. Medtem ko je velikost fotografij omejena (če jo presežemo, jo sistem avtomatično pomanjša), je količina fotografij neomejena. Facebook dovoljuje tudi nalaganje video posnetkov, pisanje prispevkov, posodabljanje statusov, s katerimi prijatelje obveščamo, kaj v danem trenutku počnemo itd.



Slika 3 - Skica prikazuje gradnike spletnih socialnih mrež, spodnje skice pa prikazujejo poudarke posameznih gradnikov, ki jih imajo konkretni primeri spletnih socialnih mrež. Kot je razvidno, ima Facebook med primeri najbolj uravnoteženo zgradbo. [11]

5. Kratek pregled zgodovine spletnih socialnih mrež

5.1 Oglasne deske

Predhodniki zdaj poznanih socialnih omrežij so bili konec sedemdesetih let sistemi tako imenovanih oglasnih desk (*BBS – Bulletin Board Systems*). Ponavadi so oglasne deske gostovale na osebnih računalnikih in uporabnik je moral z modemom poklicati gostiteljev računalnik, da je lahko pustil sporočilo.



Slika 4 - Primer oglasne deske

5.2 IRC, ICQ in pogovorne aplikacije

Program IRC (*Internet Relay Chat*) je luč sveta ugledal leta 1988. Uporabljali so ga za izmenjavo datotek, povezav in klepet ena-na-ena ali skupinsko. Je oče programov za klepet, kakršne poznamo danes. Sprva je bil namenjen le sistemom UNIX, sredi devetdesetih pa se je razširil tudi na druge operacijske sisteme (npr. mIRC za Windows, ki je izšel leta 1995). [13][14]

Program ICQ je bil naslednji korak v razvoju in je bil namensko razvit za operacijski sistem MS Windows. Poleg lastnosti, ki jih je imel program IRC, je omogočal tudi omejeno pošiljanje SMS sporočil, možnost nadaljevanja pri pošiljanju datotek (*Resumable File Transfer*), pošiljanje voščilnic, igranje igrice in iskanje po direktoriju uporabnikov. Vsaj delno je odgovoren za popularizacijo avatarjev, kratic (*LOL – Laughing Out Loud*, *BRB – Be Right Back*) in smeškov.

5.3 Zgodnje spletne socialne mreže

Strani za zmenke bi lahko jemali za neko obliko prvih spletnih socialnih mrež. Uporabniki so ponavadi lahko izdelali svoj profil in na nek način kontaktirali drug drugega. Vendar te strani niso omogočale seznama prijateljev, kar je bistvena možnost zdaj znanih spletnih socialnih mrež.

5.3.1 SixDegrees.com

Leta 1997 so vzpostavili stran Sixdegrees.com, ki je ime povzela po ideji o šeststopenjski oddaljenosti oseb (»Six degrees of separation« - če je oseba korak stran od vsake osebe, ki jo pozna, in dva koraka stran od vsake osebe, ki jo pozna vsaj ena oseba, od katere so oddaljeni en korak, potem so vsi oddaljeni največ šest korakov od katerekoli osebe na Zemlji. [15]).

Zdaj jo obravnavamo kot prvo moderno spletno socialno mrežo. Uporabnikom je omogočala izvedbo profila in prijateljstvo drug z drugim. Čeprav stran zdaj ni več prisotna, je bila nekaj časa zelo obiskana in je imela v vrhuncu nekaj čez milijon obiskovalcev. Leta 2000 je podjetje *Youth Stream Media Networks* kupilo stran za 125 milijonov dolarjev. V začetku 2001 je stran propadla, ker niso uspeli razviti ustreznega poslovnega modela, ki bi prinašal dobiček. [10][16]

Med leti 1997 in 2001 se je začelo pojavljati čedalje več socialnih strani (med pomembnejšimi **AsianAvenue**, **BlackPlanet** in **MiGente**), ki so omogočale različne načine urejanja profilov – poslovne, zasebne in namenjene iskanju partnerjev. Mreženje in zbiranje prijateljev je bilo enostransko in potrditveno uporabnika, ki ste ga dodali med prijatelje, ni bila potrebna.

5.3.2 LiveJournal

Omeniti velja še spletno stran **LiveJournal** (www.livejournal.com), ki je z delovanjem začela leta 1999. Medtem ko so njeni predhodniki uporabnikom ponujali bolj ali manj statične profile, je LiveJournal zgrajena okoli stalno spreminjajočih vsebin – uporabniških blogov. Uporabnike spodbuja k združevanju v skupine ter sledenju drug drugemu prek vnesenih vsebin.

5.3.3 Ryze.com

Novi val spletnih socialnih mrež se je začel leta 2001 z **Ryze.com**, namenjenim povezovanju podjetnikov [17]. Ustanovitelj Adrian Scott, ki je bil v San Franciscu organizator različnih dogodkov z namenom mreženja, je iskal nove načine povezovanja. Idejo je sprva predstavil svojim prijateljem podjetnikom, večinoma članom Poslovne in tehnološke skupnosti San Francisca (*San Francisco Business And Technology Ccommunity*). Podatek je pomemben, ker so bili osebno in poslovno vpleteni tudi ljudje, ki so kasneje stali za Tribe.net, LinkedIn in Friendster.

5.3.4 Friendster

Friendster velja za prvo moderno splošno namensko spletno socialno mrežo. Ustanovljena je bila leta 2002 kot konkurenca dobičkonosni strani za zmenkarije Match.com. Medtem ko so strani za zmenkarije uporabljale idejo spoznavanja popolnih neznancev, je Friendsterjeva strategija temeljila na spoznavanju prijateljevih prijateljev, ki bi po ideji morali biti bolj združljivi kot popolni neznanci.

Stran je uporabnike pridobivala z nepričakovano hitrostjo, saj je v prvih mesecih delovanja brez kakršnegakoli proračuna za marketing dosegla vrtooglave tri milijone uporabnikov [18]. Zaradi tehnične nepripravljenosti in nedovršenosti programja, predvsem zasnove podatkovne baze, je prihajalo do rednih izpadov. Zaradi tega so si uporabniki namesto obiskovanja počasne ali ob trenutkih nedostopne strani raje začeli izmenjavati naslove elektronske pošte in komunicirati po ustaljenih poteh. Poleg tega je koncept omogočal ogled profilov ljudi, ki so bili do štirih stopenj oddaljeni. Zaradi tega so uporabniki začeli masovno dodajati zanimive profile in si s tem povečevali svoj doseg. Pojavili so se tudi prvi goljufi (*Fakesters*, skovanka ki se uporablja z lažnimi profili strani Friendster), ki so se predstavljali z namišljenimi imeni

in slikami. Sledil je tudi tako imenovani socialni zlom – uporabniki, ki so na stran v prvi vrsti prihajali z namenom sproščenega združenja, so bili poleg bližnjih prijateljev soočeni tudi s svojimi starimi znanci, sodelavci ter šefi.

Friendster je na goljufe odgovoril z aktivnim čiščenjem (brisanjem) profilov. Vendar so med upravičenimi brisali tudi vse profile resničnih uporabnikov, ki niso uporabljali realističnih fotografij, ter vse profile organizacij ali podjetij. S tem je z uporabniki zanetil vojno, saj so lažni profili postali priljubljena opcija – uporabniki so se zabavali z brskanjem po profilih namišljenih osebnosti, lahko pa so imeli takšni profili tudi uporabno vlogo (npr. »Harvard University«) za iskanje vseh ljudi, ki so jih poznali. [10]

Zaradi omenjenih vzrokov se je zanimanje za stran zmanjšalo. Kljub temu ima Friendster prek stodeset milijonov registriranih uporabnikov, zanimivo pa je, da jih kar devetdeset odstotkov dostopa iz Azije (Filipini, Malezija, Indonezija, Združeni Arabski Emirati, Singapur ...). [18]

Od avgusta 2006 Friendster omogoča različne vsebinske dodatke na uporabniškem profilu. Za razvoj in uvedbo svojih aplikacij imajo razvijalci prost dostop do API-ja. Friendster's Developer Program je odprta, nelastniška platforma z odprtim sistemom prihodkov, ki podpira tako OpenSocial kot Facebook Platformo. [19]

5.4 Sodobne spletne socialne mreže

Od leta 2003 so se spletne socialne mreže pojavljale kot gobe po dežju. Večina je želela ponoviti zgodnji uspeh Friendsterja in je privzela obliko s centralnim profilom, na katerega se vse navezuje. Druge so bile bolj demografsko naravnane, na primer strani, namenjene poslu, kot so LinkedIn, Visible Path in Xing. Pojavile so se strani, usmerjene v interesna področja, kot so Dogster.com (namenjen ljubiteljem psov), Care2.com (namenjen aktivistom), Couchsurfing.org (povezuje popotnike z ljudmi, ki so pripravljeni ponuditi prenočišče), MyChurch (združuje krščanske cerkve in njene pripadnike) ...

Fenomen hitre, virusne rasti spletnih socialnih mrež so začele implementirati tudi spletne strani, katerih prvotni namen je bil deljenje multimedijskih vsebin, kot so Flickr (fotografije), Last.FM (glasba) in YouTube (video).

5.4.1 MySpace

MySpace so razvili leta 2003 in do junija 2006 [20] je tudi na račun razočaranih uporabnikov Friendsterja postal najbolj popularna spletna socialna mreža v ZDA.

MySpace je uporabnikom omogočil popolno prilagoditev izgleda profilov, uporabniki so lahko dodajali glasbo, fotografije in vključevali video iz zunanjih strani. Lastnosti so pritegnile različne glasbene skupine in nočne klube, ki so na enostaven in brezplačen način lahko izdelale svoje strani ter z minimalnim vložkom poskrbele za marketing z razdajanjem brezplačnih simboličnih daril (CD-ji, majice, VIP vstopnice itd). Na MySpaceu se je tako oblikovala zanimiva simbioza med glasbeniki, ki so lahko na preprost način kontaktirali in novačili nove pripadnike, in glasbenimi navdušenci na drugi strani, ki so pridobili delček pozornosti svoje najljubše glasbene skupine in dodatno izrazili svojo pripadnost.

MySpace je stalno dodajal nove funkcionalnosti in se prilagajal uporabnikom. Leta 2008 so postavili razvojno platformo MDP (*MySpace Developer Platform*), ki razvijalcem omogoča pisanje lastnih MySpace funkcionalnosti. MDP je osnovan na OpenSocial platformi. [21]

5.5 Facebook

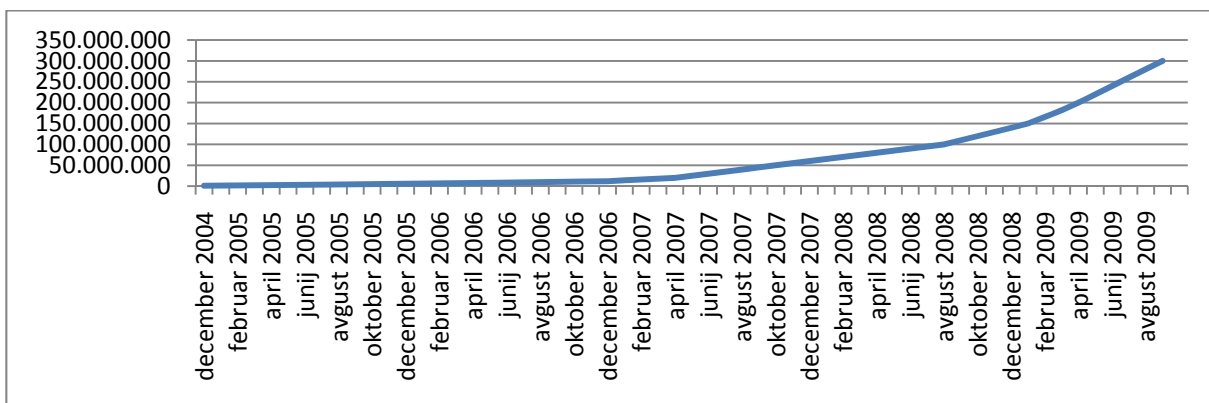
Začetki Facebooka segajo v konec leta 2003, ko je študent s Harvarda Mark Zuckerberg vdrl v zaščitena področja lokalne mreže in zbral fotografije študentov vseh domov Harvarda. Fotografije je objavil na strani Facemash – dve fotografiji drugo ob drugi, obiskovalci pa so izbirali, katera je lepša.

Stran je administracija Harvarda zaprla, Zuckerbergu pa je grozila kazen in izključitev zaradi vdora, kršenja avtorskih pravic in zakona o zasebnosti. Na koncu so bile obtožbe umaknjene, Zuckerberg pa je začel naslednji semester z razvojem nove strani. 4. februarja 2004 je objavil stran thefacebook.com. O strani je povedal nekaj prijateljem. V prvem dnevu se je novica razširila in stran je imela čez tisoč dvesto uporabnikov.

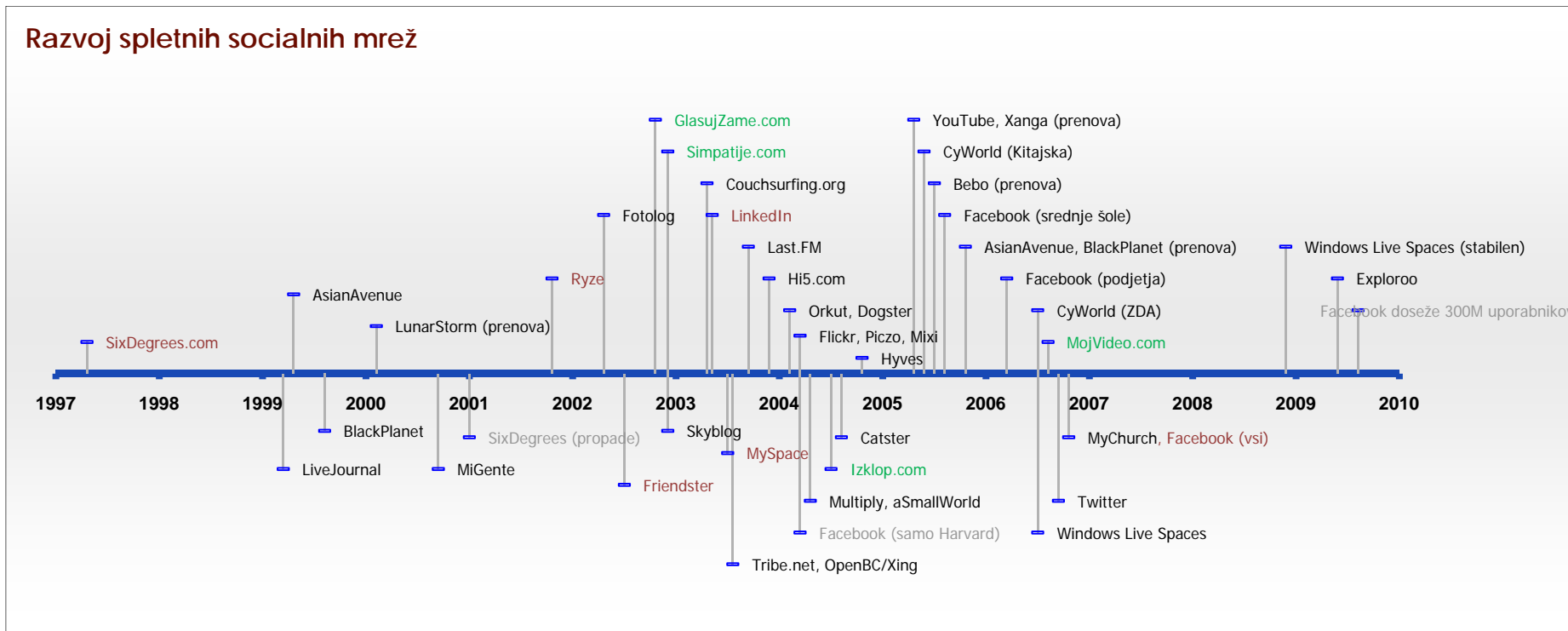
Članstvo je bilo sprva omejeno na študente Harvarda, nato so ga marca 2004 razširili na univerze Stanford, Columbia in Yale ter postopoma na vse univerze v ZDA in Kanadi.

Poleti 2004 so ustanovili podjetje Facebook, Inc. in leta 2005 prešli na domeno facebook.com, ki so jo kupili za \$200.000. [12]

Septembra 2005 so prijavo omogočili tudi srednješolcem, a le s povabilom. Za njimi pa še nekaterim večjim podjetjem, kot sta Apple Inc. in Microsoft. Šele septembra 2006 je Facebook postal odprt za vse, ki so starejši od 13 let. Že decembra je dosegel 12 milijonov aktivnih uporabnikov (uporabnikov, ki so se prijavili v zadnjih 30 dnevih), oktobra 2007 50 milijonov uporabnikov, 100 milijonov v avgustu 2008 in čez 300 milijonov v septembru 2009 [22].



Slika 5 - Prikaz števila aktivnih uporabnikov Facebooka. Za aktivne uporabnike veljajo tisti registrirani uporabniki, ki so se vsaj enkrat v zadnjih tridesetih dnevih prijavili na stran.



Slika 7 - Časovni trak nastajanja spletnih socialnih mrež, označene so le najpomembnejše – vseh je na tisoče. Točen čas nastanka je zelo težko določiti, saj vsaka stran ob navajanju podatkov upošteva subjektivne kriterije, kaj začetek sploh pomeni. Nekatere za začetek štejejo datum prve registracije domene, druge datum prijave prvega uporabnika na stran, tretje so obstajale že dolgo časa, preden so izpolnjevale pogoje za socialno mrežo. Podatki so pridobljeni iz vira označenega pod številko [10], spletne storitve Alexa.com, datume slovenskih ponudnikov pa sem izvedel s povpraševanjem preko elektronske pošte.

Z rdečo barvo so označene posamezne pomembnejše strani, z zeleno so označene slovenske spletne socialne mreže.

Po letu 2007 je označenih zanemarljivo število spletnih socialnih mrež, ker je večina po tem letu razvitih skupnosti še v vzponu in so praktično neprepoznave proti zgoraj navedenim, ki uporabnike štejejo v milijonih. Pri vseh obstoječih spletnih socialnih mrežah so po letu 2007 prisotne stalne izboljšave, vendar dramatičnih novosti ni.

6. Razvojne možnosti

Po definiciji spletne socialne mreže mora le-ta uporabnikom zagotavljati izvedbo lastnega profila, povezovanje z drugimi uporabniki in pregledovanje seznama povezav med drugimi uporabniki. Hiter vpogled v katerokoli moderno spletno skupnost odkrije, da tem pogojem zadoščajo na različne načine in ponujajo še vrsto drugih možnosti, s katerimi lahko pritegnejo uporabnike.

Imamo torej množico funkcionalnosti, ki so zgrajene okoli osnovnega jedra, kar ustreza modularnemu pristopu aplikacijskega razvoja.

Hitrejša možnost predajanja aplikacije v uporabo kot posledica modularnega razvoja pomeni možnost odzivanja na želje in potrebe uporabnikov ter odzive na poteze konkurentov. Modularni pristop ima še eno ključno lastnost – moduli so funkcionalno zaključene celote, kar pomeni razbremenitev razvijalskih skupin, saj lahko določene sklope sprogramirajo med seboj popolnoma neodvisne ekipe.

Razvijalci spletnih socialnih mrež so modularni pristop ponesli še korak dlje. Moduli, razen osnovnega jedra, so tako šibko povezani, da jih imenujejo kar aplikacije. Aplikacije se v skupni sklop povežejo s pomočjo platforme (del osnovnega jedra), ki se jo obravnava kot vmesnik API (*Application Programming Interface*), torej sklop programskih knjižnic, ki omogočajo delo in programiranje brez podrobnega poznavanja notranjega delovanja.

