

UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A  
INFORMATIKY

VYBUDOVANIE VLASTNEJ SOCIÁLNEJ  
APLIKÁCIE V RÁMCI SOCIÁLNEJ SIETE  
FACEBOOK



UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A  
INFORMATIKY

VYBUDOVANIE VLASTNEJ SOCIÁLNEJ  
APLIKÁCIE V RÁMCI SOCIÁLNEJ SIETE  
FACEBOOK

Bakalárska práca

Študijný program: Aplikovaná informatika  
Študijný odbor: 9.2.9. aplikovaná informatika  
Školiace pracovisko: Katedra Aplikovanej Informatiky  
Školiteľ: Ing. František Gyarfaš, PhD

Bratislava, 2011

Marián Gálik

## Čestné prehlásenie

Čestne prehlasujem, že som túto bakalársku prácu vypracoval samostatne s použitím uvedených zdrojov.

V Bratislave

.....

## **Pod'akovanie**

Ďakujem školiteľovi tejto bakalárskej práce, Ing. Františkovi Gyarfašovi, PhD za cenné rady, pripomienky, odborné vedenie a čas, ktorý mi poskytol. Bez nich by táto práca nedosahovala takých kvalít.

## **Abstrakt**

Tu bude abstrakt, ktorý ešte nemám napísaný a asi ho napíšem až úplne na koniec.

## **Abstract**

And now the English version of abstract.

# Obsah

<b>1 Úvod</b>	<b>5</b>
1.1 Motivácia . . . . .	5
1.2 Cieľ práce . . . . .	5
<b>2 Analýza problematiky</b>	<b>6</b>
2.1 Súčasný trendy vo vývoji a v distribúcii aplikácií . . . . .	6
2.2 Sociálna sieť Facebook . . . . .	6
2.3 Aplikácie sociálnej siete Facebook . . . . .	7
2.3.1 Koncept sociálnych aplikácií . . . . .	7
2.3.2 Typy Facebook aplikácií . . . . .	7
2.4 Technológie umožňujúce integráciu webovej aplikácie so sociálnou sieťou Facebook . . . . .	10
2.4.1 Sociálne pluginy . . . . .	10
2.4.2 Sociálne kanály . . . . .	13
2.4.3 Graph API . . . . .	14
2.4.4 Protokol OAuth 2.0 . . . . .	16
2.4.5 Protokol Open Graph . . . . .	17
2.5 Existujúce podobné riešenia . . . . .	17
2.5.1 My Top Ten . . . . .	17
2.5.2 Top It . . . . .	18
<b>3 Návrh</b>	<b>20</b>
3.1 Požiadavky na aplikáciu . . . . .	20
3.1.1 Funkcionalita aplikácie . . . . .	20
3.1.2 Integrácia so sociálnou sieťou Facebook . . . . .	22
3.2 Konceptuálna analýza . . . . .	23
3.2.1 Akcie používateľov . . . . .	23
3.2.2 Typy položiek . . . . .	25
3.3 Návrh databázového modelu . . . . .	25

3.3.1	Popis tabuliek a ich položiek . . . . .	26
3.4	Objektový návrh a popis tried . . . . .	27
3.4.1	MVC . . . . .	27
3.4.2	Controlléri . . . . .	28
3.4.3	Modely . . . . .	29
<b>4</b>	<b>Implementácia</b>	<b>31</b>
4.1	Použité technológie . . . . .	31
4.1.1	Použité knižnice a frameworky . . . . .	31

# 1. Úvod

## 1.1 Motivácia

Tu bude nejaká motivácia, ktorú asi napíšem až nakoniec

## 1.2 Cieľ práce

Tu bude nejaký cieľ práce

## 2. Analýza problematiky

### 2.1 Súčasný trendy vo vývoji a v distribúcii aplikácií

V minulosti väčšina spoločností vyvíjala a distribuovala softvér na vlastnú peš. Väčšie aj menšie spoločnosti propagovali svoje softvérové produkty na svojich webových stránkach a snažili sa získať priazeň zákazníkov. Tento trend však v súčasnostnej dobe poznačenej globalizáciou postupne upadá a veľké spoločnosti ako Google, Apple a Facebook sa snažia presunúť časť softvérového trhu pod svoju kontrolu.

Typickým príkladom takéhoto správania je spoločnosť Apple. Na iPhone nie je možné bez porušenia záruky neoprávnenými zásahmi inštalovať aplikácie, získané z iných zdrojov ako z Apple Store. Takýto prístup je v konečnom dôsledku výhodný pre všetky tri strany - vývojára, koncového používateľa a spoločnosť Apple. Vývojár má umiestnenú svoju aplikáciu v Apple Store, ktorý navštevuje veľké množstvo ľudí, vďaka čomu sa zvyšuje pravdepodobnosť nákupu aplikácie používateľmi a zisk. Používateľ sa dostane jednoduchšie k aplikáciám a keďže aplikácie prechádzajú procesom schvaľovania spoločnosťou Apple, znižuje sa riziko nákupu chybných aplikácií. Spoločnosť Apple získava províziu z každej predanej kópie aplikácie.