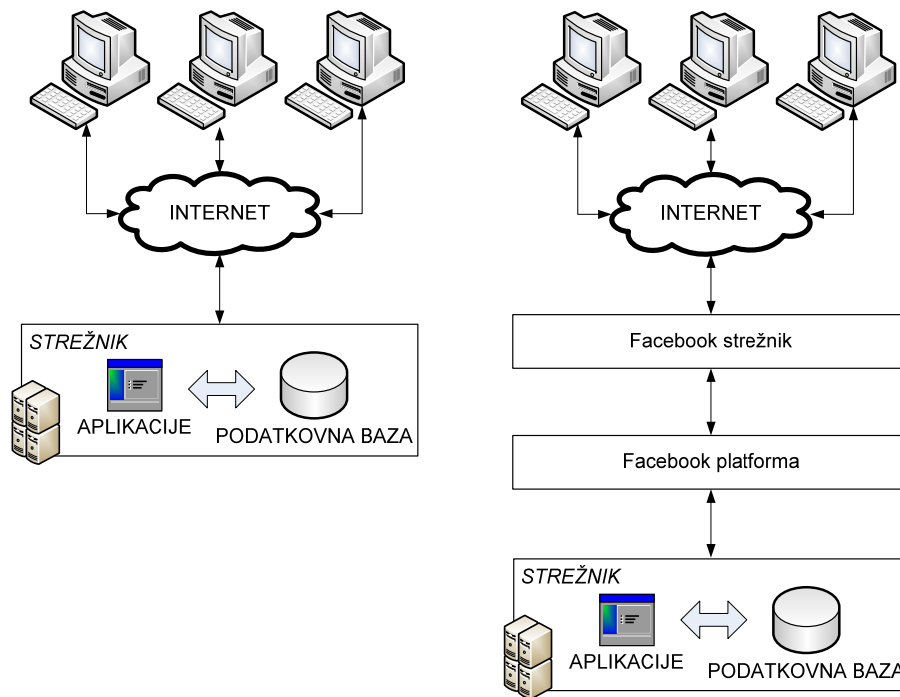
Platforme omogočajo naprednejšim uporabnikom pisanja lastnih dodatkov za priljubljene socialne mreže. V okviru Facebook Platform na primer deluje na tisoče različnih aplikacij in sto tisoč registriranih razvijalcev. [23]

Večina Facebook aplikacij ima značilno virusno naravo, ki nudi enkratno zabavo – uporabniki jih dodajo in uporabijo le enkrat ali dvakrat. Ob prvi uporabi se novica o novi aplikaciji razpošlje vsem prijateljem uporabnika, ki si tudi sami dodajo in uporabijo aplikacijo. Pojav je marsikomu moteč in se ga označuje kot »aplikacijski spam« [12]. Ko virusni val poide, aplikacija največkrat potone v pozabo.

Facebook platformi se je kot konkurenčni odziv MySpacea v povezavi z Googleom pridružila še tako imenovana OpenSocial platforma.

6.1 Facebook platforma

Splavitev vmesnika API Facebook platforme v avgustu 2006 [23] je razvijalcem omogočila vključitev podatkov (za katere uporabniki dovolijo) na zunanje strani in namizne aplikacije ter s tem povečanje obsega socialne povezanosti. Prve pol leta obstoja Facebook platforme so aplikacije lahko tekle samo znotraj Facebooka, kasneje pa se je odprla in svojo arhitekturo ponudila tudi drugim. To pomeni, da aplikacije, napisane za Facebook, lahko praktično nespremenjene tečejo tudi na drugih spletnih straneh ali mrežah.



Slika 8 - Standardni pristop k izvedbi spletnih aplikacij je postavitve aplikacije skupaj s podatkovno bazo na enoten strežnik, do katerega lahko uporabniki dostopajo preko svetovnega spleta s pomočjo unikatnega URL-ja. Facebookov model je malenkostno drugačen. Uporabnik do aplikacije dostopa preko Facebooka, čeprav je aplikacija dejansko nameščena na svojem strežniku. Facebook nastopa kot dvosmerni proksi med brskalnikom uporabnika in aplikacijo, pri čemer prestra in spreminja podatkovni tok v realnem času (na primer spreminja vsa imena JS spremenljivk). S tem Facebook ohranja nadzor nad izvajanjem.

Aplikacije pridobivajo podatke z različnimi vmesniki (avtorizacijski vmesnik, vmesnik za dostop do funkcij jedra, vmesnik za povabila itd.) in s povpraševalnim jezikom FQL (*Facebook Query Language*). Aplikacija se uporabniku izriše znotraj Facebook platna (*Canvas*), ki je glavni način interakcije s končnim uporabnikom.

Razvoj aplikacij za Facebook je zelo podoben razvoju klasičnih aplikacij z uporabo vmesnikov API. Vendar se morajo razvijalci poleg uporabe Facebook API-ja naučiti še povpraševalnega jezika FQL (*Facebook Query Language*), označevalnega jezika FBML (*Facebook Markup Language*) in skriptnega jezika FBJS (*Facebook JavaScript*). FBML je neke vrste okrnjeni HTML, specifikacija ki Facebookovim strežnikom narekuje, kako prikazati izgled posamezne aplikacije. Zapis označevalnega jezika FBML je shranjen v predpomnilniku, dokler ga ustrezní klic vmesnika API ne zamenja. Podobno kot FBML je Facebook razvil tudi svojo alternativo JavaScriptu FBJS (*Facebook JavaScript*).

Omeniti je treba še Facebook Connect, ki ni neposredno povezan s Facebook platformo, a je pri razvoju in integraciji Facebook mreže na tretjih straneh nepogrešljiv. Omogoča registracijo uporabnikov na enem mestu, z istim uporabniškim imenom in geslom pa se uporabniki lahko prijavljajo na različnih spletnih straneh in niso omejeni le na področje Facebooka. Podobno kot Facebook platforma tekmuje z OpenSocial, Facebook Connect na nek način tekmuje z OpenID.

6.2 OpenSocial

Še preden je Facebook ponudil svoj model API-ja tretjim stranem, je Google v navezi z MySpace-om predstavil OpenSocial. Glavna prednost je bila odprta implementacija API-ja za vse socialne mreže, ki je takrat Facebook še ni ponujal – aplikacije, napisane za Facebook platformo, so tekle samo znotraj Facebooka. Kasneje je Facebook, kot že omenjeno, to odpravil in prednosti OpenSociala so se zmanjšale.

Aplikacije se lahko prikazujejo kot enostavni dodatki (*gadget*) in pridobivajo informacije o socialnem grafu neposredno v času izvajanja. Potekajo lahko tako na odjemalcu kot na strežniku. Aplikaciji je dodeljen prostor za gostovanje, potem potek dogodkov krmili sama, uporabnikov brskalnik komunicira direktno z aplikacijo.

V primeru razvoja strežniške aplikacije, je le-ta podoben klasičnemu pristopu, pri čemer je manjši poudarek na API-ju kot pri Facebooku. V primeru aplikacije na strani odjemalca, programerji ponavadi vključujejo tretje JS knjižnice, kot je na primer Google Gears ali ExtJS. Prednost razvoja za OpenSocial je, da temelji na podpori uveljavljenih standardov HTML in JavaScript in se ni potrebno učiti novih programskih jezikov.

Kritike OpenSocial letijo na varnost in kompatibilnost, ki jo obljublajo. V času pisanja te diplomske naloge je najnovejša verzija OpenSocial 0.9 in je še vedno v beta fazi, kar pomeni, da bodo pomanjkljivosti sčasoma odpravljene.

V beta fazi je tudi OpenSocket, spletna storitev, ki pretvori aplikacije, napisane za OpenSocial, v Facebook aplikacije.

Trenutno je nemogoče napovedati, ali bo na dolgi rok prevladal Facebook platform ali OpenSocket, gre za boj velikanov. Za izvedbo manjših aplikacij je to praktično nepomembno, za izvedbo obsežnejših zadev pa je pri arhitekturnem načrtovanju smiselno upoštevati, da bo v prihodnosti lahko prevladal en ali drug in uporabiti vsaj trinivojsko arhitekturno zasnovo.

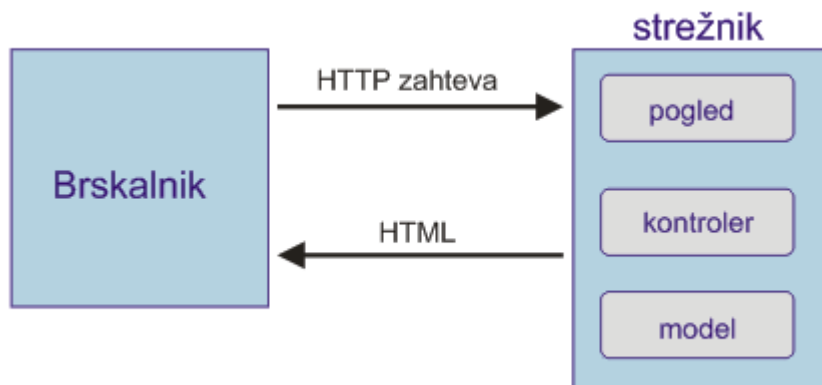


Slika 9 - Skica trinivojske zasnove aplikacije. Predstavitveni nivoji so ločeni, logika v ozadju ostaja enaka. [24]

7. Facebook Platform

7.1 Princip delovanja klasične spletne aplikacije

Če malo poenostavim potek dogodkov pri klasični spletni aplikaciji, ga lahko opišem na sledeč način. Uporabnik preko uporabniškega vmesnika v svojem brskalniku sproži HTTP zahtevo s klikom na povezavo, gumb ali kakšen podoben element. Spletna aplikacija HTTP zahtevek sprejme in ga ustrezno obdela. Brskalniku v odzivu pošlje HTML, ki se prikaže uporabniku pred računalnikom.



Slika 10 - Princip delovanja klasične spletne aplikacije.

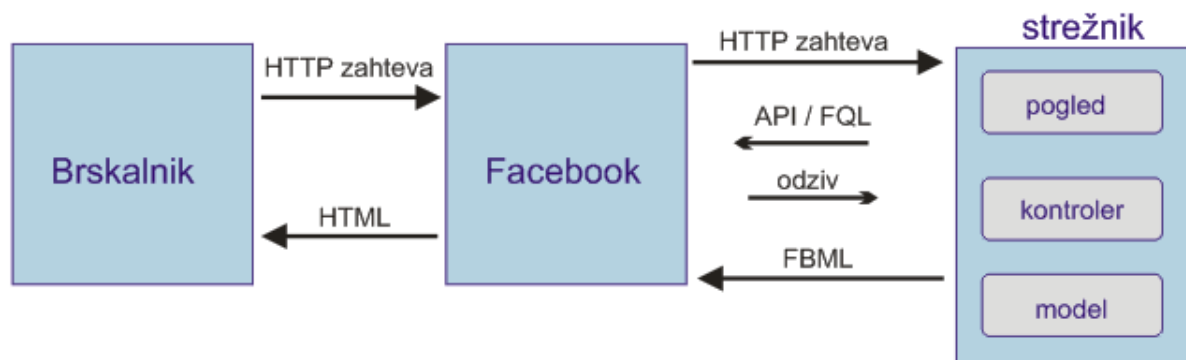
7.2 Princip delovanja Facebook aplikacije

7.2.1 Klasična Facebook aplikacija z uporabo FBML

Facebook aplikacija deluje po drugačnem principu. Uporabnik pošlje HTTP zahtevek na Facebook, ki ga interno obdela in pošlje naprej naši aplikaciji.

Določen sklop podatkov, predvsem tisti o uporabnikih, niso hranjeni v lokalni podatkovni bazi, temveč ga hrani Facebook. Do teh podatkov lahko dostopamo preko specifičnih klicev funkcij vmesnika API ali kodiranjem povpraševanj FQL (*Facebook Query Language*).

Aplikacija mora na koncu Facebooku na HTTP zahtevek odgovoriti v formatu označevalnega jezika FBML. Facebook ponovno interno predela odgovor in ga vključi v svoje ogrodje tako, da FBML transformira v standardni označevalni jezik HTML ter pošlje nazaj brskalniku.

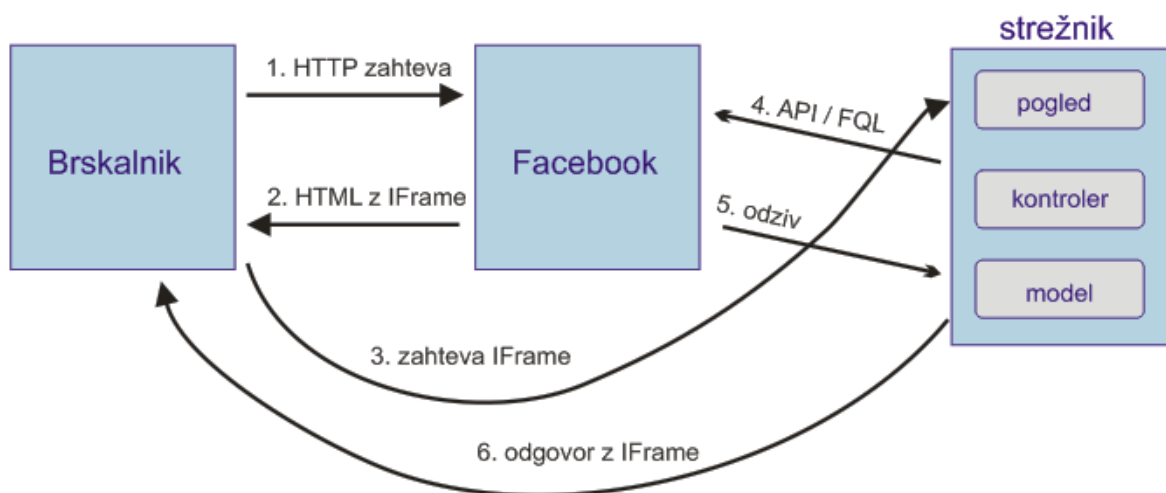


Slika 11 - Princip delovanja Facebook aplikacije z uporabo označevalnega jezika FBML in klici funkcij vmesnika API.

7.2.2 Facebook aplikacija z uporabo IFrame

Poleg prej opisane možnosti Facebook ponuja alternativo - delo z IFramei.

Princip delovanja je tudi tu dokaj enostaven. Ko uporabnik sproži zahtevo za aplikacijo, Facebook pripravi spletno stran, ki okvirja IFrame aplikacije. Naslov IFramea kaže na strežnik aplikacije. Facebook naslovu pripne svoje parametre, ki aplikaciji podajajo več informacij o prijavljenem uporabniku. Ti parametri med drugim dopuščajo preverjanje, da je zahteva res prišla s strani Facebooka itd.



Slika 12 - Princip delovanja Facebook aplikacije z uporabo IFrameov.

7.3 Facebook API

Ne glede katero od opisanih opcij izberemo, se vsa komunikacija odvija preko klicev funkcij vmesnika API, ki so razporejene v skupine (funkcije administracije, funkcije za prijavo in avtorizacijo, funkcije za pridobivanje podatkov, funkcije za založništvo, funkcije Facebook povezave, funkcije za mobilni Facebook, funkcije za delo s fotografijami, funkcije za delo z dogodki, funkcije za lastne oznake). Komunikacija poteka po principu REST. [25]

7.3.1 REST – Representational State Transfer

REST je koncept programske arhitekture, namenjen porazdeljenim sistemom, kot je svetovni splet. Opisuje kako naj bodo definirani in naslovljeni porazdeljeni viri s poudarkom na enostavnosti izmenjave informacij in razširljivosti.

Bistveni koncept pri REST je obstoj virov informacij, ki so naslovljeni z globalno identifikacijo, kot je pri HTTP na primer URI (*Uniform Resource Identifier*). Za upravljanje s temi viri omrežne komponente med seboj komunicirajo preko vnaprej definirane vmesnika (kot je HTTP) in si izmenjujejo predstavitve teh virov (dokumenti z informacijami).

Vsaka od komponent lahko posreduje zahteve, vendar se zaveda in razume le svoje lastne zahteve. Tako mora aplikacija za uspešno sporazumevanje z virom informacije poznati le dve stvari: identifikacijo vira informacije in zahtevano akcijo. Aplikacija mora seveda razumeti tudi format dokumenta, ki ga prejme kot odziv, tipično je to ponavadi HTML, XML ali JSON, lahko tudi slika, navaden tekst itd. Pomembno je, da aplikaciji za uspešno sporazumevanje ni potrebo poznati, ali se na poti uporablja predpomnilnik, požarni zid, tuneli, posredovalni strežnik (proksi) itd. [26]

7.3.2 Odgovori vmesnika API in lokalizacija

Odgovori na klice funkcij vmesnika API so tipično v formatu kot jih pozna PHP pri poljih, formatu JSON ali XML.

Odgovori so privzeto lokalizirani, če lokalizacija obstaja. Uporabniku, ki dostopa do Facebooka v jeziku, ki ni enak ameriški angleščini, bo vmesnik API vračal odgovore, prilagojene njegovemu jeziku. Seveda pod pogojem, da je aplikacija, ki jo uporabnik pregleduje, prevedena in so prevodi v administraciji omogočeni.

V administraciji aplikacije se izbere le privzeti jezik, administracija prevodov je popolnoma ločena. Glavna prednost, ki jo platforma ponuja, je možnost, da lahko uporabniki sami prevajajo našo aplikacijo v svoj jezik.

Trenutno Facebook podpira več kot 65 jezikov med katerimi je tudi slovenščina. [27]

7.3.3 Primerjava FBML IFrame

Splošno gledano ima FBML prednost pred IFramei.

IFrame aplikacije morajo namreč vedno izvesti API klice, da lahko vkomponirajo socialno vsebino, kot je na primer uporabniško ime, slika uporabnika itd. Ti API klici pa pomenijo dodaten prenos po omrežju, zaradi česar je IFrame počasnejši kot FBML.

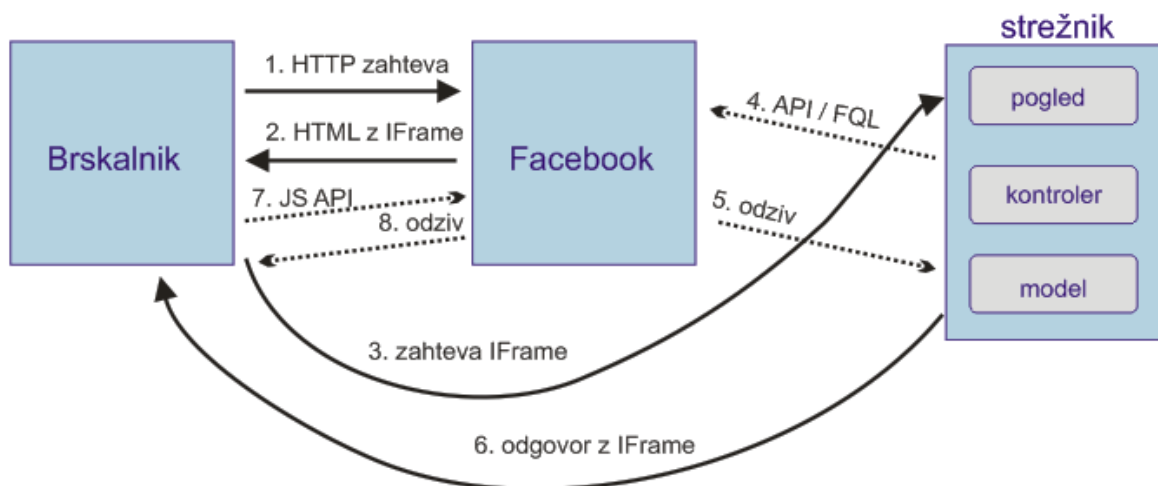
Poleg tega so Facebook strežniki praviloma hitrejši in ponujajo višje hitrosti prenosa ter enakomerno hitrost ne glede na lokacijo končnega uporabnika. To je z uporabo aplikacije, ki teče na enem strežniku, po možnosti še deljenem, težko doseči.

7.4 JavaScript vmesnik API

Dobro leto nazaj je Facebook zaradi uvedbe Facebook Connect predstavil XFBML, ki omogoča JS API klice, po principu AJAX (*Asynchronous JavaScript and XML*). V tem

primeru večina API klicev med aplikacijo in Facebookom ni več potrebnih. Aplikacija brskalniku pošlje odgovor, v katerem manjkajo podatki, ki se nahajajo na Facebooku. Te podatke potem brskalniki zahteva z JS klici vmesnika API.

V tem primeru se ob začetnem prihodu uporabnika na aplikacijo znotraj Facebooka sicer zgodijo koraki od ena do osem. Potem se večino časa odvijata predvsem koraka sedem in osem.



Slika 13 - Princip delovanja Facebook aplikacije z asinhronskimi klici. [28]

7.5 FBML

FBML je kratica za Facebook Markup Language in je v osnovi podmnožica HTML (*Hyper Text Markup Language*), z dodanimi Facebooku specifičnimi oznakami.

Možna je tudi razširitev FBML-ja z definiranjem lastnih oznak. Definicija se izvede s pomočjo API klica `Fbml_registerCustomTags()`, v katerem kot parameter podamo oznake, zakodirane v format JSON. Oznake lahko uporabljamo za lastno uporabo ali jih damo na voljo tudi drugim razvijalcem. V času pisanja je možnost pisanja razširitev za FBML v fazi beta testiranja.

7.6 XFBML

Facebook uporablja XFBML kot način vključevanja FBML-ja na zgenerirano HTML stran.

XFBML uporablja lastne oznake, med katerimi je tudi `fb:serverfbml`. Ta omogoča razčlenbo FBML-ja na Facebookovih strežnikih in prikaz znotraj IFramea. Oznake XFBML morajo biti vedno pravilno zaprte z namenom spoštovanja DOM-a in pravilnega prikaza v vseh brskalnikih.

Za uporabo XFBML-ja je potrebno v nastavitvah aplikacije nastaviti URL povezave. Več o nastavitvah aplikacije bo razloženo kasneje.

Na svojem strežniku je treba ustvariti datoteko `sd_receiver.htm`. Datoteka služi kot referenca na preko-domensko komunikacijsko knjižnico. Knjižnica odjemalca JavaScript jo

uporablja za vzpostavljanje povezave med spletnimi stranmi, Facebookom in storitvami brskalnika.

7.7 FBJS

FBJS (*Facebook JavaScript*) je rešitev za razvijalce, ki v svojih aplikacijah potrebujejo uporabo JavaScripta. Ko Facebook prejme FBJS, ga razčleni in pred vsa imena funkcij in spremenljivk doda predpono z identifikacijsko številko aplikacije. S tem se doseže, da so posamezne aplikacije omejene in ne posegajo v področja drugih.

Glavne razlike med FBJS in JS se pokažejo pri obravnavi elementov DOM. Lastnosti objektov niso dostopne direktno kot pri JS, temveč preko `getter` in `setter` funkcij (JS `name` je dostopen preko `getName` in `setName`). [29]

7.8 FQL

FQL je kratica za Facebook Query Language, ki omogoča dostop do podatkov podobno kot pri drugih jezikih strukturiranega povpraševanja SQL. Stavki povpraševanja so v obliki `SELECT [polja] FROM [naziv_tabele] WHERE [pogoji]`. Po potrebi sta na voljo še `ORDER BY` in `LIMIT`. Pod `FROM` lahko navedemo le eno tabelo, v čemer se FQL razlikuje od standardnih jezikov SQL. Podpovpraševanja lahko izvedemo z `IN`. Dodatna omejitev FQL je, da lahko iščemo le po indeksiranih stolpcih.

Določeni stolpci tabel so zasnovani v obliki polj. Taka je na primer tabela uporabnikov `users`, ki med drugim v stolpcu `education_history` tipa polje hrani informacije o šolanju uporabnika. FQL omogoča dostop do polj v celoti ali le do njihovih delov. Recimo `SELECT education_history` ali `SELECT education_history.name`.

Večino podatkov lahko dobimo tudi s klici funkcij vmesnika API, ki so v resnici le ovojnice v katerih se skrivajo povpraševanja FQL. Prednosti FQL pred uporabo klicev API so [30]:

- Zmanjšanje količine podatkov, ki se prenašajo. Namesto pridobivanja vseh možnih informacij, ki jih vrača generična funkcija vmesnika API, lahko natančno opredelimo zahtevane podatke.
- Zmanjšanje količine povpraševanj. Z enim zahtevnejšim povpraševanjem FQL lahko nadomestimo več posameznih klicev funkcij vmesnika API. Če za pridobivanje informacij o uporabnikovih prijateljih uporabljamo klasične funkcije API, bi najprej poklicali `friends_get()` in nato za vsak posamezen zadetek še funkcijo `users_getInfo()`. Klice lahko nadomestimo z enim povpraševanjem v stilu `SELECT uid, name, birthday FROM user WHERE uid IN (SELECT uid2 FROM friend WHERE uid1 = $userId)`.
- Omogoča poenoten dostop do podatkov, ni se potrebno učiti in iskati v množici funkcij vmesnika API.

Definiramo lahko tudi več FQL povpraševanj in jih izvršimo naenkrat tako, da jih v formatu JSON podamo kot parameter funkciji `fql_multiquery()`.

8. Anatomija Facebook aplikacije

Facebook aplikacije so lahko v najrazličnejših oblikah in obsegih. Facebook ima predstavljenih devet bistvenih integracijskih točk, s katerimi se posamezna aplikacija lahko poveže s končnim uporabnikom. [31]

8.1.1 Imenik aplikacij (Application directory)

Seznam Facebook aplikacij, kjer je vsaka aplikacija predstavljena s kratkim opisom, logotipom in oceno uporabnikov.

8.1.2 O aplikaciji (About page)

Facebook stran z opisom aplikacije, do katere uporabniki pridejo preko imenika aplikacij.

8.1.3 Platno aplikacije (Canvas page)

Glavna stran aplikacije, ki je lahko v formatu FBML ali zunanja stran, prikazana v IFrameu. Vsebuje tudi kontrolno ploščo aplikacije (*User Dashboard*).

8.1.4 Profil

Facebook profil je predstavitev uporabnikove identitete. Na profilu je možnih več vrst integracije.

8.1.4.1 Okvir na profilu (Profile box)

Okvirji na profilu ponavadi nosijo zadnje posodobljene informacije o akcijah uporabnika in odražajo uporabnikovo željo po predstavljanju svoje identitete preko aplikacije. Uporabnik lahko okvir aplikacije namesti v sam profil (na glavno stran) ali pod zavihek okvirjev.



Slika 14 - Primer uporabnikovega profila.

8.1.4.2 Menu aplikacij

Menu aplikacij bi lahko primerjali s Start menujem v MS Windows. Je menu levo spodaj v opravilni vrstici z bližnjicami do nameščenih Facebook aplikacij.

8.1.4.3 Priljubljene

Uporabnik lahko aplikacijo doda med priljubljene. Priljubljene aplikacije se izpisujejo v opravilni vrstici poleg menuja aplikacij.



Slika 15 - Na spodnjem delu zaslona je menu aplikacij.

8.1.4.4 Zavihek okvirji (Boxes tab)

Pod zavihkom okvirja se prikazujejo okvirji nameščenih aplikacij. Uporabnik lahko posamezne okvirje premika na glavno stran profila in obratno.

8.1.4.5 Odsek za informacije o aplikaciji (Application info section)

Pod zavihkom *Info* lahko uporabniki predstavijo svoje interese, delovne izkušnje, izobrazbo itd. na strukturiran in urejen način. Pod zavihek lahko uporabniki dodajo tudi informacije o aplikaciji.


Personal Information

Interests: jazz, frisbee, california,
 Favorite Music: John Coltrane, Joanna Newsom
 Favorite Quotes: We have to take the car, because the bike's on fire – Los Campesinos

Education and Work

College: Harvard '07
 Computer Science
 High School: Cheltenham High School '03
 Employer: Facebook
 Position: ???

My Smilies

Good Smilies:  **Happy**
 The original and still undefeated.

Slika 16 - Zavihek Info

8.1.4.6 Založnik (Publisher)

Aplikacija se lahko poveže tudi z založnikom, tako da objavlja na uporabnikovem zidu (*wall*) ali zidovih njegovih prijateljev.

8.1.4.7 Obrazci krmilnika

Obrazci krmilnika so posebna komponenta FBML, ki aplikacijam omogoča objavljanje zgodb krmilnika v imenu uporabnika.

8.1.5 Nastavitve zasebnosti

Uporabniki lahko uredijo nastavitve zasebnosti, kar vključuje tudi dovoljenja za objavljanje novic preko založnika ali krmilnika novic, kje se informacije o aplikaciji nahajajo (glavna stran profila, zavihek okvirjev ...) in kdo lahko elemente aplikacije vidi.

8.1.6 Krmilnik novic (Feed)

Aplikacije lahko dostopajo do uporabnikovih novic in dodajajo vsebino. Na primer, ko uporabnik doda novo fotografijo, aplikacija objavi novico o novo objavljeni fotografiji med uporabnikove novice.

8.1.7 Opozorila (Alerts)

Aplikacije lahko uporabnikom pošljejo obvestila preko elektronske pošte. Uporabnik mora vsaki posamezni aplikaciji dovoliti pošiljanje pošte.

8.1.8 Vabila (Request)

Aplikacija lahko generira vabila, ki se pojavijo v desnem zgornjem kotu domače strani uporabnika. Vabila so poslana s strani uporabnikovih prijateljev in navadno od uporabnika zahtevajo neke vrste odziv.

8.1.9 Interakcija z uporabniki (Interacting with users)

Facebook API ne omogoča izvajanje Facebook specifičnih akcij direktno, aplikacija lahko to obide z uporabo direktnih povezav. [31]

Akcija	URL
Ogled uporabnikovega profila	http://www.facebook.com/profile.php?id=XXXXX
Dregljaj uporabnika	http://www.facebook.com/poke.php?id=XXXXX
Pošiljanje sporočila uporabniku	http://www.facebook.com/message.php?id=XXXXX&subject=XXXXX&msg=XXXXX
Dodajanje uporabnika med prijatelje	http://www.facebook.com/addfriend.php?id=XXXXX
Pregled uporabnikovih fotografij	http://www.facebook.com/photos.php?id=XXXXX
Pregled fotografij, na katerih je uporabnik	http://www.facebook.com/photo_search.php?id=XXXXX
Pregled uporabnikovega zidu	http://www.facebook.com/wall.php?id=XXXXX
Pregled uporabnikovih zapiskov	http://www.facebook.com/notes.php?id=XXXXX

Tabela 1 – Seznam direktnih povezav za izvedbo specifičnih akcij.

9. Razvoj prototipa aplikacije

9.1 Ozadje

Eden od priljubljenih načinov promocij in pospeševanja prodaje so nagradne igre. Potekajo lahko v različnih oblikah, vsem pa je skupno, da udeležence navdajajo z evforijo, navdušenostjo in pričakovanjem dobitka. Izvedba je odvisna od osnovnega namena, zakaj podjetje sploh organizira nagradno igro. Na primer, če želi podjetje pospešiti prodajo, potem si potencialne stranke verjetno želi fizično prisotne v svojih prodajnih centrih. Če pa je namen zgolj promocija, utrjevanje blagovne znamke ali seznanjanje potrošnikov s ponudbo, je cilj čim večji obseg omenjanja imena in predstavitve ponudbe.

Pred nekaj leti bi iz časopisa izrezali kuponček, nanj pripisali svoje podatke in ga v prvem primeru osebno odnesli na prodajno mesto, v drugem pa verjetno poslali po pošti. V zadnjem času je čedalje bolj pogosta spletna izvedba nagradnih iger, kjer udeleženci povečujejo možnost dobitkov s priporočili prijateljem. Namesto na kupon svoje podatke vpišejo v spletni obrazec, zraven pripišejo še elektronske naslove vseh svojih prijateljev, ki jih želijo o nagradni igri obvestiti. Več prijateljev vnesejo, večje možnosti pridobitve nagrade imajo.

Tovrstne spletne aplikacije so tipični primer virusnega marketinga, saj se obvestilo o nagradni igri podjetja zelo hitro razširi. Problematično je pogojevanje udeležbe s pošiljanjem obvestil ali povečevanje verjetnosti zadetka s številom priporočil. Na ta način obiskovalci ne vpisujejo le naslovov svojih prijateljev, ki bi jih nagrada morda zanimala. Z namenom izboljšanja lastnih možnosti vnašajo tudi naslove svojih znancev, katerih interesna področja ne pokrivajo sklopa nagradne igre. Posledično so reklamna obvestila prejemnikom lahko moteča in vsiljiva, čemur bi se dobro zasnovana virusna kampanja morala izogibati.

Poleg tega je takšno početje tudi na meji zakonitega, saj 1. odstavek 45a člena Zakona o varstvu potrošnikov (ZVPot) pravi: *»Podjetje lahko uporablja sistem klicev brez posredovanja človeka, faksimile napravo in elektronsko pošto samo z vnaprejšnjim soglasjem posameznega potrošnika, ki mu je sporočilo namenjeno«* [32]. Ker avtomatizem ob pošiljanju reklamnega sporočila za naslov pošiljatelja uporabi naslov udeleženca (ki je ponavadi prijatelj), ljudje tovrstnih sporočil ponavadi ne prijavljajo inšpekcijskim službam, pa tudi pregon je zaradi tega otežen.

Spletne socialne mreže, kjer so uporabniki povezani in medsebojno oddaljeni le miškin klik, so tovrstnim promocijam pisane na kožo. Prednost pred virusnimi promocijami na samostojnih spletnih straneh je v tem, da uporabnikom ni potrebno vnašati elektronskih naslovov. Ponavadi je urejen grafični vmesnik, ki uporabniku ponudi seznam povezanih prijateljev, ki jih uporabnik zgolj odključa. Druga prednost je, da podjetje deluje pod okriljem SSM. Dokler se drži navodil SSM, lahko pošilja obvestila označenim uporabnikom. SSM deluje kot črna škatla in v ozadju na več nivojih avtomatično skrbi za uporabnikovo zaščito zasebnosti.

Mnogo slovenskih podjetij je že spoznalo moč virusnega marketinga in socialnih mrež. Množica podjetij na Facebooku uporabljajo možnosti skupin in predstavitev strani. Zanimivo pa je, da praktično ne uporabljajo izvedbe nagradnih iger. V času pisanja (sredina novembra 2009) na Facebooku nisem našel reklame za niti eno nagradno igro.

9.2 Opis aplikacije

Zaradi zgoraj navedenih ugotovitev sem se odločil, da razvijem prototip aplikacije za izvedbo nagradnih iger. Aplikacija mora podjetjem omogočati izvedbo nagradne igre z upoštevanjem elementov za spodbujanje virusnega marketinga.

Za to potrebujem vnos nagradne igre, seznam nagradnih iger, med katerimi lahko uporabniki izbirajo, način pristopa uporabnika k nagradni igri, sistem izvedbe virusnega marketinga, ki ga bom omejil na Facebook in žreb srečnih dobitnikov.

Aplikacija je v osnovi enostavna in bom edini razvijalec. Osnovni namen je na realnem primeru spoznati Facebook platformo.

9.3 Problemska domena

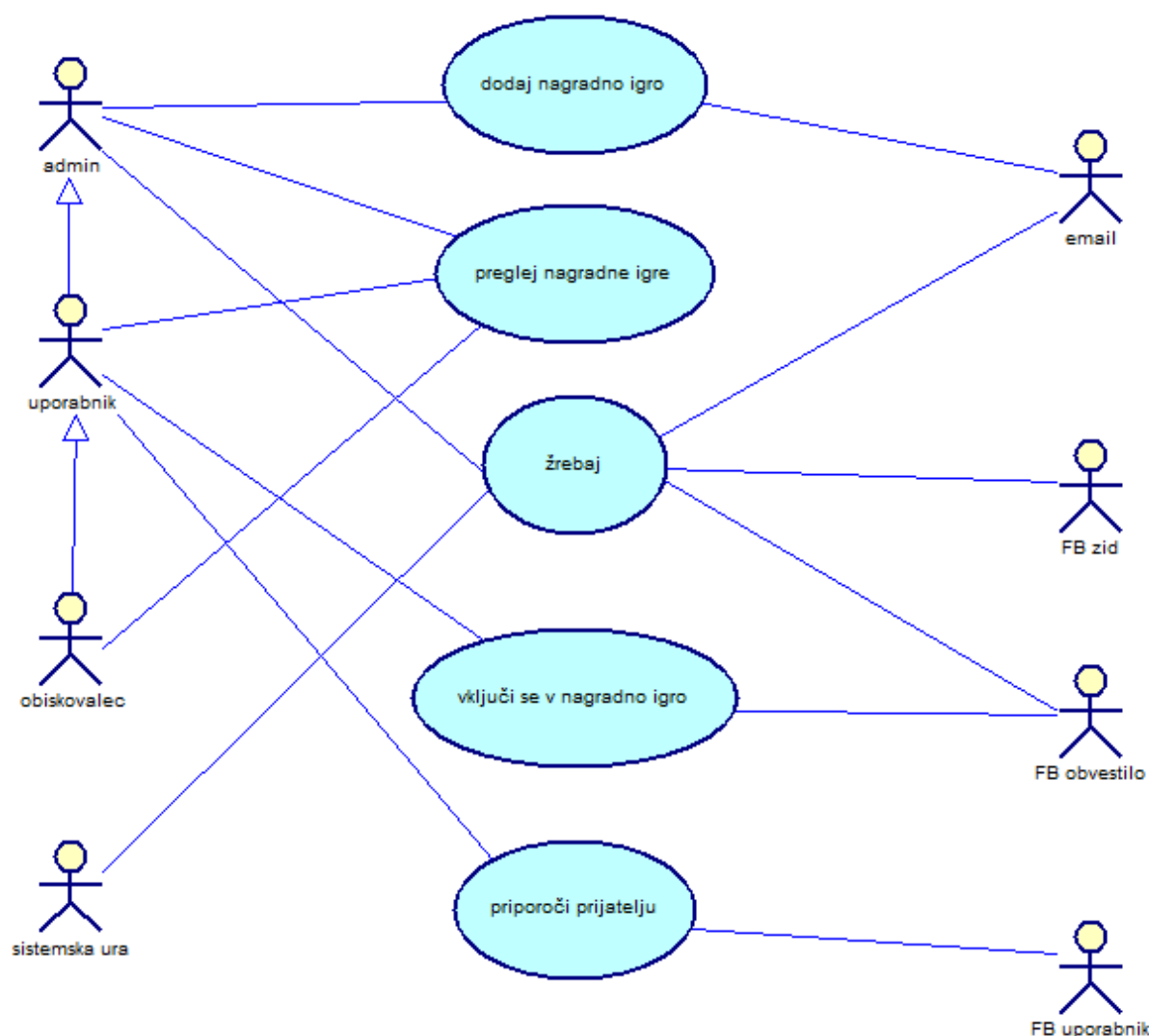
Vsaka nagradna igra bo imela svoj naziv, kratek opis in omejen rok trajanja. Igru se nastavi nagradno vprašanje, na katerega uporabniki odgovarjajo. Vprašanje ima več možnih odgovorov, med katerimi je potrebno označiti pravi. Ker izvajalec aplikacije ne nastopa kot organizator nagradne igre je potrebno za vsako nagradno igro določiti tudi njenega sponzorja. Sponzorju se določi naziv, naslov, kontaktni telefon, elektronsko pošto in spletno stran. Sponzor srečnemu dobitniku nagradne igre podeli nagrado. Nagrad posamezne nagradne igre je lahko več (prva nagrada, druga nagrada ...), vsaka ima svoj naziv in opis.