V prípade spoločností Google a Facebook je tento model podobný, avšak v niektorých bodoch sa líši. Nedochádza totiž ku procesu schvaľovania a v prípade sociálnej siete Facebook sa jedná o bezplatné aplikácie. Každopádne ide o súčasný trend a preto má zmysel sa zaoberať vývojom pre spomínané platformy.

### 2.2 Sociálna sieť Facebook

Facebook je sociálna sieť, ktorá bola uvedená do prevádzky vo februári roku 2004. Svojím používateľom umožňuje vytvoriť si vlastný profil, pridať ostatných používateľov medzi priateľov a interagovať s nimi. Interakciu s ostatnými používateľmi je možné dosiahnuť prostredníctvom posielania správ alebo prostredníctvom sociálnych kanálov. V januári roku 2011 sociálnu sieť Facebook využívalo viac ako 600 miliónov aktívnych používateľov.

Kľúčovými funkciami, ktoré sieť Facebook odlišujú od väčšiny ostatných sociálnych sietí je možnosť vytvárania skupín a aplikácií. Skupina je entita, ktorú môže

vytvoriť akýkoľvek používateľ siete Facebook. Ostatní používatelia sa môžu stať jej členmi. Hlavnou funkcionalitou je možnosť autora skupiny šíriť informácie prostredníctvom sociálnych kanálov ku všetkým jej členom.

## 2.3 Aplikácie sociálnej siete Facebook

Platforma Facebook umožňuje tretím stranám vytvárať webové aplikácie, ktoré sú integrované do prostredia siete Facebook a preto sa mnohým bežným používateľom javia ako súčasť sociálnej siete. Vďaka tomu Facebook poskytuje ďaleko rozsiahlejšiu funkcionalitu ako by sa na prvý pohľad mohlo zdať. Úplná integrácia aplikácie do GUI siete Facebook však nie je nutnosťou. Integrovať je možné aj úplne nezávislé webové stránky a poskytovať tak svojim návštevníkom personalizovanosť pri ich návšteve. V súčasnosti je viac ako 2.5 milióna webových stránok integrovaných so sieťou Facebook.

### 2.3.1 Koncept sociálnych aplikácií

Z teoretického hľadiska je možné akýkoľvek webovú aplikáciu integrovať so sieťou Facebook. Keďže integrovaním aplikácie s Facebookom sa aplikácia stáva sociálnou, je nutné brať pri návrhu aplikácie do úvahy princípy, postupy a koncepty zaužívané v sociálnych aplikáciách. Sociálne aplikácie by mali umožňovať používateľom interagovať s ostatnými členmi komunity a plne využívať služby sociálnej siete. Veľmi podstatným je tzv. „virálny efekt“, vďaka ktorému je možné získavať nových používateľov aplikácie pomocou sociálnych kanálov. Aplikácia by teda mala nabádať ostatných používateľov ku interakciám s používateľom, ku ktorému sa viažu sociálnym vzťahom. Väčšina používateľov sociálnej siete takéto správanie očakáva.

### 2.3.2 Typy Facebook aplikácií

Aplikácie sociálnej siete Facebook sa delia na dva základné typy, ktoré sú diametrálne odlišné. Napriek tomu majú pár spoločných črtov. Všetky Facebook aplikácie majú pridelené Application ID, Application Secret, Canvas Page a Canvas URL. Application ID a Application secret slúžia k autentifikácii a identifikácii aplikácie. Canvas Page a Canvas URL, slúži na namapovanie URL vrámci siete Facebook na URL samotnej webovej aplikácie.

#### **Príklad:**

Canvas Page:	top-n-whatever
Canvas URL:	<a href="http://top-n-whatever.freewebsite.net/">http://top-n-whatever.freewebsite.net/</a>

Facebook URL	Namapované na
<code>http://www.facebook.com/top-n-whatever</code>	<code>http://top-n-whatever.freewebtips.net/</code>
<code>http://www.facebook.com/top-n-whatever/argument1</code>	<code>http://top-n-whatever.freewebtips.net/argument1</code>
<code>http://www.facebook.com/top-n-whatever/arg1/arg2</code>	<code>http://top-n-whatever.freewebtips.net/arg1/arg2</code>

## FBML aplikácie

Základnou konvenciou FBML (Facebook Markup Language) aplikácií je spôsob, akým sú integrované do GUI siete Facebook. Skript aplikácie v tomto prípade generuje iba časť webstránky, ktorá nie je samostatne funkčná. Obsahuje špeciálne FBML tagy, ktoré reprezentujú niektoré pre Facebook špecifické prvky GUI. O sparsovanie FBML tagov a nahradenie konkrétnymi GUI prvkami sa stará Facebook. Tok dát v prípade návštevy URL `http://www.facebook.com/top-n-whatever` je nasledovný:

1. Facebook vykoná request na `http://top-n-whatever.freewebtips.net/` a stiahne vygenerovaný XHTML / FBML kód - časť webstránky.
2. Vygenerovaný kód obsahuje FBML tagy, ktoré Facebook sparsuje a nahradí konkrétnymi GUI prvkami.
3. Facebook vygeneruje svoje vlastné GUI a na špecifické miesto umiestni kód konkrétnej FBML aplikácie.
4. Výsledný XHTML kód sa prezentuje koncovému používateľovi.