Uporabniki pregledujejo nagradne igre. Ko izberejo igro in odgovorijo na zastavljeno vprašanje, se jih doda v »boben« za žreb. Takrat ima uporabnik možnost priporočiti igro svojim prijateljem. Žreb se izvede programsko, za vsako nagrado nagradne igre se izžreba eno osebo ter dve dodatni za rezervo.

Čeprav bo prototip v slovenskem jeziku, bo zasnova aplikacije omogočala prevode.

9.4 Primeri uporabe - PU

Za boljšo predstavbo, kako bo aplikacija delovala, podajam diagram in opise primerov uporabe.



Slika 17 - Diagram primerov uporabe.

9.4.1 PU Dodaj nagradno igro

9.4.1.1 Kratak opis

Vnos nagradne igre v sistem.

9.4.1.2 Glavni tok

1. Administrator izbere dodajanje nove nagradne igre.
2. Sistem prikaže vnosno formo s polji:
 - a. Naziv nagradne igre
 - b. Izbira pokrovitelja
 - c. Datum trajanja nagradne igre
 - d. Opis
 - e. Nagradno vprašanje in možni odgovori
 - f. Nazivi in opisi nagrad.
3. Administrator izpolni zahtevane podatke in potrdi vnos.
4. Sistem preveri vnos in shrani nagradno igro.

5. Sistem obvesti pokrovitelja o vnosu nagradne igre preko elektronske pošte.

9.4.1.3 *Alternativni tok*

9.4.1.3.1 Dodaj pokrovitelja

Želenega pokrovitelja ni v naboru pokroviteljev.

Administrator izbere dodajanje novega pokrovitelja.

1. Sistem prikaže vnosno formo za vnos pokrovitelja s polji:
 - a. Naziv
 - b. Naslov
 - c. Telefon in faks
 - d. Email
 - e. Spletna stran.
2. Administrator vnese zahtevane podatke in potrdi vnos.
3. Sistem preveri vnos in shrani pokrovitelja.

9.4.1.4 *Posebne zahteve*

Uporabnik, ki dodaja nagradno igro, mora imeti privilegije administratorja.

9.4.2 **PU Preglej nagradne igre**

9.4.2.1 *Kratek opis*

Pregled seznama nagradnih iger.

9.4.2.2 *Glavni tok*

1. Uporabnik pride na začetno stran aplikacije.
2. Sistem prikaže seznam vseh aktivnih nagradnih iger, v katerih uporabnik še ni sodeloval, ločeno seznam aktivnih nagradnih iger, v katerih uporabnik sodeluje in čaka na žreb, in seznam nagradnih iger v katerih je uporabnik letos srečni dobitnik.

9.4.2.3 *Napotki za razširitev*

Dodaj možnost filtriranja in iskanja nagradne igre.

9.4.3 **PU Žrebaj**

9.4.3.1 *Kratek opis*

Žrebanje izmed nabora uporabnikov, ki so odgovorili na nagradno vprašanje in s tem sodelujejo v nagradni igri.

9.4.3.2 *Glavni tok*

1. Sistemska ura ali administrator zažene skripto za žreb.
2. Sistem izmed vseh sodelujočih uporabnikov naključno izbere srečnega dobitnika.
3. Sistem shrani žreb.
4. Sistem o žrebu obvesti vse sodelujoče preko elektronske pošte.

9.4.3.3 *Posebne zahteve*

Uporabnik, ki dodaja nagradno igro mora imeti privilegije administratorja ali sistemske ure.

9.4.3.4 *Napotki za razširitev*

Izžrebanemu uporabniku je potrebno poslati elektronsko sporočilo o dobitku, na katerega mora odgovoriti in navesti svoje osebne podatke, vključno z davčno številko, ki so pogoj za izdajo nagrade.

9.4.4 **PU Sodeluj**

9.4.4.1 *Kratek opis*

Uporabnik izpolni pogoje za sodelovanje v nagradni igri.

9.4.4.2 *Glavni tok*

1. Uporabnik iz seznama nagradnih iger, v katerih še ne sodeluje, izbere nagradno igro.
2. Sistem prikaže podatke in reklamne informacije o nagradni igri in pokrovitelju.
3. Sistem prikaže vprašanje in nabor možnih odgovorov.
4. Uporabnik iz nabora odgovorov izbere enega in potrdi vnos.
5. Sistem preveri pravilnost odgovora.
6. Sistem shrani uporabnika kot udeleženca nagradne igre.
7. Sistem uporabnika preusmeri na priporočilo prijateljem.

9.4.4.3 *Posebne zahteve*

Uporabnik mora biti prijavljen v sistem in imeti naloženo našo aplikacijo.

9.4.5 **PU Priporoči prijatelju**

9.4.5.1 *Kratek opis*

Uporabnik priporoči nagradno igro svojim prijateljem.

9.4.5.2 *Glavni tok*

1. Uporabnik iz seznama svojih prijateljev izbere, komu želi priporočiti nagradno igro.
2. Uporabnik lahko vnese elektronske naslove prijateljev, ki niso na Facebooku.
3. Uporabnik potrdi vnos.
4. Sistem prikaže potrditveno okno s seznamom vseh izbranih prijateljev in predogledom sporočila, ki ga bodo prijatelji prejeli.

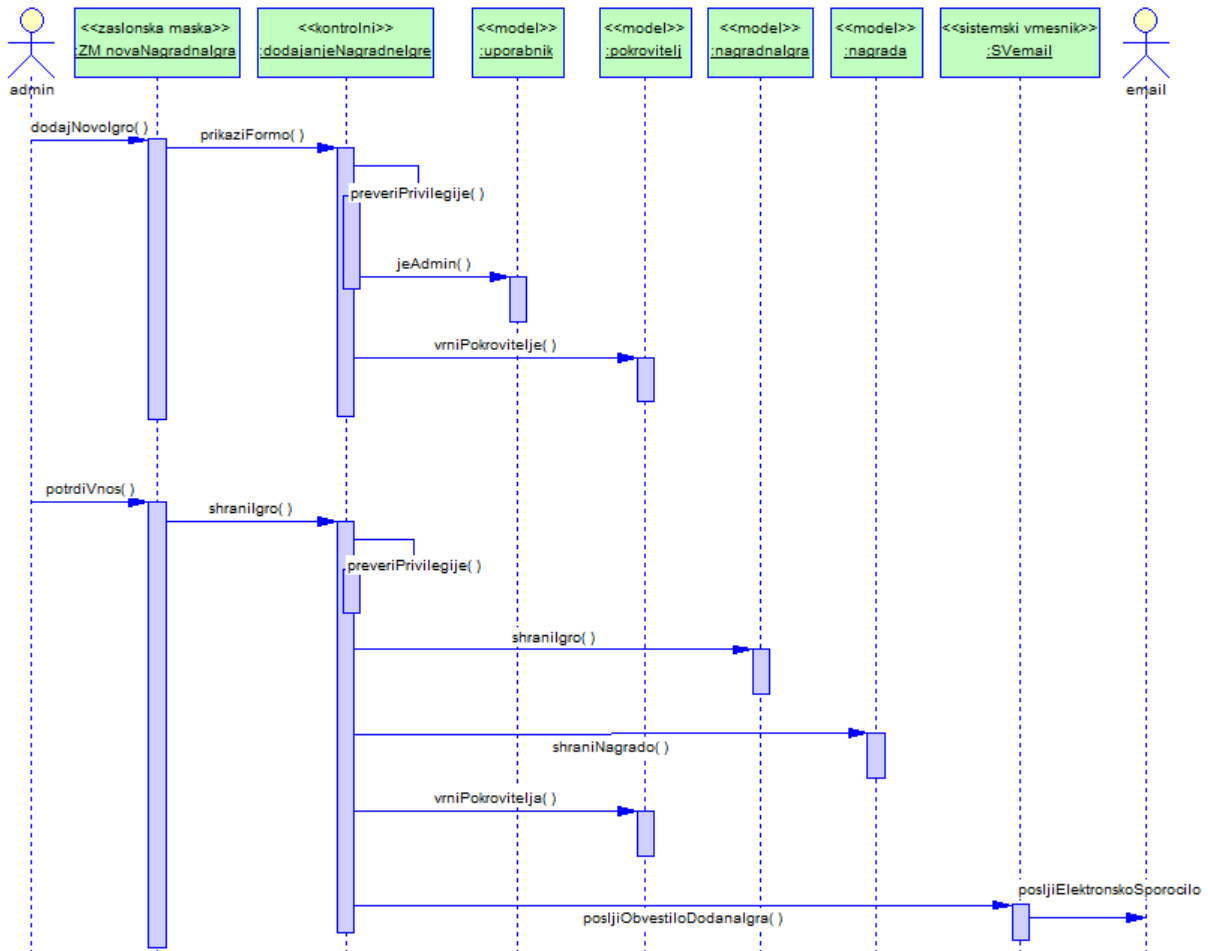
9.4.5.3 *Alternativni tok*

1. Uporabnik pritisne gumb preskoči.
2. Sistem preusmeri uporabnika na seznam nagradnih iger.

9.5 **Diagrami zaporedja**

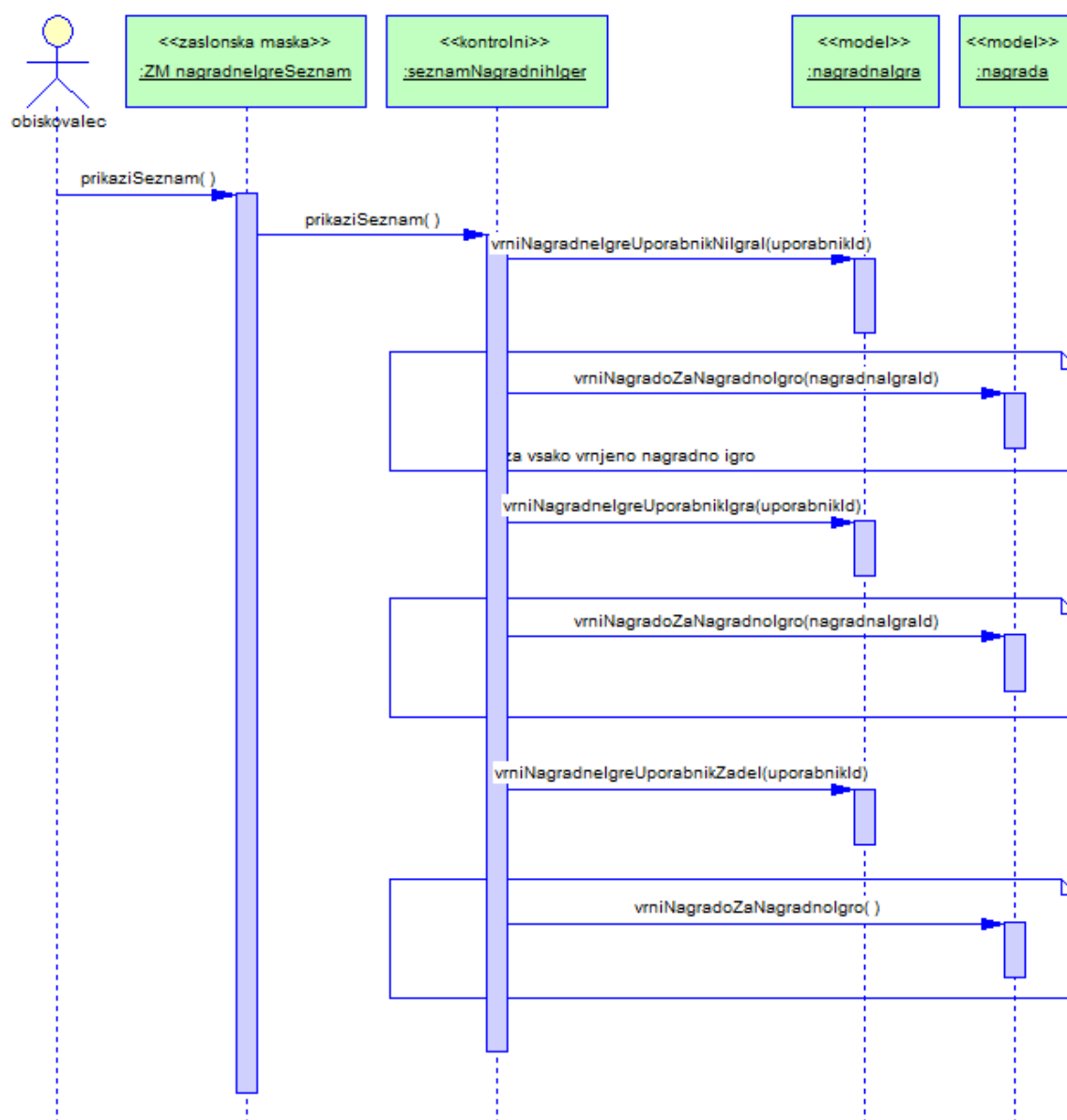
Za lažje razumevanje poteka in kako razredi med seboj komunicirajo sem pripravil diagrame zaporedja.

9.5.1 Dodaj nagradno igro

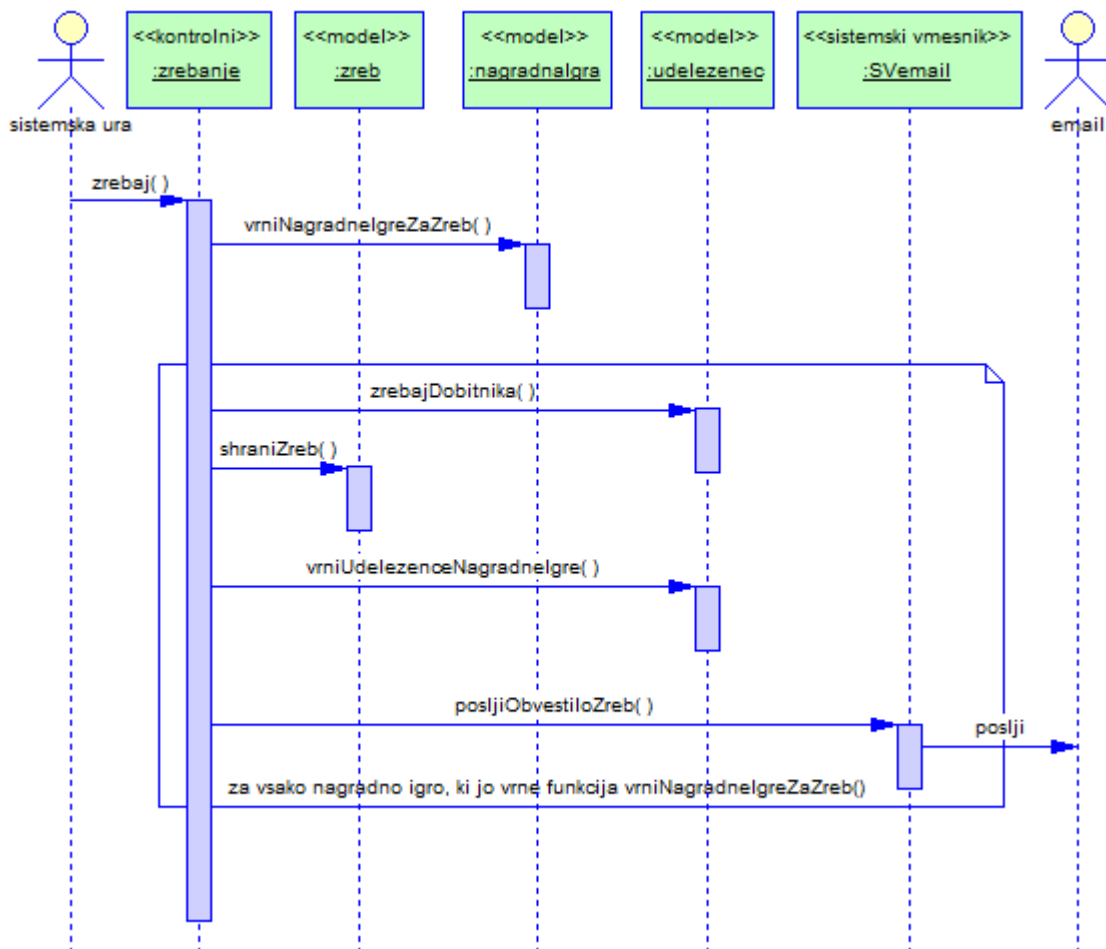


Slika 18 - Diagram zaporedja za primer uporabe *Dodaj nagradno igro*.

9.5.2 Preglej nagradne igre

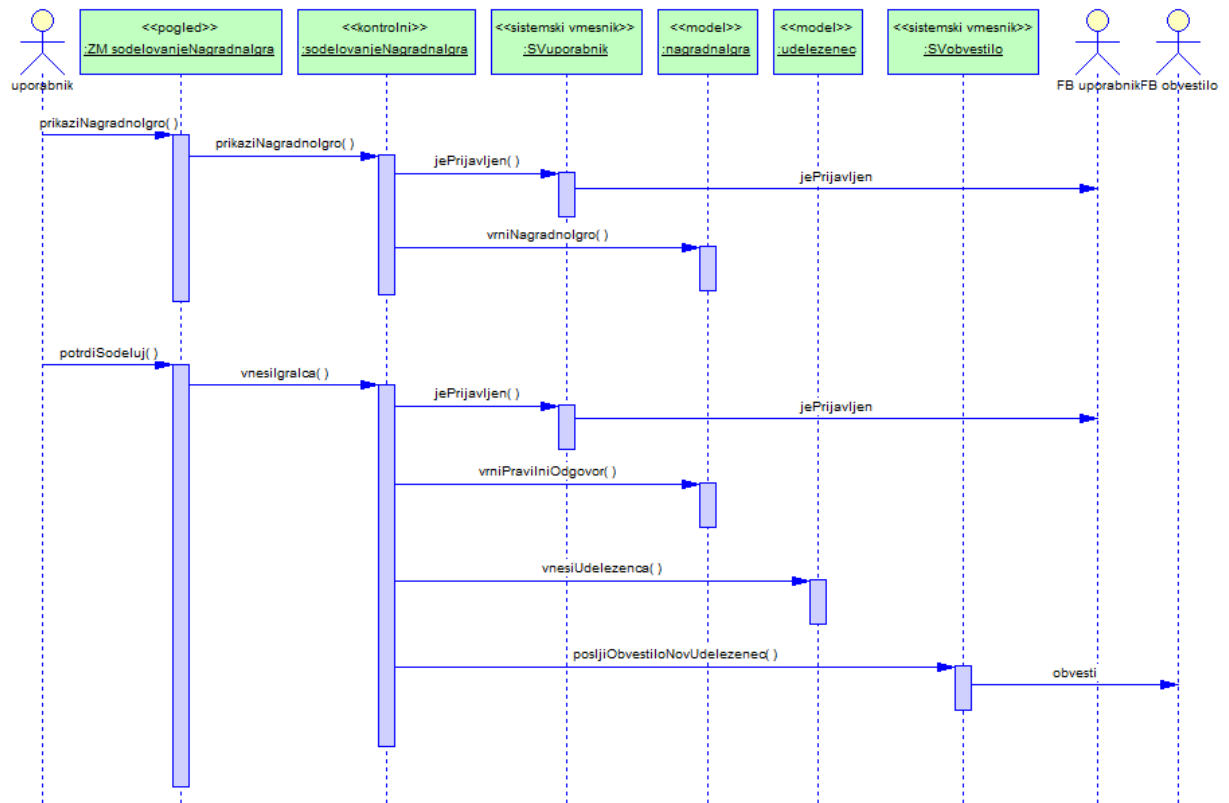
Slika 19 - Diagram zaporedja za primer uporabe *Preglej nagradne igre*.

9.5.3 Žrebaj



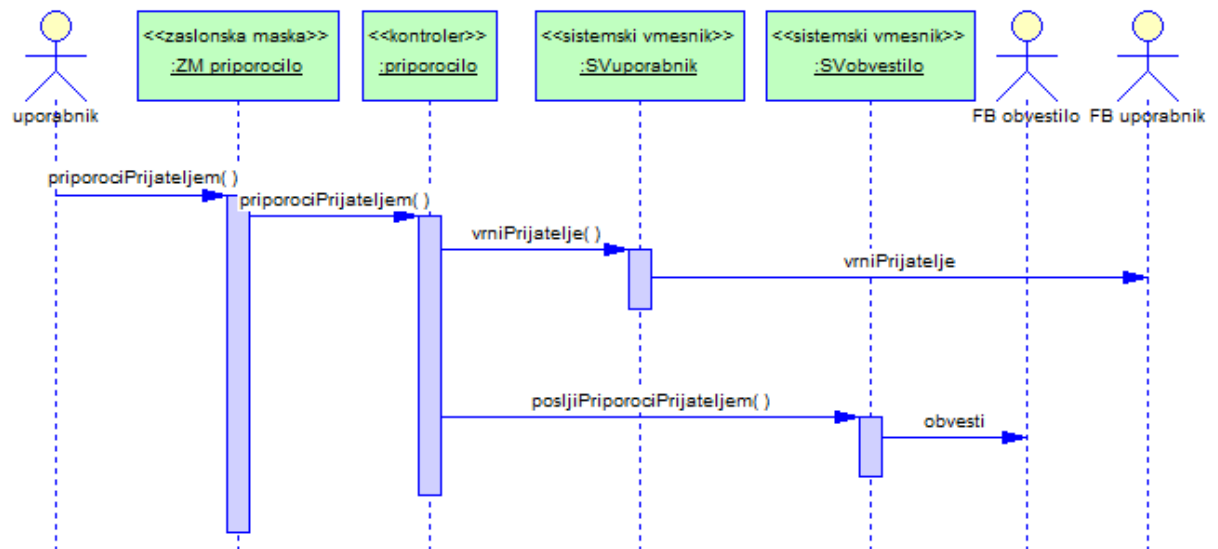
Slika 20 - Diagram zaporedja za primer uporabe Žrebaj.

9.5.4 Sodeluj



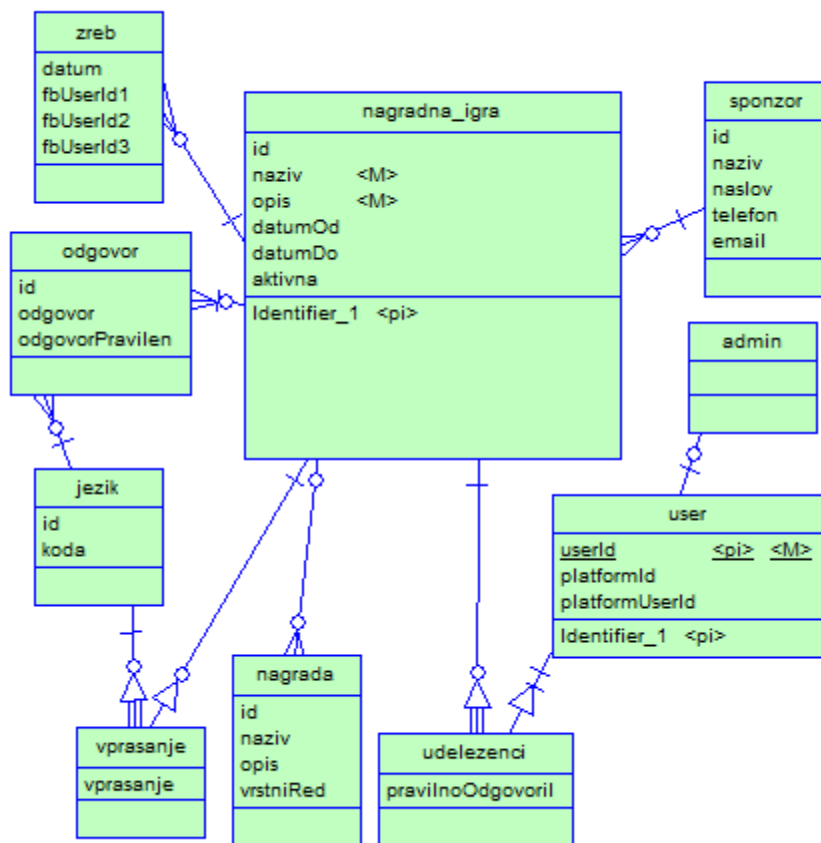
Slika 21 - Diagram zaporedja za primer uporabe *Sodeluj*

9.5.5 Priporoči



Slika 22 - Diagram zaporedja za primer uporabe *Priporoči*.

9.6 Entitetni diagram podatkovne baze



Slika 23 - Entitetni diagram.

10. Arhitektura

Prototip aplikacije bo sicer teklen na Facebook platformi, vendar želim arhitekturo osnovati tako, da bo kasneje možno izvesti aplikacijo tudi na OpenSocial ali na samostojni spletni strani. Torej je potrebno nekako ločiti posamezen dostop do podatkov, predstavitevno in poslovno logiko. Tovrstnim zahtevam ustrežata MVC (*Model-View-Controller*) ali trinivojski arhitekturni koncept.

10.1 MVC

Arhitektura MVC (*Model-View-Controller*) deli aplikacijo na tri nivoje - model, pogled in kontroler.

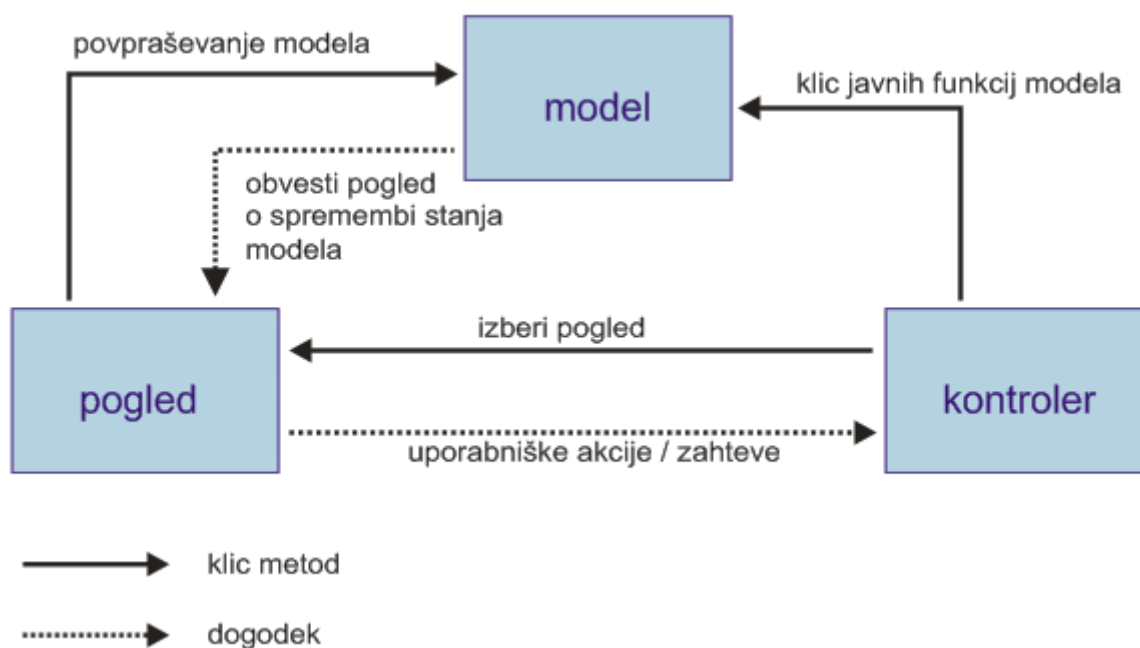
Kontroler koordinira dinamično procesiranje. Pri spletnih aplikacijah na podlagi vsebine GET ali POST določi, katera akcija se bo izvedla.

Model predstavlja podatke s katerimi aplikacija razpolaga. Vsebuje logiko za dostop do podatkov in njihovo manipulacijo. Če želimo na primer shraniti podatek o začetku nagradne igre, model vsebuje tudi logiko, ki preveri, ali datum ustreza pravilom (je veljaven gregorijanski datum, je datum kasnejši kot trenutni ...).

Pogled je prikaz aplikacije, ponavadi v obliki uporabniškega vmesnika, ni pa nujno. Pri spletnih aplikacijah je to navadno HTML.

Potek je sledeči: [33]

- uporabnik sproži dogodek (s pritiskom na gumb, potrditvijo spletnega obrazca ...)
- kontroler prevzame vnos in ga pretvori v primerno akcijo, ki jo razume model
- kontroler obvesti model o akciji uporabnika, kar povzroči spremembo stanja modela
- pogled z izpraševanjem od modela dobi podatke in generira ustrezen uporabniški vmesnik za prikaz. Odvisno od izvedbe lahko izpraševanje sproži kontroler ali ob spremembi stanja avtomatično model.
- uporabniški vmesnik čaka naslednjo uporabniško akcijo, ki ponovi cikel.



Slika 24 - Prikaz arhitekture MVC. [34]

10.2 Trinivojska arhitektura

Trinivojsko arhitekturo natančneje delijo na trinivojsko in tristopenjsko. Razlika med njima je, da pri trinivojski govorimo zlasti z vidika logične porazdelitve, medtem ko pri tristopenjski govorimo o fizični razporeditvi na osnovi strežnikov. [35]

Pojma sta ponavadi združena in govorimo zgolj o trinivojski arhitekturi, ali imamo v mislih fizično ali logično porazdelitev, pa se razbere iz konteksta.

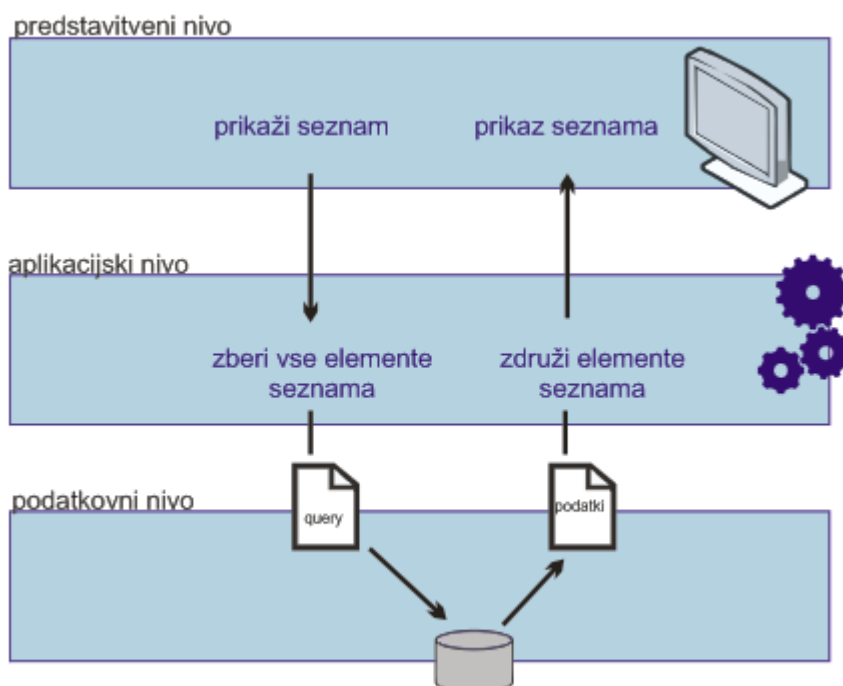
Trinivojska arhitektura deli logiko na predstavitevni nivo, aplikacijsko logiko s poslovnimi pravili in podatkovni nivo.

Predstavitevni nivo je najvišji nivo aplikacije in prikazuje podatke uporabniku ali odjemalcu (odjemalec je seveda lahko tudi druga aplikacija, ki podatke ne dobiva preko grafičnega

vmesnika, temveč preko protokolov, kot je npr. XMLP). Pri spletnih aplikacijah predstaviteni nivo tipično predstavlja spletni strežnik ali del aplikacije, ki odjemalcu posreduje ali generira vsebino v formatu HTML, XML itd. Če je vsebina statična, potrebe po spodnjih nivojih niti ni. Enako velja v primeru v predpomnilniku shranjene dinamično generirane vsebine.

Nivo aplikacijske logike se označuje tudi kot logični nivo ali procesna logika. Vsebuje poslovno logiko in kontrolira funkcionalnost aplikacije. Pri spletnih aplikacijah je to aplikacijski strežnik, kot je na primer PHP, Java EE, ASP.net itd.

Podatkovni nivo sestavljajo podatkovni strežniki, ki podatke ohranjajo nevtralne in neodvisne od aplikacijskih strežnikov in poslovne logike.

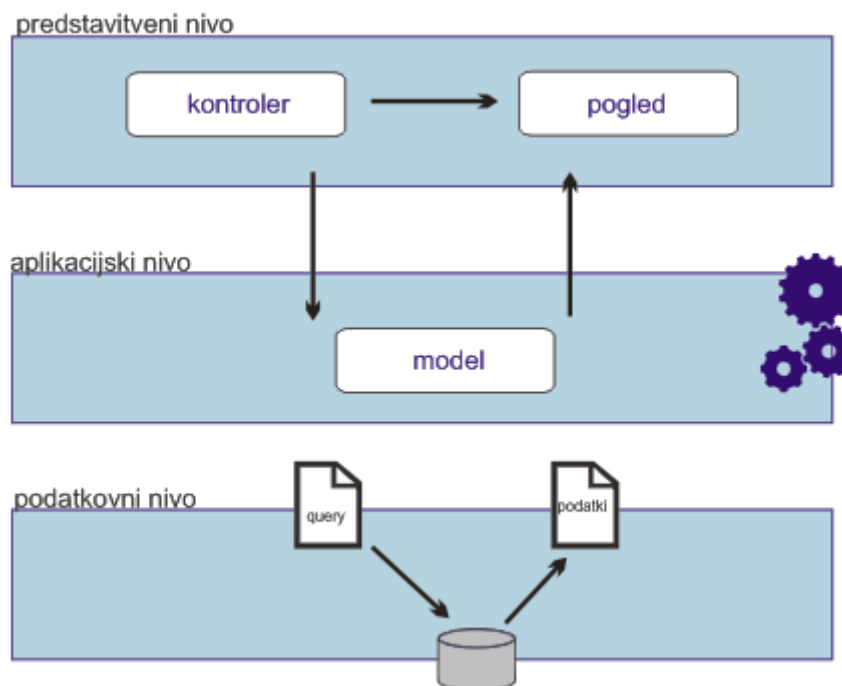


Slika 25 - Prikaz trinivojske arhitekture.

Obe arhitekturi sta si zelo blizu. Med strokovnjaki obstajajo trenja, ali je MVC samostojni arhitekturni koncept ali le implementacija trinivojske.

Zagovorniki samostojnih konceptov izpostavljajo razliko v topologiji. Trinivojska arhitektura je namreč striktno linearna, predstaviteni nivo nikoli ne komunicira direktno s podatkovnim, ampak vedno preko nivoja aplikacijske logike. Pri MVC-ju je stvar drugačna, trikotna. Pogled pošilja sporočila kontrolerju, kontroler modelu in model direktno posodablja pogled. [36]

Drugo mnenje obravnava pogled in kontroler MVC-ja kot del predstavitvene plasti trinivojske arhitekture, modeli MVC-ja pa so del logične plasti. [37]



Slika 26 – Skica arhitekture MVC kot del trinivojske arhitekture. [38]

10.3 Prilagoditev konceptov potrebam

10.3.1 Facebook API

Ko sem pričel s programiranjem lastne aplikacije, so knjige, v katerih sem spoznaval prve korake, priporočale uporabo klasičnega pristopa z uporabo označevalnega jezika FBML [39][40]. Posledično sem za svojo aplikacijo uporabil tovrstni pristop.

10.3.2 Arhitektura

Za arhitekturo sem uporabil linearen pristop. Pogled in model torej komunicirata samo s kontrolerjem in ne med seboj. Od tu naprej bom pod pojmom predstavitveni nivo smatral sklop, kjer so shranjeni pogledi, pod pojmom aplikacijski nivo sklop, kjer se nahajajo kontrolerji, in pod pojmom podatkovni nivo sklop, kjer se nahajajo modeli.

Facebook se s stališča načrtovanja arhitekture od klasičnih spletnih aplikacij razlikuje pravzaprav samo z vidika hranjenja podatkov. Kot omenjeno, lahko podatke pridobivamo preko klicev funkcij vmesnika API ali kodiranjem povpraševanj FQL. In ravno tu se pri pristopu z uporabo označevalnega jezika FBML pojavi prva dilema.

Po definiciji zgoraj opisanih arhitekturnih konceptov naj bi klici vmesnika API spadali v kontroler, saj zaporedje njihov klicev pomeni tudi zaporedje procesiranja. Povpraševanja FQL, s katerimi pridobivamo surove podatke, pa naj bi sodila v modele. Če se striktno držimo teh načel, pa porušimo prvotni razlog uvedbe arhitekturnih konceptov – prenosljivost kode. Modeli in kontrolerji, ki vsebujejo klice specifičnih Facebook funkcij, namreč niso uporabni na OpenSocial platformi ali v kakšnem drugem okolju.

Zato sem poleg omenjenih treh stereotipnih razredov uvedel še sistemske vmesnike, ki se ravno tako kot pogledi razlikujejo od platforme do platforme.

10.3.3 Zajem podatkov

Kljub sistemskim vmesnikom sem njihov obseg zmanjšal na minimum tako, da sem vse potrebne podatke zajel v lokalno podatkovno bazo že ob prvem obisku uporabnika. Ko uporabnik Facebooka doda specifično aplikacijo, Facebook sproži zahtevek na URL povratnega klica ob avtorizaciji, ki jo v konfiguraciji aplikacije navede razvijalec. Na tem URL-ju se lahko pripravi specifična logika, ki se izvede enkratno, ko uporabnik doda aplikacijo Facebook platforme. V tej logiki sem poskrbel za prepis vseh podatkov uporabnika v lokalno bazo.

Takšen postopek je zaradi redundance podatkov seveda rizičen in nepriporočljiv. Poleg tega tovrstno hrambo podatkov prepoveduje tudi politika zasebnosti na Facebooku [41]. Politika zasebnosti v alineji III. predvideva, kateri podatki o uporabniku se lahko trajno shranjujejo.