FBML tagy umožňujú okrem popisovania používateľského rozhrania zapisovať aj logické konštrukcie. Existuje veľké množstvo FBML tagov, z ktorých každý môže mať nastavené rôzne argumenty. Vytváranie nových FBML aplikácií sa neodporúča, keďže podľa oficiálnej dokumentácie FBML API postupne prestane byť podporované. Z týchto dvoch dôvodov táto práca obsahuje na ukážku iba pár FBML tagov, bez podrobnejších popisov. Sú tu uvedené, aby mal čitateľ prehľad.

**fb:comments** Vygeneruje komentáre viažúce sa k istému ID. O logiku súvisiacu s pridávaním a zobrazovaním komentárov sa stará Facebook, nie vývojár aplikácie.

**fb:friend-selector** Vygeneruje formulár, ktorý používateľovi umožní vybrať priateľa. Parametrami je možné niektorých priateľov filtrovať.

**fb:if-is-friends-with-viewer** Tag, ktorého obsah sa zobrazí len v prípade, že aktívny používateľ je priateľom používateľa s bližšie špecifikovaným ID.

## **IFrame aplikácie**

Narozdiel od FBML aplikácií, IFrame aplikácie generujú kompletný XHTML kód, ktorý je možné zobrazit samostatne. Vďaka tomu je možné použiť tento typ aplikácií aj v prípadoch, kedy nechceme webovú aplikáciu plne integrovať do Facebook GUI, pretože chceme, aby bola funkčná aj pre používateľov, ktorí nepoužívajú sociálnu sieť Facebook. Do užívateľského prostredia sociálnej siete Facebook je možné takúto aplikáciu integrovať pomocou XHTML IFrame tagu.

Ako náhrada FBML tagov môžu byť v IFrame aplikácii použité XFBML tagy. Narozdiel od FBML sú XFBML tagy parsované až po načítaní stránky vďaka JavaScriptu, čo urýchli dobu celkového načítania dokumentu. XFBML tagov je oveľa menej a používajú sa iba na renderovanie istých špecifických prvkov GUI siete Facebook. Niektoré XFBML tagy majú svoje varianty v podobe IFrameov - tie parsovať netreba.

Tok dát v prípade návštevy URL `http://www.facebook.com/top-n-whatever` je nasledovný:

1. Facebook načíta základnú štruktúru dokumentu a paralelne počas načítavania ďalších elementov (CSS, skripty, obrázky) sa vykoná priamy request na `http://top-n-whatever.freewebs.net/` a stiahne vygenerovaný XHTML dokument.
2. Výsledný XHTML kód obsahujúci IFrame s vnoreným dokumentom sa prezentuje koncovému používateľovi.
3. XFBML tagy sa dodatočne parsujú pomocou JavaScriptu.

## **Porovnanie FBML a IFrame aplikácií**

### **Výhody IFrame aplikácií**

1. Vďaka paralelnému načítavaniu prostriedkov sa vyznačujú IFrame aplikácie kratším časom načítavania ako FBML aplikácie.
2. Keďže pri IFrame aplikáciách je XHTML kód aplikácie oddelený od zvyšku prostredia siete Facebook, uľahčuje to možnosti používania kaskádových štýlov CSS, pretože nemôže dôjsť ku konfliktu so štýlmi použitými pre GUI siete Facebook.
3. Aplikácia nemusí byť nutne integrovaná do používateľského rozhrania siete Facebook.
4. Používanie vlastných JavaScriptov a JavaScriptových knižníc nie je Facebookom obmedzované.

5. Nie je nutné prispôbovať sa niektorým konvenciám sociálnej siete Facebook, keďže dokument je izolovaný IFrameom
6. Netreba sa učiť nový „jazyk“ pozostávajúci z väčšieho počtu FBML tagov

### **Výhody FBML aplikácií**

1. Možnosť do istej miery programovať logiku aplikácie pomocou FBML tagov
2. Netreba riešiť problémy, ktoré vznikli kvôli použitiu IFrameu - napríklad dynamická zmena výšky stránky

## **2.4 Technológie umožňujúce integráciu webovej aplikácie so sociálnou sieťou Facebook**

### **2.4.1 Sociálne pluginy**

Sociálny plugin je samostatná časť webovej aplikácie, ktorú je možné umiestniť pomocou XFBML tagu alebo IFrameu na webovú stránku. V prípade použitia XFBML tagu je nutné na stránku umiestniť JavaScript, ktorý tag sparsuje - bez toho by bol plugin nefunkčný. Cieľom sociálneho pluginu je sprostredkovať jednoduchú funkcionálnu obohatenú o osobnú skúsenosť.