Parameter	Opis
uid	ID uporabnika
nid	ID primarnega omrežja
eid	ID dogodka
gid	ID skupine
pid	ID fotografije
aid	ID foto albuma
flid	ID seznama prijateljev
listing_id	ID seznama tržnice
page_id	Facebook ID strani
proxied_email	Posredovalni naslov elektronske pošte
notes_count	Skupno število zapiskov uporabnika
profile_update_time	Čas, ko je uporabnik zadnjič posodobil svoj profil

Tabela 2 - Seznam podatkov o uporabniku, ki jih smemo hraniti neomejeno.

K sreči je tip moje aplikacije takšen, da razen unikatne identifikacijske številke uporabnika (*uid*) drugih podatkov o uporabniku ne potrebujem in do nekonsistentnosti in kršenja politike zasebnosti posledično ne prihaja.

10.4 Izbira programskega jezika

Kot že nekajkrat omenjeno, Facebook omogoča programski zajem in manipulacijo podatkov preko aplikacijskega programskega vmesnika. Vmesnik, ki temelji na REST formatu, omogoča klicanje metod preko HTTP GET in POST zahtev. Za komuniciranje preko HTTP protokola je možno uporabiti praktično katerikoli programski jezik, ki zna obravnavati HTTP zahteve. Facebook zagotavlja uradno podporo naslednjim knjižnicam [42]:

- PHP 5
- JavaScript
- Facebook Connect za iPhone
- ActionScript 3.0 (podprt s strani Adobe)
- Microsoft SDK (podprt s strani Microsofta)

- Force.com (podpira jo salesforce.com)

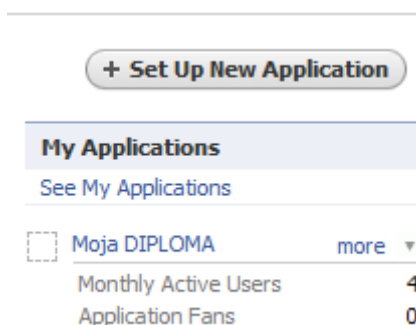
.NET knjižnico podpira organizacija SocialCash.com. Poleg omenjenih je na voljo mnogo drugih knjižnic (vključno s knjižnico za ASP.NET, C++, C#, Javo, Perl, Python, itd.), ki jih podpira skupnost Facebook platforme.

Ker je PHP prvotni jezik, v katerem je jedro Facebooka napisano, je uradno podprt, predvsem pa že dalj časa v njem programiram, je bila odločitev zanj samoumevna. Za podatkovno bazo sem izbral MySQL, ki je brezplačna in dobro integrirana s PHP-jem.

11. Razvoj Facebook aplikacije v praksi

Razvijalec aplikacij za Facebook platformo se mora najprej registrirati na Facebooku. Po registraciji je treba znotraj Facebooka namestiti aplikacijo za razvijalce, ki omogoča razvoj in upravljanje nastavitev aplikacij za Facebook platformo. Namesti se jo na naslovu <http://www.facebook.com/developers>.

Po namestitvi se nam prikaže stran, na kateri je v osrednjem delu prikazan blog glavnih razvijalcev, na desni strani zgoraj kontrolna plošča naših aplikacij, pod njo pa status delovanja platforme.



Slika 27 - Desno zgoraj se nahaja kontrolna plošča s seznamom naših aplikacij.

Kliknemo na gumb za nastavitev nove aplikacije. V naslednji zaslonski maski vnesemo zeleno ime aplikacije in potrdimo naše strinjanje s splošnimi pogoji.

The screenshot shows the 'Essential Information' form for creating a new application. It includes a text input field for 'Application Name', a 'Terms' section with the question 'Do you agree to the Facebook Terms?' and radio buttons for 'Agree' and 'Disagree' (where 'Disagree' is selected), and a 'Create Application' button. A note on the right states: 'Cannot contain Facebook trademarks or have a name that can be confused with an application built by Facebook.'

Slika 28 - Vzpostavljanje nove aplikacije.

Po vnosu je aplikacija znotraj Facebooka kreirana, potrebno je nastaviti še nastavitve in aplikacijo sprogramirati.

The screenshot shows the Facebook application dashboard for 'Moja DIPLOMA'. At the top, it indicates the 'Directory Status: Not Submitted' and provides a link to submit the application. Below this, there are three key metrics: Monthly Active Users (0), Application Fans (0), and Total Users (6). The dashboard is divided into two columns of settings and links. The left column includes 'Restrictions' (Sandbox Mode), 'API Key' (f0362740704356d29411cae8c381820c), 'Application Secret' (1cfe474664b738a9cdd43f74dedd41), and 'Application ID' (156727819860). The right column includes links for 'Create Feed Template', 'DataStoreAdmin', 'Edit Application Profile', 'Edit Settings', 'Reset Secret Key', 'Statistics', 'Translations', and 'View Application Profile'.

Slika 29 - Nadzorna plošča aplikacije.

11.1 Nastavitve

Ko ustvarimo aplikacijo, se je potrebno prebiti še skozi njene nastavitve opisane spodaj. Nastavljamo samo tiste, ki so pomembne za našo aplikacijo.

11.1.1 Osnova (*Basic*)

Nujne informacije (<i>Essential Information</i>)	
Ime aplikacije	Naziv aplikacije.
ID aplikacije	Unikatna številka aplikacije.
API ključ	Ključ, ki ga potrebujemo za klicanje aplikacijskega programskega vmesnika.
Skrivnost	Skrivnostni niz znakov, ki omogoča dodatno zaščito proti poneverjanju.
Osnovne informacije (<i>Basic Information</i>)	
Opis	Tekst z opisom aplikacije. Prikazan je na strani aplikacije.
Ikona	Ikona aplikacije velikosti 16x16 pikslov, prikazana na različnih mestih, med drugimv seznamu priljubljenih aplikacij in seznamu nameščenih aplikacij
Logo	Logo velikosti 75x75 pikslov, tipično prikazan v zgornjem levem kotu na strani aplikacije.
Jezik	Primarni jezik aplikacije.
Razvijalci	Seznam razvijalcev. Razvijalci lahko urejajo nastavitve aplikacije, jo testirajo in pregledujejo v peskovnik načinu (<i>sandbox mode</i>), njihova imena pa so prikazana tudi na strani profila aplikacije.

	Vnos je možen z izborom s seznama prijateljev v tekstovnem polju na način tipkanje v naprej (<i>typeahead</i>).
Kontaktne informacije (<i>Contact Information</i>)	
Kontaktna elektronska pošta razvijalca	Kontaktni naslov s katerim lahko Facebook pride v stik z razvijalcem.
Naslov uporabniške podpore	Naslov na katerega se lahko obrnejo uporabniki. Vnesemo lahko elektronsko pošto ali naslov spletne strani.
URL-ji za uporabnike (<i>User-Facing URLs</i>)	
URL za priljubljene	Naslov, kamor se preusmeri uporabnika, ko klikne na zaznamek priljubljenih aplikacij. Če pustimo prazno, je obnašanje privzeto in se uporabnika preusmeri na naslov Platna ali Povezave.
URL pomoči	Absolutna povezava, na kateri se nahaja pomoč uporabnikom.
URL nastavitve zasebnosti	Absolutna povezava, kjer se nahaja stran z nastavitvami zasebnosti.
URL pogojev uporabe aplikacije	Absolutna povezava do pogojev uporabe aplikacije.

Tabela 3 - Nastavitve: Osnova

11.1.2 Avtorizacija namestitve (*Authentication*)

Nastavitve avtorizacije (<i>Authentication Settings</i>)	
Možnost nameščanja na ...	Na katere tipe Facebook profilov se lahko aplikacija namesti. Izbiramo med Uporabniškimi profili in Facebook stranmi, lahko dovolimo namestitev obema tipoma.
Mehanizem povratnega klika (<i>Authentication Callback URLs</i>)	
URL povratnega klika po avtorizaciji	Ko uporabnik doda aplikacijo, se izvrši klic navedenega URL naslova. Priročno, če želimo hraniti kakšne podatke o uporabniku v lokalni bazi. Poleg tega je nujno, če želimo v vsakem trenutku vedeti, koliko uporabnikov imamo. Facebook namreč ponuja API klic, ki vrača število uporabnikov, ki je aplikacijo namestilo. Vendar to število vključuje tudi tiste, ki so aplikacijo že odstranili. URL ne sme biti naslov znotraj Facebooka.
URL povratnega klika ob odstranitvi	Ko uporabnik odstrani aplikacijo, se naredi klic na navedeni URL naslov. Ob klicu lahko logika poskrbi za odstranitev podatkov o uporabniku, vodenje statistike itd. URL ne sme biti naslov znotraj Facebooka.

Tabela 4 - Nastavitve: Avtorizacija namestitve

11.1.3 Profili (*Profiles*)

Zavihek profila (*Profile tab*)

Naziv zavihka	Naziv zavihka aplikacije na profilu.
URL zavihka	Facebook pobere vsebino zavihka z navedenega URL naslova, ki mora biti relativen glede na stran platna.
Škatla profila (<i>Profile Box</i>)	
Privzeta postavitev	V kateri stolpec se škatla profila prvotno postavi. Uporabnik lahko kasneje škatlo poljubno premika po svojem profilu. Izbiramo med dvema možnostma: širok in ozek.
Info področje (<i>Info Section</i>)	
URL povratnega klica ob posodobitvi informacij	Ko uporabnik posodobi informacije v svojem profilu, se naredi klic navedene povezave.
Založnik profila (<i>Profile Publisher</i>)	
Objavi tekst	Oznaka za povezavo založnika, ko se ustvari vsebina na prijateljevem profilu.
URL povratnega klica ob objavi	Facebook uporabi vsebino za založnika s tega URL naslova.
Samoobjava teksta	Oznaka za Založnika, ko se ustvari vsebina na uporabnikovem lastnem profilu.
URL povratnega klica ob samo-objavi	Facebook uporabi vsebino za založnika s tega URL naslova.
Facebook administracija (<i>Info Section</i>)	
URL urejanja	Povezava, na kateri lahko administratorji urejajo nastavitve aplikacije.

11.1.4 Platno (*Canvas*)

Obvezni URL-ji (<i>Required URLs</i>)	
URL platna	Osnovni URL naslov, na katerem se nahaja aplikacija v formatu <code>http://apps.facebook.com/XXXXXXXXX/</code> .
URL povratnega klica za platno	Facebook pobere vsebino platna iz navedenega URL naslova.
Neobvezni URLji (<i>Optional URLs</i>)	
URL preusmeritve po avtorizaciji	URL naslov, kamor se preusmeri uporabnika, ko prvič naloži aplikacijo. Privzeta vrednost je enaka URL-ju platna.
Nastavitve platna (<i>Canvas settings</i>)	
Način prikazovanja	Ali se stran platna prikazuje kot FBML ali IFrame. Pri IFrame se lahko uporablja XFBML.
Velikost IFrame	Izbiramo lahko med opcijama pametne velikosti in možnostjo spreminjanja velikosti. Pametna velikost, ki je privzeta, avtomatično prilagodi IFrame tako, da zavzame celoten prostor, namenjen platnu. Druga opcija dovoljuje aplikaciji kontrolo nad velikostjo IFramea.
Velikost platna	V preteklosti je Facebook razvijalcem za njihove aplikacije ponujal platno v širini 646 pikslov, sedaj je na voljo 760 pikslov. Opcija je le za vzdrževanje združljivosti. Privzeta nastavitve je 760 pikslov.
Hitre spremembe	Pohitri spremembe na platnu tako, da preskoči nalaganje Facebookovega IFramea. Privzeto je ta nastavitve izključena.

Tabela 5 - Nastavitve: Platno

11.1.5 Povezava (*Connect*)

Nastavitve Facebook povezave (<i>Facebook Connect Settings</i>)	
URL povezave	Osrednji URL spletne strani za povezavo.
Logo povezave	Slika, ki se prikaže v dialogu za vzpostavitev Facebook povezave s stranjo. Omejitev je 99 pikslov širine in 22 pikslov višine.
URL predogleda	Ob zahtevku Facebook povezave se iz tega naslova črpa vsebina za prikaz.
Osnovna domena	Omogoča implementacijo Facebook povezave preko več poddomen (npr. uporaba webeks.net bi omogočila tudi fb.webeks.net, www.webeks.net, nekaj.webeks.net ...)
URL za pridobitev uporabniškega računa	Uporabnik se z uporabo Facebook povezave na spletni strani tretje osebe predstavlja z uporabniškim imenom, ki ga uporablja na Facebooku. Če se v določenem trenutku odloči izbrisati svoj račun pri Facebooku, se ga usmeri na navedeni URL, da lahko ohrani račun na spletni strani tretjih oseb.
Povezave s prijatelji (<i>Friend Linking</i>)	
Dostop do povezave s prijatelji	<p>Izbira med možnostma »zahtevano« in »ni zahtevano«. Z označeno možnostjo zahtevano naredimo šele korak k prijavi za to možnost, ki jo mora ekipa Facebooka še potrditi.</p> <p>Možnost omogoča, da se uporabnikom predlaga povezava z njihovimi znanci.</p> <p>Namišljen primer: Moja spletna stran webeks.net ima svojo registracijo uporabnikov. Ana in Janez sta prijatelja na Facebooku in oba sta se prijavila na moji spletni strani. Ana se odloči, da si želi povezati svoj račun prek Facebook povezave. Uporaba API klika <code>connect.registerUsers</code> bo predlagala Ani povezavo z Janezom.</p> <p>Na ta način se socialni graf na spletni strani razvija hitreje.</p>
Kako stran uporablja povezavo s prijatelji	Kot omenjeno, je možnost dosegljiva le na zahtevo, ki jo mora potrditi ekipa Facebooka. Ekipi je potrebno razložiti, zakaj in kako se bo na spletni strani uporabljala Facebook povezava. Poleg tega je potrebno na spletni strani imeti implementacijo lastnega sistema povezovanj s prijatelji.
Lastno povezovanje prijateljev	Izbiranje opcij da ali ne. Uporabiti je potrebno da, če spletna stran uporablja sistem uvažanja kontaktov, ali kakšno drugo možnost iskanja uporabnikov.
Dokumentacija lastnega povezovanja prijateljev	Vnosno polje za razlago, kako deluje sistem uvažanja kontaktov, ali URL povezavo do dokumentacije.
Paketi predlog (<i>Template Bundles</i>)	
Že oddane predloge	Seznam oddanih predlog za uporabo pri

avtomatični objavo enovrstičnih novičk na Facebooku.
--

Tabela 6 - Nastavitve: Povezava

11.1.6 Gradniki (*Widgets*)

Administracija gradnikov (<i>Widgets Administration</i>)	
Razvijalci aplikacije	Izpisani seznam razvijalcev aplikacije, ki se jih doda pod zavihkom Osnove > Razvijalci.
Administratorji	Administratorji lahko urejajo nastavitve gradnikov. Vsi razvijalci imajo privilegije administratorjev.
Moderatorji	Moderatorji gradnikov lahko urejajo vsebino, črno in belo listo.
Naročniki obvestil	Možnost odključati prijavo. Naročniki prejmejo obvestilo, ko se doda nova vsebina.
Globalne nastavitve za komentarje	
Način dovoljenj	Možna izbira med načinom črna lista (<i>blacklist</i>) in bela lista (<i>whitelist</i>). Pri prvem načinu imajo dostop omogočen vsi, ki niso navedeni na seznamu, pri drugem imajo dostop samo navedeni uporabniki.
Bela lista	Možnost vnašanja uporabnikov iz nabora prijateljev.
Črna lista	Seznam prijateljev.
Dovoli anonimne komentarje	Ali dovoljujemo uporabnikom brez Facebook računa pisanje komentarjev z uporabo svojega imena in elektronskega naslova. Možna izbira da ali ne.

Tabela 7 - Nastavitve: Gradniki

11.1.7 Napredno (*Advanced*)

Napredne nastavitve (<i>Advanced Settings</i>)	
Tip aplikacije	Izbira med spletno in namizno aplikacijo.
Peskovnik način	Omogočeno ali onemogočeno. V omogočenem načinu lahko aplikacijo vidijo in uporabljajo le razvijalci, za običajne uporabnike pa ne obstaja.
Mobilna integracija (<i>Mobile Integration</i>)	
SMS integracija	Omogoča integracijo s SMS. Izbira med omogočeno in onemogočeno.
iPhone aplikacijski ID	Aplikacije za iPhone, ki uporabljajo Connect
Priponke (<i>Attachments</i>)	
Tekst za priponke	Oznaka za povezavo do dodajanja priponk. Recimo »Priprnite fotografije«, »Priprnite datoteke« ...
URL povratnega klica za priponke	Facebook dobi vsebino vmesnika za dodajanje priponk preko klica tega naslova.
Varnost (<i>Security</i>)	
Bela lista strežnikov	Facebook blokira dostop vsem, ki uporabljajo naš API ključ in ne prihajajo s tu navedenih IP

	naslovov. Lista je seznam IP-jev, ločenih z vejico.
Skrivnost seje je izjema beli listi	API ključ zadošča za večino aplikacij, ki so zgrajene za Facebook platformo. Aplikacije, ki uporabljajo JavaScript ali Flash za API klice, izpostavljajo ključ za dostop do vmesnika API. Zaradi varnosti se ne priporoča vključevanje tega ključa v kodo. Namesto ključa se uporablja skrivnost seje, katere veljavnost je omejena s trajanjem seje.
Pravno (Legal)	
Video / Najem	Če stran objavlja novice, ki vključujejo prodajo filmov (videa), najemov ali naročil, potem se moramo strinjati z Zakonom o varstvu videa in zasebnosti (<i>Video Privacy Protection Act</i>) in predhodno pridobiti izrecno soglasje (po opt-in principu) pred izmenjavo uporabniških podatkov.
Oglaševanje (Advertising)	
Račun za oglaševanje	Email, s katerim smo se prijavili za račun na Facebooku.

Tabela 8 - Nastavitve: Napredno

11.2 Inicializacija Facebook vmesnika

11.2.1 Nastavitve strežnika

Za gostovanje aplikacije moramo zagotoviti lasten strežnik, kjer so shranjene vse datoteke. Za potrebe gostovanja svojega bloga in različnih testnih spletnih strani že nekaj časa uporabljam spletno gostovanje ponudnika DreamHost iz ZDA. Za deset dolarjev mesečno mi ponujajo neomejeno prostora na podatkovnem disku, gostovanje poljubnega števila domen in neomejen prenos vsebin.

Ugodna cena ima seveda tudi svoje posledice, to je relativno počasna odzivnost strežnika, ki si ga deli več uporabnikov.

Prednost vztrajanja na DreamHostu je poleg cene tudi velikost podjetja, ki ponuja različne tipe gostovanj vključno z najemom namenskih strežnikov. Sprememba zakupljenega paketa gostovanja iz deljenega na namenski strežnik se lahko izvede v nekaj minutah, prenos datotek pa poteka avtomatično. Hitrost povečanja zmogljivosti je v okolju virusnega marketinga, kjer gre lahko za eksponentno rast števila uporabnikov izjemnega pomena.

Na katerem naslovu ali domeni je aplikacija nameščena nastavimo v administraciji nastavitve znotraj Facebooka.

11.3 Uradne knjižnice za programski jezik PHP

Iz nabora ponujenih aplikacijskih programskih vmesnikov sem, kot že omenjeno in razloženo, izbral vmesnik napisan v programskem jeziku PHP. Vmesnik API je potrebno prenesti s Facebooka in ga naložiti na lasten strežnik, kjer bo nameščena aplikacija.

V knjižnici vmesnika API za programski jezik PHP sta dva direktorija - *php* in *footprint*. V *php* direktoriju so datoteke ovojnice *JSON* in datoteka *facebookapi_php5_restlib.php*, datoteka *facebook_desktop.php*, datoteka *facebook_mobile.php* in *facebook.php*. Imena datotek so zgovorna sama po sebi. V direktoriju *footprint* se nahaja nekaj datotek s kodo, ki služi kot primer.