Práve o personalizovanú skúsenosť sa stará sieť Facebook, tým že vrámci sociálneho pluginu zobrazuje používateľovi relevantné dáta obsahujúce položky súvisiace s jeho priateľmi.

#### **Like Button**

„Like Button“ patrí k najrozšírenejším sociálnym pluginom. Umiestňuje sa na konkrétnu podstránku aplikácie a zobrazuje tlačidlo „To sa mi páči“. Kliknutím na takéto tlačidlo sa vykonajú nasledujúce udalosti:

1. Pri každom tlačidle sa zobrazuje počet používateľov, ktorým sa daná položka páči. Tento počet sa kliknutím zvýši.
2. Používateľovi sa pridá na jeho Facebook nástenku informácia o kliknutí.
3. V niektorých prípadoch sa môže pridať informácia o obľúbenosti položky aj do panela so základnými informáciami o používateľovi. Či sa tak stane závisí od druhu položky.

### Príklad použitia:

```
<fb:like
  show_faces="true"
  width="730"
  href="http://top-n-whatever.freewebsites.net/charts/meta/1"
></fb:like>
```




### Activity Feed


Úlohou sociálneho pluginu „Activity Feed“ je zobraziť stručný prehľad aktivít ostatných používateľov v rámci danej domény. Plugin sa snaží primárne zobrazovať aktivity priateľov práve prihláseného používateľa, je však bežným javom, že plugin z dôvodu nedostatku dát zobrazuje aj všeobecné dáta. Dáta sa zobrazujú v rámci celej domény, je ich však možné filtrovať. Filtre však nefungujú spoľahlivo a častokrát sa stáva, že plugin zobrazuje úplne nerelevantné dáta. Plugin je možné nakonfigurovať tak, aby sa v jeho spodnej časti zobrazovali odporúčania používateľovi. V prípade odporúčaní však filtrovanie nie je možné nastaviť vôbec.


### Príklad použitia:

```
<fb:activity
  site="http://top-n-whatever.freewebsites.net/"
  width="220" height="400" header="true"
  recommendations="true"
></fb:activity>
```


### Recent Activity



**Marian Galik** shared **Testovací rebricek** · about a month ago



**Zuzana Zuzqa Koyšová** shared **Oblíbené hi-gainové gitarové zesilňovače** · about a month ago



**Marian Galik** shared **Top N Whatever** · about a month ago

---


**Naj web stránky**  
One person shared this.


**Testujem**  
One person shared this.


**Top N Whatever**  
One person shared this.

 Facebook social plugin    logged out view

## Login Button

„Login Button“ sprostredkúva funkcionality, umožňujúcu používateľom prihlásiť sa do webovej aplikácie pomocou účtu sociálnej siete Facebook. Tlačidlo login okrem prihlasovania slúži na autorizáciu Facebook aplikácie.

### Príklad použitia:

```
<fb:login-button></fb:login-button>
```



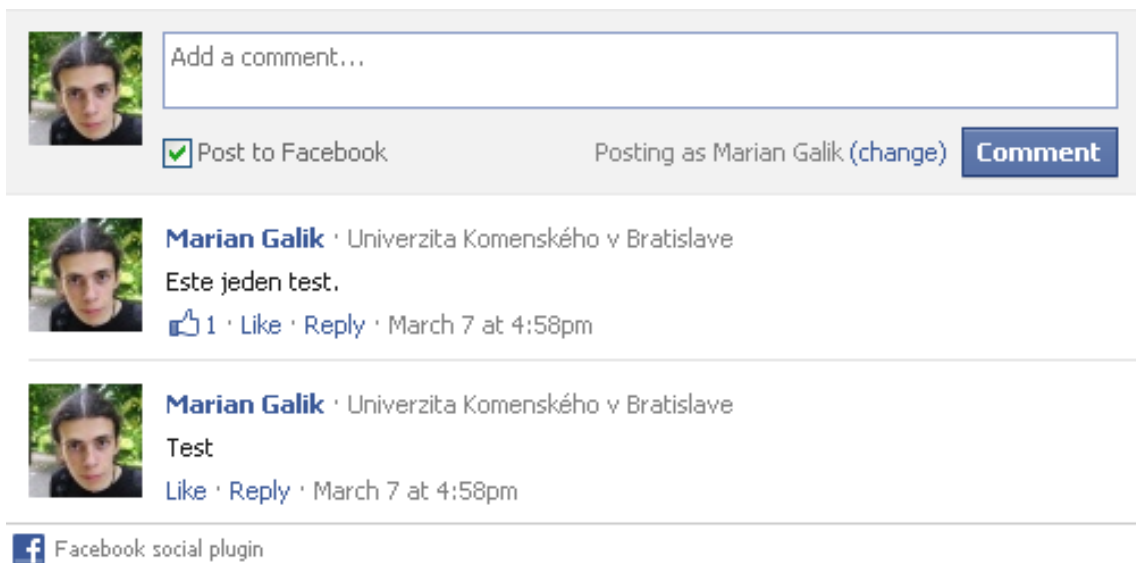
## Comments

Tento sociálny plugin implementuje funkcionality komentovania jednotlivých položiek. Komentáre sa viažu vždy konkrétnej URL adrese. Nie je možné odosielať anonymné komentáre, používateľ sa vždy musí prihlásiť do siete Facebook. Implementovaná je aj administrátorská časť umožňujúca mazať komentáre tvorcom aplikácie a používateľom, ktorých tvorca bližšie špecifikuje. Poradie komentárov je určené sieťou Facebook v závislosti od relevantnosti pre konkrétneho používateľa a nie je možné ho meniť. Taktiež sa môže stať, že niektorí používatelia nevidia komentáre niektorých

iných používateľov. Aký algoritmus používa sieť Facebook na určovanie relevantnosti komentárov nie je presne známe, ale vďaka experimentom sa mi podarilo zistiť, že relevantnosť závisí od priateľských vzťahov medzi jednotlivými používateľmi.

### Príklad použitia:

```
<fb : comments
  numposts="10" width="500" publish_feed="true "
  href="http://top-n-whatever.freewebsites.net/charts/meta/1"
></fb : comments>
```



## 2.4.2 Sociálne kanály

Sociálne kanály slúžia na šírenie informácií medzi používateľmi. Sú veľmi dôležitou súčasťou siete Facebook, pretože ich je možné využiť vo svoj prospech na spropagovanie informácií, produktov, webových stránok a webových aplikácií.

### News Feed

Sociálny kanál „News Feed“ je úzko spätý s nástenkami používateľov, skupín a stránok. Používatelia siete Facebook vytvárajú medzi sebou spojenia tým, že si pridávajú iných používateľov medzi priateľov. Podobné spojenia nevznikajú len medzi dvojicami používateľov, ale aj medzi používateľom a objektom, ktorý používateľ označil ako obľúbený kliknutím na tlačidlo „To sa mi páči“ (Like). V „News Feede“ používateľa sa zobrazujú novinky z nástenok ostatných objektov a používateľov, s ktorými má tento používateľ vytvorené spojenie.

Vytvárať položky v „News Feede“ teda nie je možné priamo, ale pridávaním položiek na nástenku používateľa. Zapisovať na nástenku je možné len v prípade, že

prihlásený používateľ má privilégiá zapisovať na konkrétnu nástenku. Privilégiá komu je dovolené zapisovať na nástenku nastavuje majiteľ nástenky. Samotný zápis je možné vykonať dvomi spôsobmi:

1. **Prostredníctvom JavaScriptového SDK** - v tomto prípade má používateľ aplikácie možnosť rozhodnúť, či naozaj vykonať zápis a taktiež môže pozmeniť text.
2. **Prostredníctvom volania Graph API** - takýto zápis je vykonaný bez vedomia používateľa.

## Request

Request sú najvhodnejším spôsobom ako dať ostatným používateľom vedieť o Facebook aplikácii. Narozdiel od násteniek, requesty sú viditeľné iba používateľom, ktorému boli poslané. Existujú dva druhy requestov:

1. **Používateľom vytvorené requesty** - tento typ requestov je vygenerovaný akciou používateľa v aplikácii. Používateľovi sa zobrazí dialóg pomocou JavaScript SDK, v ktorom vyberie priateľov, ktorým chce poslať request.
2. **Requesty generované aplikáciou** - Facebook aplikácie môžu generovať requesty bez pričinenia používateľa, prostredníctvom Graph API. V tomto prípade sa príjemcovi ako odosielateľ javí samotná aplikácia.

Requesty sú pre používateľov viditeľné prostredníctvom notifikácií a taktiež sa zobrazujú v „dashboarde“.

### 2.4.3 Graph API

Graph API je kľúčové aplikačné rozhranie, ktoré poskytuje sociálna sieť Facebook vývojárom aplikácií. Základným konceptom je, že používatelia siete a objekty záujmu používateľov tvoria vrcholy grafu. Vzťahy medzi jednotlivými entitami su reprezentované hranami. Vďaka tomuto konceptu Facebook poskytuje unifikované rozhranie pre prácu s jednotlivými entitami.