11.3.1 Inicializacija

Za izvedbo spletne aplikacije je pomembna datoteka *facebook.php*, ki vsebuje razred z metodami, potrebnimi za povezavo naše aplikacije s Facebookom. Za inicializacijo moramo znotraj naše aplikacije vključiti datoteko:

```
require_once 'fb_lib/facebook.php';
```

Nato je potrebno ustvariti nov objekt vmesnika in mu kot parametre podati ključ vmesnika API in skrivnost aplikacije.

```
$appapikey = 'f0362740704356d29411cae8c381820c';
$appsecret = '1cfe474664b738a9cdd43f74dedd41';
$facebook = new Facebook($appapikey, $appsecret);
```

S tem je vmesnik API v aplikaciji ustrezno inicializiran in nam omogoča komunikacijo s strežniki Facebooka preko klica svojih funkcij.

11.4 Preverjanje uporabnika

V določenih primerih želim preveriti ali je uporabnik prijavljen v sistem. V drugih želim preverjati pogoj, da je uporabnik administrator.

11.4.1 Preverjanje prijavljenosti v sistem

Za preverjanje, ali je uporabnik prijavljen v sistem, obstaja več načinov. Prvi način je preverjanje na Facebook strežniku z oznakami označevalnega jezika FBML. Uporabimo lahko oznako *Fb:if-is-app-user*, ki vsebino znotraj oznake prikaže le, če je uporabnik naložil aplikacijo in se strinja s pogoji uporabe.

```
<fb:if-is-user uid="51000">Zaščitena vsebina!</fb:if-is-user>
```

Druga možnost je klic funkcije *require_login()*, ki v primeru da je uporabnik prijavljen vrne identifikacijsko številko uporabnika. Če uporabnik ni registriran, ga preusmeri na stran za prijavo uporabnika. Funkcija ni dokumentirana, odkril sem jo slučajno ob pregledovanju obstoječih primerov kode.

Funkcija *require_login()* ne preverja, ali ima uporabnik nameščeno aplikacijo, ampak samo ali je prijavljen v Facebook. Za preverjanje nameščene aplikacije uporabimo klic funkcije *users_isAppUser()* vmesnika API.

Sam sem uporabil drugo možnost, možnost klicev funkcij vmesnika API. Funkcije sem, kot sem že razložil in kot je razvidno iz diagramov zaporedja, vključil v funkcijo *jePrijavljen()* razreda sistemskega vmesnika *SVuporabnik*.

11.4.2 Preverjanje, ali je uporabnik administrator

Pokrovitelje in nagradne igre lahko vnašajo le administratorji. Podatke o uporabnikih, ki so administratorji, hranim v lokalni podatkovni bazi v tabeli *admin*. Za preverjanje posameznega uporabnika imam pripravljeno funkcijo *jeAdmin()* v sistemskem vmesniku *SVuporabnik*.

11.5 Dovoljenja aplikacije

Za izvedbo določenih akcij mora aplikacija od uporabnika pridobiti dovoljenje. To se stori z uporabo oznake *Fb:prompt-permission*, ki prikaže povezavo. Ko uporabnik klikne nanjo, se odpre pogovorno okno, v katerem se uporabnika sprašuje za zahtevana dovoljenja.

Katera dovoljenja zahtevamo od uporabnika, navedemo v parametru oznake *perms*. Vrednost parametra podamo kot seznam z vejico ločenih vrednosti. [43]

Akcije, ki jih ne moremo izvesti brez dovoljenja uporabnika:

- pošiljanje elektronske pošte (vrednosti parametra *perms* je *email*);
- branje objav uporabnika (vrednost *read_stream*);
- objavljane sporočil v založniku uporabnika (vrednost *publish_stream*);
- ali uporabnik dovoljuje nepovezan dostop do svojih podatkov (lokalno lahko hranimo njegove osebne podatke, ki niso predvideni za trajno shranjevanje) (vrednost *offline_access*);
- posodabljanje uporabniškega statusa (vrednost *status_update*);
- nalaganje in označevanje fotografij (vrednost *photo_upload*);
- prijava in urejanje dogodkov (vrednost *create_event*);
- nastavljanje statusa udeležbe dogodka (vrednost *rsp_event*);
- pošiljanje sporočil SMS uporabniku (vrednost *sms*);
- nalaganje video posnetkov na uporabnikov profil (vrednost *video_upload*);
- pisanje, urejanje in brisanje zapisov v uporabniškem profilu (vrednost *create_note*).

Preverjanje, če je uporabnik aplikaciji dodelil dovoljenja, lahko izvedemo preko klica funkcije *users_hasAppPermission()*. Funkciji kot parameter *ext_perm* podamo vrednost, ki označuje katero dovoljenje preverjamo.

11.5.1 Primer zapisa kode za preverjanje dovoljenj

```
$dovolEmail= $facebook->
    api_client->users_hasAppPermission("email");
$dovolPublish = $facebook->
    api_client->users_hasAppPermission("publish_stream");

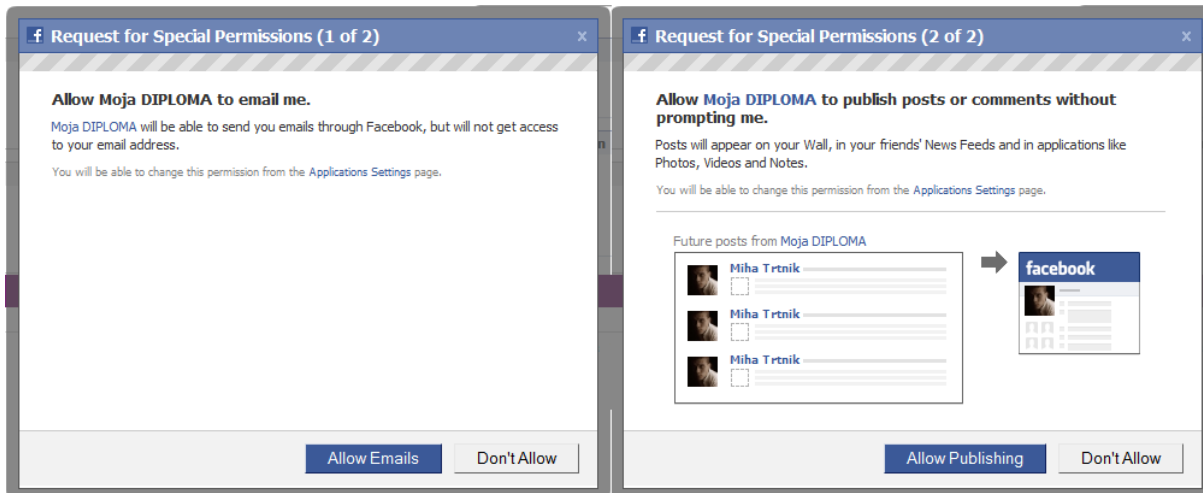
if(!$imaDovoljenjeEmail || !$imaDovoljenjePublish)
{
    echo ' <fb:prompt-permission perms="email, publish_stream">
        Za pravilno izvedbo nagradnih iger potrebujemo tvoje
        dovoljenje!
    </fb:prompt-permission>';
} else {
```

```

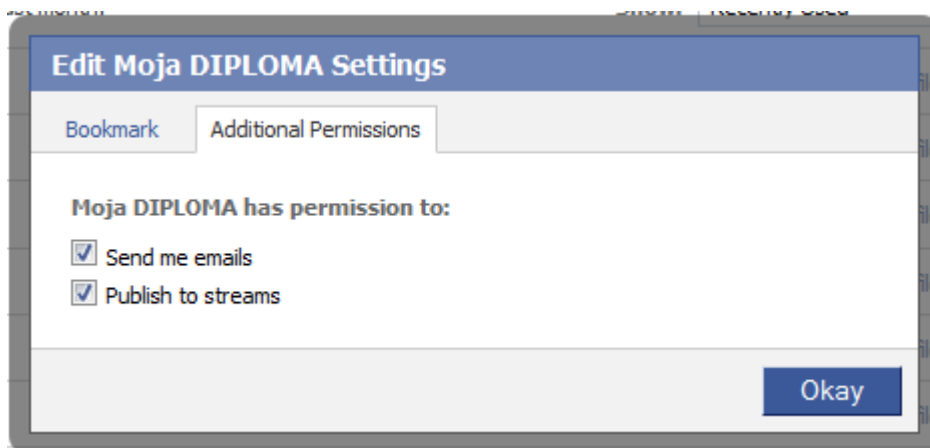
    //show application screen
}

```

11.5.2 Zaslonska maska



Slika 30 - Zaslonska maska - Na levi zahteva za dovoljenje pošiljanje elektronske pošte, na desni zahteva dovoljenja za objavo obvestil.



Slika 31 - Uporabnik lahko dovoljenja kadarkoli nastavi (ukine) pod nastavitvami dovoljenj posamezne aplikacije.

11.6 Izvedba primera uporabe Dodaj nagradno igro

Na zaslonski maski pri dodajanju nagradne igre sem želel prikazati vnosna polja naziv, opis, datum pričetka in datum zaključka nagradne igre, izbiro pokrovitelja oz. sponzorja nagradne igre, ali je nagradna igra aktivna za prikazovanje, nagradno vprašanje z možnimi odgovori ter vnosnimi polji z nazivom in opisom nagrad.

Za izpis forme sem uporabil oznako *Fb:editor* označevalnega jezika FBML, ki jo Facebook pretvori v formo (jezika HTML), ki se pošilja preko metode POST. Naslov URL ciljne strani potrjene forme se določi s parametrom *action*. Naslov URL mora biti relativen. [44]

11.6.1 Enovrstična polja s tekstom

Enovrstična vnosna polja, kot so na primer naziv, nagradno vprašanje in naziv nagrad se lahko predstavi z oznako *Fb:editor-text*. Oznaka se prevede v oblikovano vnosno polje skupaj z naslovom, ki ga podamo oznaki preko parametra *label*.

11.6.2 Izbira iz nabora vrednosti

Za izbiro iz nabora vnesenih pokroviteljev sem uporabil spustni seznam, ki uporablja tehniko pisanja vnaprej. Oznaka za seznam je *Fb:typeahead-input* in je v fazi beta testiranja. Za uporabo je potrebno oznako oviti z *Fb:fbml* in preko atributa *version* zahtevati uporabo jezika FBML verzije 1.1 (`<fb:fbml version="1.1">`).



pok
Pokrovitelj 1
Pokrovitelj 2
Pokrovitelj 3

Slika 32 - Vnosno polje s pisanjem vnaprej.

11.6.3 Vnos besedila

Za vnos opisov sem sprva iskal vnosna polja v stilu urejevalnikov besedil WYSIWYG, ki uporabniku ob urejanju prikazujejo končni izdelek vključno z oblikovanjem teksta. Čeprav Facebook podpira vnos oblikovanega besedila, na moje začudenje ni pripravljenega urejevalnika WYSIWYG. Uporabnik mora torej vnašati v navadno tekstovno polje in ročno pisati oznake za oblikovanje besedila (npr. `` za odebelitev).

Za vnos besedila v večvrstično polje sem uporabil oznako *Fb:editor-textarea*, ki mora biti obdana z oznako *FB:editor*.

11.6.4 Vnos poljubnih elementov forme

Za vnos poljubnih elementov forme, kot je na primer oznaka *checkbox* ali oznaka za spustni seznam *select*, lahko uporabimo oznako *Fb:editor-custom* jezika FBML.

11.6.5 Izbira datuma

Za izbiro datuma in časa pričetka ter zaključka nagradne igre sem želel uporabiti oznako *Fb:editor-date* in jo kombinirati s *Fb:editor-time*. Tu je sledilo prvo veliko razočaranje nad zrelostjo Facebook platforme in uporabe označevalnega jezika FBML.

Oznaka *Fb:editor-date* generira dva spustna seznama. Prvi je za izbiro dneva in ima izpisane vrednosti od 1 do 31. Drugi je za izbiro meseca in ima izpisane tri črkovne okrajšave mesecev. Seznama za izbiro leta ni, kar je za moje potrebe prva pomanjkljivost omenjene oznake. Druga večja pomanjkljivost se pokaže pri preverjanju datumov, oznaka namreč generira le spustne sezname, ne pa tudi logike v jeziku JavaScript, ki bi dinamično preverjala pravilnost izbranih datumov. Generirana spustna seznama tako omogočata uporabniku vnos datumov kot je na primer 31. februar, kar je seveda nepravilno. Ustreznost datuma je potrebno preverjati na strežniku, preverjanje vnosnih polj zgolj na strežniku je za uporabnike zamudno in moteče.

Oznaka kot parametra dovoljuje le naslov *label* in vrednost *value*. Unikatnega imena *name* ni možno podati za parameter. Posledično se lahko na eni strani uporabi oznako le enkrat, če je uporabljena večkrat, se upošteva vrednost zadnje.

Stanje oznaka *Fb:editor* v dokumentaciji ni označeno kot beta, kar me preseneča. Zaradi zgoraj navedenih pomanjkljivosti mi ni jasno, v kakšnem primeru bi oznako lahko koristil. Za moje potrebe je obstoječa oznaka popolnoma neuporabna. Razočaranje je toliko večje, ker polja za izbiro datuma in časa na Facebooku že obstajajo in so zelo dobra. Uporabljena so pri aplikacijah, ki so osnovni del Facebooka, kot je na primer organizacija dogodkov.

The image shows a screenshot of the Facebook event creation interface. It includes fields for 'Title (required)', 'Location', 'Start Time', 'End Time', and 'Privacy'. A calendar widget is open, showing the month of December 2009. The date '6' is selected. The time is set to 4:25 pm. There are 'Create Event' and 'Cancel' buttons at the bottom.

Slika 33 - Facebook že uporablja koledarček za izbiro datumov, ki pa ni na voljo preko oznak jezika FBML.

Oznaka *Fb:time* lahko prejme tri parametre, naslov *label*, unikatno ime *name* in začetno vrednost *value*. Prikaže tri spustne sezname. V prvem so izpisane ure od 1 do 12, v drugem lahko izbiramo minute, v tretjem pa je določena izbira, ali gre za dopoldanski ali popoldanski čas. Minute so izpisane po petnajst minutnih korakih, kar ni vedno željeno. Prav tako ni možno izbirati med 12 in 24-urnim formatom, kar me moti.

Medtem ko obstaja izjemna količina brezplačnih koledarčkov in izbiralcev datumov ter časa napisanih v jeziku JavaScript, jih v jeziku FBJS pravzaprav ni. Napisati sem ga moral sam. Pomoč pri izvedbi sem iskal na spletnih forumih razvijalcev aplikacij Facebook platforme, kjer sem v enem od sporočil [45] našel osnovno kodo, iz katere sem izhajal.

Kodi sem dodal možnost dinamičnega spreminjanja zaloge vrednosti spustnega seznama dni. Tako se v seznamu na primer prikažejo le vrednosti do 28 (ali do 29, če gre za prestopno leto) v primeru, da je izbran mesec februar. Za ugotavljanje števila dni v mesecu sem uporabil:

```
var numDays = 32 - new Date(year, month-1, 32).getDate();
```

Ko funkcija *Date()* kot parameter dobi večje število dni kot jih je v podanem mesecu leta, avtomatično prenese datum v prihodnji mesec. Funkcija *getDate()* vrne dan meseca, začenši s

pričetkom meseca. Tako je na primer 32. marec obravnavan kot 1. april. Če od 32 odštejemo 1, dobimo pravilno število dni v marcu.

Ko sem pripravil kodo v jeziku FBJS, sem se lotil še kodiranja razreda v jeziku PHP, ki mi omogoča generiranje kode jezika FBJS, tako da imam lahko na eni strani več vnosnih polj z različnimi imeni.

```
//vključim razred
require_once('_lib/calendar.php');
$koledarcek = new fbCalendarInput();
//vnosno polje z izbiro datuma in časa
$x->setDateTimeField("prvoIme");
//vnosno polje samo z izbiro datuma
$x->setDateField("drugoIme");
```

Kodo sem prosto objavil na svojem blogu [46] in jo tako dal na voljo vsem razvijalcem, ki bi jo morda še potrebovali.

Slika 34 - Koledarček ki ga generira moja koda.

11.6.6 Primer zapisa oznak za vnosno formo

```
<fb:fbml version="1.1">
  <fb:editor action="index.php?action=dodajNagrada">
    <fb:editor-text label="Naziv" name="title" id="title"
      value="" />
    <fb:editor-custom label="Datum od:">
      <?php $scal->setDateTimeField("dateFrom"); ?>
    </fb:editor-custom>
    <fb:editor-custom label="Nagrada igra aktivna: ">
      <input type="checkbox" name="sweepActive"
        id="sweepActive" class="checkbox" />
    </fb:editor-custom>

    <fb:editor-buttonset>
      <fb:editor-button value="Add"/>
```


```


        <fb:editor-cancel />
    </fb:editor-buttonset>
</fb:editor>
</fb:fbml>


```

11.6.7 Zaslonska maska

Naziv:

Pokrovitelj: 

Datum od: . .  ob :

Datum do: . .  ob :

Opis:

Nagradna igra aktivna:

Vprašanje:

Odgovor:

ta je pravilni:

ta je pravilni:

ta je pravilni:

Nagrada 1:

Nagrada Opis:

Nagrada 2:

Nagrada Opis:

Nagrada 3:

Nagrada Opis:

or

Slika 35 - Vnos nagradne igre.

11.7 Izvedba primera uporabe – Preglej nagradne igre

Iz lokalne podatkovne baze glede na unikatno identifikacijsko številko uporabnika pridobim podatke o nagradnih igrah, kot je to prikazano v diagramu zaporedja.

Klici funkcij so lokalni. Pri oblikovanju prav tako ni nobenih posebnosti, uporabljene so oznake, ki veljajo tudi v označevalnem jeziku HTML.

Medtem ko sem pri dodajanju nagradne igre uporabljal zgolj oznake jezika FBML, ki vključujejo tudi že svoje oblikovanje, sem tu uporabil lastno oblikovanje elementov, prikazanih na zaslonski maski. Oblikovanje sem, kot je to pri spletnih straneh v navadi, shranil v ločeni datoteki *style.css* v formatu CSS (*Cascading Style Sheets*).

Vključevanje datotek formata css je možno na dva načina. Prvi način je klasičen, vendar v primeru Facebooka prinaša nekaj težav. Naslov href mora biti absoluten naslov datoteke css na našem strežniku. Poleg tega Facebook skrbi za avtomatično in neomejeno predpomnenje datoteke. To pomeni, da je potrebno ob vsaki spremembi stilov datoteko preimenovali ali njenemu naslovu dodati parameter z oznako verzije. V razvijalskem okolju je to moteče.

```
<link rel="stylesheet" type="text/css" media="screen"
href="http://yourapp.com/stylesheets/style.css?v=1.0" />
```

Uporabil sem trik in s pomočjo ukazov jezika PHP prebral celotno datoteko oblikovanja css in jo vključil v pogled.

```
<style>
    <?php echo htmlentities(file_get_contents('style.css', true));
?>
</style>
```

11.7.1 Zaslonska maska

The screenshot displays a web interface with a list of contests on the left and an administrator sidebar on the right.