Aby bolo možné pristupovať k objektom unifikovane, každý objekt má svoj jedinečný identifikátor - ID.

## Čítanie dát

Informácií o objekte je možné získať HTTP GET requestom na URL <https://graph.facebook.com/ID>. Výsledkom takéhoto requestu sú dáta vo formáte JSON.

### Príklad:

```
{
  "id": "1444714044",
  "name": "Marian Galik",
  "first_name": "Marian",
  "last_name": "Galik",
  "link": "http://www.facebook.com/marian.magaman.galik",
  "username": "marian.magaman.galik",
  "gender": "male",
  "locale": "en_US"
}
```

Rozhranie taktiež umožňuje pristupovať k zoznamu vrcholov spojených hranou s vrcholom ID. Existuje niekoľko typov hrán, v závislosti od typu súvisaceho objektu: priatelia, novinky, správa z nástenky, „like“, filmy, knihy, poznámky, ...

Keďže tieto informácie nie sú verejne prístupné, okrem ID objektu a typu hrany je potrebné špecifikovať autentifikačný token, ktorý nás oprávňuje pristupovať k údajom konkrétneho používateľa.

### Príklad:

URL pre získanie zoznamu priateľov používateľa s ID 1444714044:

[https://graph.facebook.com/1444714044/friends?access\\_token=...](https://graph.facebook.com/1444714044/friends?access_token=...)

## Zapisovanie dát

Graph API neslúži len na získavanie dát o objektoch, dáta je tiež možné vytvárať. Pri vytváraní dát sa používa HTTP POST request, URL zostáva rovnaké. Parametre závisia od typu objektu. Bežnou hranou, ktorú je možné vytvoriť prostredníctvom Graph API je položka nástenky. Pre vytváranie niektorých typov hrán je nutné, aby aplikácia obdržala privilégiá od používateľa, ktorého sa daný zápis týka.

### Príklad pridania záznamu na nástenku používateľa s ID 1444714044:

URL: <https://graph.facebook.com/1444714044/feed>

<b>Argument:</b>	<b>Hodnota:</b>
access_token	Autentifikačný token
message	Hlavný text (obsah) položky
picture	URL obrázku položky
link	URL položky
name	Názov položky
caption	Nadpis položky
description	Stručný popis položky
source	URL videa položky

#### 2.4.4 Protokol OAuth 2.0

Protokolu OAuth 2.0 slúži na autentifikáciu a autorizáciu aplikácií a na prihlasovanie používateľov siete Facebook. Hlavnou paradigmou protokolu je oddelenie autorizacej časti od samotného získavania dát. Zabezpečuje sa to vygenerovaním autentifikačných tokenov. Pre získavanie prístupu k dátam teda netreba uvádzať citlivé údaje ako heslo používateľa, ku ktorým Facebook aplikácia nemá prístup, ale autentifikačný token.

##### Proces prihlásenia používateľa a autorizácie aplikácie

Prihlasovanie používateľa a autorizácia aplikácie prebieha vždy cez formuláre generované sociálnou sieťou Facebook. Používateľovi je možné vygenerovať takýto formulár pomocou presmerovania na URL v tvare: [https://www.facebook.com/dialog/oauth?client\\_id=arg1&redirect\\_uri=arg2](https://www.facebook.com/dialog/oauth?client_id=arg1&redirect_uri=arg2), kde **arg1** je identifikátor aplikácie a **arg2** je URL, na ktoré bude po ukončení autorizácie používateľ presmerovaný. V prípade potreby je možné špecifikovať doplnkové práva, ktoré Facebook aplikácia vyžaduje. Rovnaké správanie je možné dosiahnuť pomocou sociálneho pluginu „Login Button“.

##### Proces odstránenia aplikácie

V nastaveniach aplikácie je možné špecifikovať URL, na ktoré sieť Facebook vykoná request v prípade, že si niektorý z používateľov aplikáciu odstráni. Vďaka tomu je možné odstraňovať dáta už neexistujúcich používateľov aplikácie. Tento request obsahuje ID používateľa a je podpísaný certifikátom siete Facebook. V prípade správnej implementácie sa teda nemôže stať, že by útočník zneužíval túto funkcionálnu a mazal dáta používateľov, ktorý si aplikáciu neodstránili.

## 2.4.5 Protokol Open Graph

Protokol Open Graph umožňuje popisovať webové stránky dopĺňujúcimi meta dátami, vďaka ktorým dokáže sieť Facebook presnejšie klasifikovať položku v rámci sociálneho grafu. Bez tohoto popisu by sociálne plugíny mali k dispozícii len minimum informácií o danom vrchole.

Dopĺňujúce meta dáta musia byť súčasťou hlavičky HTML dokumentu v tvare:

```
<meta property="atribut" content="hodnota" />
```

### Definované atribúty protokolu Open Graph

**og:title** Názov objektu

**og:type** Typ objektu (napríklad: movie, activity, band, school, ...)

**og:image** Obrázok vzťahujúci sa k danému objektu. Rozmer obrázka musí byť aspoň 50 x 50 pixelov a pomer strán nesmie byť väčší ako 3 : 1.

**og:description** Stručný popis objektu

**og:url** URL adresa objektu

**og:site\_name** Názov webovej stránky obsahujúcej objekt

Protokol Open Graph obsahuje ďalšie atribúty, ktoré nie sú vo vyššie uvedenom zozname, keďže nie sú použité v tejto bakalárskej práci.

## 2.5 Existujúce podobné riešenia

### 2.5.1 My Top Ten

URL: <http://apps.facebook.com/mytopten>

#### Klady

1. Možnosť výberu kategórie rebríčka.
2. Počas písania položiek aplikácia pomocou technológie AJAX zobrazuje návrhy na doplnenie názvu položky.
3. Možnosť pridávať popis k položkám.
4. Pri položkách sa zobrazuje možnosť nákupu danej položky cez internetový obchod Amazon.com.

## Zápory

1. Nie je možné vytvoriť viacej rebríčkov rovnakej kategórie.
2. Aplikácia obsahuje iba 9 základných kategórií a nie je možné vytvoriť vlastnú kategóriu.
3. Rebríček nemôže obsahovať viacej ako 10 položiek.
4. Rebríčky a položky nie je možné hodnotiť a komentovať, interakcia s priateľmi je na veľmi nízkej úrovni.
5. Nie je možné zasahovať do rebríčkov priateľov.
6. Do popisu položky nie je možné embedovať video.

### 2.5.2 Top It

URL: <http://apps.facebook.com/mytopit>

## Klady

1. Možnosť výberu z viacerých typov rebríčkov. V závislosti od typu rebríčka je možné do rebríčka embedovať YouTube videá a pridávať fotografie z albumov umiestnených na sociálnej sieti Facebook.
2. Možné vybrať jednu z preddefinovaných kategórií rebríčkov.
3. Počet položiek nie je limitovaný.
4. Možnosť pridávať popis k položkám.
5. Rebríčky je možné komentovať a „Likeovať“ - interaktivita je na dobrej úrovni.
6. Možnosť nastaviť ku každému rebríčku zvlášť viditeľnosť pre všetkých používateľov siete Facebook.
7. Možnosť vytvoriť klon rebríčka.

## Zápory

1. Pri písaní položiek aplikácia neponúka návrhy na doplnenie.
2. Nie je možné kombinovať rôzne elementy v rámci rebríčka. (Napríklad YouTube video k prvej položke a fotografiu z albumu k druhej položke)

3. Nie je možné zasahovať do poradia rebríčka priateľov a hodnotiť jednotlivé položky.
4. Verejné rebríčky sú zobrazované mierne neprehľadne.

## 3. Návrh

Táto kapitola rozoberá proces stanovenia požiadaviek na výslednú aplikáciu, ich analyzovania a vytvorenia návrhu.

### 3.1 Požiadavky na aplikáciu

Keďže na webovú aplikáciu je možné sa pozeráť z dvoch rôznych hľadísk, rozdelil som požiadavky do dvoch skupín. V prvej časti sa zaoberám funkcionalitou aplikácie ako samostatného celku. Odhliadam teda od faktu, že aplikácia je integrovaná so sociálnou sieťou Facebook a vymedzujem iba tú funkcionalitu, ktorá s Facebookom nesúvisí a je špecifická pre samotnú aplikáciu. V druhej časti popisujem interakciu a integráciu aplikácie so sociálnou sieťou.

#### 3.1.1 Funkcionalita aplikácie

##### Spravovanie rebríčkov

Aplikácia umožní vytvárať používateľom rebríčky obľúbených položiek. Každý rebríček má názov a môže obsahovať podrobný popis, rebríčka ako celku.

Okrem toho rebríček pozostáva z usporiadaného zoznamu  $N$  položiek, kde  $N \in [1, \infty)$ . Položka rebríčka pozostáva z názvu a môže mať špecifikovaný podrobnejší popis. Pri prezeraní rebríčka bude možné kliknúť na všetky URL, ktoré používateľ zadal do popisu rebríčka. Výnimku tvoria URL na videá umiestnené na portály YouTube.com. Tie budú embedované vo forme videa.

Používateľ aplikácie môže vytvárať, upravovať a mazať svoje rebríčky. Pri zmazení rebríčka dochádza k zmazaniu všetkých dát súvisiacich so zmazaným rebríčkom (položky, hodnotenia položiek, posledné návštevy rebríčka, odpovede na rebríček). Aplikácia umožňuje vytvárať aj anonymné rebríčky, ktoré nemajú špecifikovaného majiteľa.