Contests List:

- Traja do 1. 1. 2010**
Testna nagradna igra
 Sponzor: Pokrovitelj 1
 Nagrade: Nagrada 1, Nagrada 2
- Traja do 15. 1. 2010**
Igra z nagradami
 Sponzor: Pokrovitelj 1
 Nagrade: Nagrada 1, Nagrada 2
- Traja do 27. 5. 2011**
Nagradna igra 2
 Sponzor: Pokrovitelj 1
 Nagrade: Super nagrada 1, Darilo leta 2009

Administrator Sidebar:

- Okvir administratorja**
 Dodaj nagradno igro
 Dodaj sponzorja
- Nagradne igre v katerih si udeležen**
 Neka nagradna igra X

Slika 36 - Pregled nagradnih iger.

11.8 Izvedba primera uporabe – Sodeluj

Ko uporabnik iz seznama nagradnih iger izbere zeleno se mu prikaže zaslonska maska za vključitev v nagradno igro.

Iz lokalne podatkovne baze pridobim podatke o zahtevani nagradni igri in jih prikažem uporabniku. Uporabnik izbere odgovor in potrdi formo. Sistem shrani uporabnika kot sodelujočega v lokalno podatkovno bazo. Nato pošljemo obvestilo o udeležbi uporabnika v nagradni igri.

Obvestila se pošiljajo preko funkcije *stream_publish* vmesnika API. Funkciji v parametru *message* podamo naslov sporočila, vsebino, zakodirano v format JSON ali XML, pa prek parametra *attachment*. Ob dnu vsake objave se prikaže povezava za komentiranje in povezava všeč mi je. Temu paru lahko dodamo še druge preko parametra *action_links*, podana vsebina mora biti v formatu JSON.

11.8.1 Primer kode za objavo obvestila v založniku

```
$message = 'se je udeležil nagradne igre!';
$attachment = array(
    'name' => 'Nagradna igra X',
    'href' => 'http://http://apps.facebook.com/igra/',
    'caption' => '{*actor*} se poteguje za nagrado X',
    'description' => 'Nagradna igra poteka do 15.
        januarja! Udeleži se je tudi ti!!!!',
    'media' => array(array('type' => 'image',
        'src' =>
            'http://fb.webeks.net/view/facebook/nagradnaIgra.jpg,
            'href' => 'http://http://apps.facebook.com/webekss/')));

$action_links = array( array(
    'text' => 'Priigraj nagrado!',
    'href' => 'http://http://apps.facebook.com/webekss'));


$attachment = json_encode($attachment);
$action_links = json_encode($action_links);

$facebook->api_client->stream_publish($message,
                                     $attachment,
                                     $action_links);
```


11.8.2 Zaslonska maska

Testna nagradna igra	
Nagrade:	Nagrada 1 Opis nagrade ena Nagrada 2 Nagrada 2 je super!
Vprašanje:	Ali stvar deluje? <input type="radio"/> Morda <input checked="" type="radio"/> Seveda <input type="radio"/> Ne <input type="button" value="Odgovori!"/> or Prekliči
Opis:	Testni opis nagradne igre. Namenjeno demonstraciji delovanja aplikacije za mojo diplomu.
Pokrovitelj: Pokrovitelj 1	
Naslov: Pod devetimi gorami Pošta: 1000 Mesto: Država: Slovenija Splet:	
Pričetek	
1. 1. 2009	
Žrebanje	
1. 1. 2010	

Slika 37 - Uporabnik odgovarja na nagradno vprašanje.



Miha Trtnik se je udeležil nagradne igre!



Nagradna igra X
 Miha se poteguje za nagrado X
 Nagradna igra poteka do 15. januarja! Udeleži se je tudi ti!!!

53 seconds ago via Moja DIPLOMA · [Comment](#) · [Like](#) · [Priigraj nagrado!](#)

Slika 38 - Ko odgovori, se med novicami na njegovem profilu prikaže obvestilo o udeležbi.

11.9 Izvedba primera uporabe Priporoči prijatelju

Na tem mestu sem vključil prvi element virusnega marketinga. Uporabniku prikažem seznam njegovih prijateljev, ki jim lahko priporoči igranje nagradne igre.

Oznaka *FB:request-form* jezika FBML zgradi formo, ki ob potrditvi pošlje zahteve ali povabila zahtevanim uporabnikom. Obvezno je potrebno oznaki podati parameter *type*, ki je pravzaprav naziv povabila. V mojem primeru sem podal niz znakov *nagradnaIgra*, tako da se na spodnjih slikah razbere, kje je niz uporabljen. S parametrom *content* podamo vsebino zahtevka ali povabila. Vsebino zapišemo v formatu označevalnega jezika FBML, ki lahko vsebuje povezave in posebno oznako *Fb:req-choice*, ki definira gumb za pošiljanje zahtevka. Parameter *invite* je privzeto nastavljen na vrednost *false*. V tem primeru se uporabnikom pošilja zahtevke, če vrednost spremenimo na *true*, se pošilja povabila.

Prikazu seznama prijateljev je namenjena oznaka *Fb:multi-friend-input* jezika FBML. Oznaka se preslika v zaslonsko masko, ki uporabniku omogoča označevanje večjega števila svojih prijateljev. Parameter *actiontext* določa besedilo, ki se prikaže uporabniku na vrhu forme. Pomemben je tudi parameter *exclude_ids*, v katerega se lahko poda seznam identifikacijskih števil uporabnikov, ločenih z vejico, ki jih želimo izključiti iz prikaza.

Za vsakega označenega prijatelja se samodejno doda oznaka *hidden* z imenom oznake *ids[]* in vrednostjo, ki ustreza unikatni identifikacijski številki označenega uporabnika.

Seznam prijateljev uporabnika za prikaz pridobi oznaka *Fb:multi-friend-input*. Zaradi tega ni bilo potrebno implementirati načrtovane funkcije *vrniPrijatelje()* v sistemskem vmesniku SVuporabnik. Namesto nje sem dodal funkcijo *vrniPrijateljeZNamescenoAplikacijo()*, ki vrača identifikacijske številke uporabnikovih prijateljev z nameščeno aplikacijo. Funkcija izvede sledeče povpraševanje FQL:

```
$fql = 'SELECT uid FROM user
      WHERE uid IN
      (SELECT uid2 FROM friend WHERE id1='.$platformUserId.')
      AND is_app_user = 1';
```

11.9.1 Primer zapisa oznak za formo priporoči prijatelju

```
<fb:request-form
  action="index.php"
  method="post"
  type="nagradaIgra"
  invite ="true"
  content="Poglej nagradno igro X in se pridruži!
    <fb:req-choice
      url='http://apps.facebook.com/mojaDiploma'
      label='Preveri nagradno igro!!!' />" >
  <fb:multi-friend-selector
    actiontext="Izberi prijatelje, ki jih želiš povabiti k
sodelovanju."
    rows="3" exclude_ids="<? echo $friends; ?>" />
</fb:request-form>
```












11.9.2 Zaslonska maska

Izberi prijatelje, ki jih želiš povabiti k sodelovanju. Skip

Add up to 4 of your friends by clicking on their pictures below.

Find Friends:

Filter Friends ▾ **Only 2 more** All Selected (2)

 A.S. Perov Monash	 Allan Trtnik Topeka, KS	 Alja Kralj	 Alja Kralj
 Alja Kralj Slovenia	 Alja Kralj Slovenia	 Alja Kralj Slovenia	 Alja Kralj
 Alja Kralj	 Alja Kralj Slovenia	 Alja Kralj Slovenia	 Alja Kralj Slovenia

Invite by E-mail Address: Use commas to separate e-mails

Slika 39 - Izbiranje prijateljev za pošiljanje vabil.

Send this nagradnaIgra invitation?

To:

Message:

Cancel

Preview:

You have a nagradnaIgra invitation.

 Miha Trtnik sent an invitation using Moja DIPLOMA:

Poglej nagradno igro X in se pridruži!

Slika 40 - Preden sistem pošlje vabilo, lahko uporabnik pregleda sporočilo ali doda lastno.

11.10 Izvedba primera uporabe Žrebaj

Za žrebanje sem spisal enostavno skripto, ki deluje na lokalni podatkovni bazi in izmed udeležencev nagradne igre naključno izbere dobitnika. Za naključni generator sem uporabil kar vgrajenega v podatkovno bazo MySQL. Povpraševanje SQL v funkciji *zrebajDobitnika()* se glasi:

```
SELECT * FROM udelezenci WHERE pravilnoOdgovoril = '1' ORDER BY
RAND() LIMIT 1
```

Udeležencem je ob žrebu treba poslati elektronsko sporočilo. Seznam udeležencev pridobim s funkcijo *vrniUdelezenceNagradneIgre()* modela *udelezenec*. Za pošiljanje elektronske pošte sem uporabil funkcijo *notifications_sendEmail()* vmesnika API. Seznam naslovnikov se poda kot seznam z vejico ločenih vrednosti.

11.10.1 Primer zapisa kode za pošiljanje elektronske pošte

```
$uids = $udelezenec->vrniUdelezenceNagradneIgre();
$csv = implode(",", $uids);

$naslov = "Žreb nagradne igre: 'Neka nagradna igra X'";
$vsebina = "7. decembra 2009 ob 15.40 smo izvedli"; //...
$htmlVsebina = "<b>7. decembra 2009 ob 15.40</b> smo izvedli"; //...

$facebook->api_client->notifications_sendEmail($csv, $naslov,
$vsebina, $htmlVsebina);
```

11.10.2 Zaslonska maska

Žreb nagradne igre: 'Neka nagradna igra X'

Inbox | X

☆ from **Moja DIPLOMA** [hide details](#) 3:45 PM (0 minu) [Reply](#) ▼

<apps+m4mmks5n@facebookappmail.com>

reply-to Facebook
<apps+m4mmks5n@facebookappmail.com>

to ● Miha Trtnik <...>

date Mon, Dec 7, 2009 at 3:45 PM

subject Žreb nagradne igre: 'Neka nagradna igra X'

mailed-by facebookappmail.com

7. decembra 2009 ob 15.40 smo izvedli žreb nagradne igre z naslovom 'Neka nagradna igra X'.

Srečni dobitnik je Janez Novak.
Vsem ostalim se zahvaljujemo za udeležbo.

Dobitnika prosimo, da nam preko elektronske pošte posreduje osebne podatke, na podlagi katerih lahko izvedemo prenos nagrade dobitniku.

This email was sent by Moja DIPLOMA. You can [disable emails here](#).

Slika 41 - Elektronska pošta, ki jo prejme uporabnik ob žrebu.

12. Zaključek

Facebook vsebuje elemente, ki omogočajo lažje izvajanje in širjenje virusnih idej. Vendar je zgolj orodje in medij po katerem se ideje širijo. Zasnova in splavitev zanimive ideje je še vedno v domeni kreativnega uma.

Programiranje aplikacij za Facebook se ne razlikuje veliko od klasičnega programiranja. Ko razvijalec enkrat ugotovi osnovni princip dela, razvoj poteka hitro. Težave se po mojem mnenju pojavljajo predvsem zaradi odnosa Facebooka do razvijalcev. Število uporabnikov namreč še vedno hitro narašča. Uradna številka ob začetku pisanja tega dela je bila 300 milijonov aktivnih uporabnikov, v času pisanja zaključka, dva meseca kasneje, se je številka povečala na 350 milijonov. Ogromna vrednost, ki se skriva v številu uporabnikov, pa Facebooku omogoča ravnodušen in nezavzet pristop do razvijalcev. Tako je uradna dokumentacija funkcionalnosti Facebook platforme urejena v slogu wiki, kjer lahko vsak dodaja in ureja vsebino. S tem je Facebook preložil odgovornost pisanja dokumentacije na pleča razvijalcev samih. To je sicer v določenih primerih koristno, saj lahko razvijalci dodajajo primere kode in si medsebojno pomagajo. Nikakor pa ta princip ne bi smel biti uporabljen za dokumentiranje osnovnih funkcionalnosti. Vsebinska dokumentacija je na več mestih nepoenotena, ponekod pa celo zastarela.

Označevalni jezik FBML omejuje uporabo standardnega jezika HTML. Vendar je njegov glavni namen razširiti nabor možnih oznak. Žal pa vse oznake, ki so jih dodali, ne delujejo kot bi morale. Primer je oznaka za izbiro datumov.

Če označevalni jezik FBML v večini primerov olajša in pohitri delo, je zgodba z jezikom FBJS drugačna. S spreminjanjem standardnega jezika JavaScripta v jezik FBJS so razvijalce omejili, saj vsa odprta koda v jeziku JavaScript, ni uporabna za projekte, namenjene Facebook platformi. Svojo nepopolno izbiro so po mojem mnenju ugotovili, kar je bil tudi eden od glavnih razlogov za uvedbo IFrameov.

Nezrelost, ki bi ji lahko evfemistično rekli tudi nenehen razvoj, se vidi pri razvoju vmesnika API in delovanja Facebooka kot celote. Za ključne parametre virusnega marketinga, o katerih sem pisal, je v dokumentaciji označeno, da jih bodo v kratkem zamenjali. Datum ni natančno opredeljen, navedeno je le, da zamenjava pride konec leta 2009 ali v začetku 2010. Na kakšen način bodo zamenjali funkcije za pošiljanje obvestil, ni pojasnjeno. Razvijalcu se postavi logično vprašanje, ali bo moral čez noč popravljati svojo kodo, da bo aplikacija po uvedbi sprememb še delovala?

Kljub omenjenim težavam je razvoj aplikacij za Facebook in druge socialne mreže v porastu. Dostop do 350 milijonov uporabnikov namreč odtehta nekaj izgubljenih ur zaradi neurejene dokumentacije in stresno delo ob spremembah vmesnika API.

13. SEZNAM UPORABLJENIH VIROV

- [1] A. Narayanan, V. Shmatikov, »De-anonymizing Social Networks«. Dostopno na: http://www.cs.utexas.edu/~shmat/shmat_oak09.pdf
- [2] J. Goldman, *Facebook Cookbook*, O'Reilly, 2008, poglavje 1
- [3] Metcalfe's law. Dostopno na: http://en.wikipedia.org/wiki/Metcalfe%27s_law
- [4] Vlasta Merc, *Virusni marketing in virusno oglaševanje*, diplomsko delo, 2009
- [5] Dr. Ralph F. Wilson, »The Six Simple Principles of Viral Marketing«, *Web Marketing Today*, 2005. Dostopno na: <http://www.wilsonweb.com/wmt5/viral-principles.htm>
- [6] Dairylea - Robots and Cheesy Movies. Dostopno na: <http://www.atticmedia.com/?p=109>
- [7] A. H. Maslow, »A Theory of Human Motivation«, *Psychological Review*, št. 50, str. 370-396, 1943. Dostopno na: <http://psychclassics.yorku.ca/Maslow/motivation.htm>
- [8] (2009) Google Wave Invite Pulled From eBay After Bids Skyrocket. Dostopno na: http://www.huffingtonpost.com/2009/09/30/google-wave-invite-sellin_n_304396.html
- [9] Viral marketing: Notable examples. Dostopno na: http://en.wikipedia.org/wiki/Viral_marketing#Notable_examples
- [10] D.M. Boyd, N.B. Ellison, »Social Network Sites: Definition, History, and Scholarship«, *Journal of Computer-Mediated Communication*, št. 13, zv. 1, čl. 11, 2007. Dostopno na: <http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html>
- [11] (2007) Social Software Building Blocks. Dostopno na: <http://nform.ca/publications/social-software-building-block>
- [12] Facebook. Dostopno na: <http://en.wikipedia.org/wiki/Facebook>
- [13] Internet Relay Chat. Dostopno na: <http://en.wikipedia.org/wiki/Irc>
- [14] mIRC. Dostopno na: <http://en.wikipedia.org/wiki/Mirc>
- [15] Six degrees of separation. Dostopno na: http://en.wikipedia.org/wiki/Six_degrees_of_separation
- [16] SixDegrees.com. Dostopno na: <http://en.wikipedia.org/wiki/SixDegrees.com>
- [17] Ryze. Dostopno na: <http://en.wikipedia.org/wiki/Ryze>
- [18] friendster.com – Site info from Alexa. Dostopno na: <http://www.alexa.com/siteinfo/friendster.com>
- [19] Friendster Developers Platform. Dostopno na: <http://www.friendster.com/developer>
- [20] MySpace, America's Number One. Dostopno na: <http://mashable.com/2006/07/11/myspace-americas-number-one/>
- [21] OpenSocial. Dostopno na: <http://wiki.developer.myspace.com/index.php?title=Category:OpenSocial>
- [22] Company Timeline. Dostopno na: <http://www.facebook.com/press/info.php?timeline>
- [23] High-Level Specification. Dostopno na: <http://developers.facebook.com/specification.php>
- [24] J. Goldman, *Facebook Cookbook*, O'Reilly, 2008, poglavje 4
- [25] API. Dostopno na: <http://wiki.developers.facebook.com/index.php/API>
- [26] Representational State Transfer. Dostopno na: http://en.wikipedia.org/wiki/Representational_State_Transfer

- [27] Internationalization. Dostopno na:
<http://wiki.developers.facebook.com/index.php/Internationalization>
- [28] Choosing between an FBML or IFrame Application. Dostopno na:
http://wiki.developers.facebook.com/index.php/Choosing_between_an_FBML_or_IFrame_Application
- [29] FBJS. Dostopno na: <http://wiki.developers.facebook.com/index.php/FBJS>
- [30] FQL. Dostopno na: <http://wiki.developers.facebook.com/index.php/FQL>
- [31] Anatomy of a Facebook App. Dostopno na:
http://wiki.developers.facebook.com/index.php/Anatomy_of_a_Facebook_App
- [32] »Zakon o varstvu potrošnikov ZVPot-UPB2«, Ljubljana: Uradni List, št. 98, 2004.
Dostopno na: <http://www.uradni-list.si/1/objava.jsp?urlid=200498&stevilka=4288>
- [33] Model–view–controller. Dostopno na: <http://en.wikipedia.org/wiki/Model-view-controller>
- [34] N. Gulzar, »Fast Track to Struts: What it Does and How«, *TheServerSide.com*, 2002.
Dostopno na:
<http://www.theserverside.com/articles/content/StrutsFastTrack/StrutsFastTrack.pdf>
- [35] (2006) 3-Tier, 3-Layer, MVC: a Trio of Famous Trios. Dostopno na:
http://blogs.msdn.com/diegumzone/archive/2006/10/09/3_2D00_Tier_2C00_-_3_2D00_Layer_2C00_-MVC_3A00_-a-Trio-of-Famous-Trios.aspx
- [36] Multitier architecture. Dostopno na:
http://en.wikipedia.org/wiki/Multitier_architecture
- [37] (2000) Using MVC Pattern in Web Interactions. Dostopno na:
<http://www.uidesign.net/Articles/Papers/UsingMVCPatterninWebInter.html>
- [38] Multitier architecture. Dostopno na:
http://en.wikipedia.org/wiki/Multitier_architecture
- [39] H. Hayder, dr. M. A. Bain, *Learning Facebook Application Development*, Birmingham:Packt Publishing, 2008, str. 29
- [40] W. Graham, *Facebook API Developers Guide*, New York:Apress, 2008, str. 18
- [41] Developer Principles and Policies. Dostopno na:
<http://developers.facebook.com/policy>
- [42] User:Client Libraries. Dostopno na:
http://wiki.developers.facebook.com/index.php/User:Client_Libraries
- [43] Fb:prompt-permission. Dostopno na:
<http://wiki.developers.facebook.com/index.php/Fb:prompt-permission>
- [44] Fb:editor. Dostopno na: <http://wiki.developers.facebook.com/index.php/Fb:editor>
- [45] Need date picker. Dostopno na:
<http://forum.developers.facebook.com/viewtopic.php?id=7002>
- [46] Date-picker FBML component. Dostopno na:
<http://www.webeks.net/programming/facebook-platform/date-picker-fbml-component.html>