##### Spätná väzba

Ku rebríčku je možné pridávať komentáre a označiť rebríček za obľúbený. Každú položku rebríčka je možné hodnotiť na škále 1 - 5. Ostatní používatelia aplikácie môžu

pridávať k ostatným rebríčkom položky o ktorých si myslia, že v rebríčku chýbajú. Autor rebríčka má právo tieto položky zmazať alebo pridať do svojho rebríčka.

Na základe hodnotenia jednotlivých položiek je možné rebríček preusporiadať a tým získať verziu rebríčka, ktorá narozdiel od pôvodnej verzie nereprezentuje iba názor autora, ale názor celej sociálnej skupiny, ktorá s rebríčkom interagovala. V tejto skupinovej verzii sa okrem pôvodných položiek zobrazujú položky, ktoré nie sú súčasťou rebríčka, ale boli pridané ostatnými používateľmi.

Ku každému rebríčku je možné pridať niekoľko vlastných rebríčkov ako odpoveď („Chart response“). Odkazy na odpovede sa zobrazujú pri prezeraní rebríčka.

## **Prehľad**

Aplikácia si pre každého používateľa pamätá maximálne 3 naposledy navštívené rebríčky a prehľad noviniek. Novinky pozostávajú z chronologicky usporiadaných udalostí týkajúcich sa prihláseného používateľa a jeho priateľov. Typy noviniek:

1. Pridanie rebríčka
2. Editovanie rebríčka
3. Jeden z používateľov pridal položku k rebríčku prihláseného používateľa
4. Ohodnotenie položky rebríčka
5. Vytvorenie odpovede k jednému z rebríčkov

## **Nastavenia**

Aplikácia umožňuje nastaviť úroveň súkromia používateľa a spôsob akým komunikovať prostredníctvom sociálnych kanálov siete Facebook. Používateľ aplikácie si môže zvoliť skupinu používateľov, ktorý vidia jeho rebríček a skupinu používateľov, ktorý s ním môžu interagovať. Na výber sú nasledujúce úrovne súkromia:

1. Nízka - prezerateľ rebríčky a interagovať s nimi môžu všetci
2. Stredná - prezerateľ rebríčky môžu všetci, ale interagovať s nimi môžu iba priatelia
3. Vysoká - prezerateľ a interagovať s rebríčkami môžu iba priatelia

## **Zjednodušenie používania**

Pre zjednodušenie používania bude možné vytvoriť nový rebríček na báze iného existujúceho rebríčka. Táto funkcionálna ulahčí používateľom prácu v prípade, že sa

chcú pri tvorbe vlastného rebríčka inšpirovať iným existujúcim rebríčkom. Pri vyplňaní názvov položiek rebríčka sa používateľovi zobrazí možnosť automatického doplnenia názvu a popisu položky, podľa podobných položiek existujúcich v anonymných rebríčkoch.

### **3.1.2 Integrácia so sociálnou sieťou Facebook**

#### **Základná integrácia**

Používatelia budú autentifikovaní prostredníctvom svojho Facebook konta. Aplikácia bude využívať zoznam používateľových FB priateľov na kategorizáciu rebríčkov. Rebríčky budú teda zaradené do jedného z 3 zoznamov v závislosti od autora rebríčka.

1. Súkromné rebríčky prihláseného používateľa.
2. Rebríčky priateľov prihláseného používateľa.
3. Rebríčky používateľov, ktorý nie sú v zozname priateľov práve prihláseného používateľa.
4. Anonymné rebríčky (rebríčky bez autora).

#### **Publikovanie prostredníctvom sociálnych kanálov**

Aplikácia umožní používateľom publikovať informácie o špecifických akciách. Záznamy sa zapisujú na nástenku používateľa, čoho žiadaným side efektom je výskyt v „News Feede“ jeho priateľov. Nasledujúci zoznam sumarizuje akcie, pri ktorých dochádza k publikovaniu na nástenku majiteľa rebríčka.

1. Vytvorenie rebríčka
2. Úprava rebríčka
3. Pridanie položky iným používateľom
4. Ohodnotenie položky
5. Pridanie „Chart response“

Každý používateľ si môže nastaviť spôsob, akým má ku publikovaniu dochádzať. Na výber sú nasledujúce možnosti:

1. Nepublikovať vôbec

2. Pred publikovaním sa spýtať, či si používateľ želá publikovať a dať mu možnosť upraviť výsledný text.
3. Publikovať automaticky

### **Vyjadrovanie názoru prostredníctvom sociálnych pluginov**

Používateľom bude umožnené vyjadriť svoj názor v rámci aplikácie a sociálnej siete Facebook prostredníctvom sociálnych pluginov „Like Button“ a „Comments“. Pluginy budú umiestnené na stránke s konkrétnym rebríčkom a budú pre každý rebríček nezávislé. Súčasťou úvodnej stránky s prehľadom bude sociálny plugin „Activity Feed“, ktorý bude zachytávať aktivitu používateľov v rámci siete Facebook.

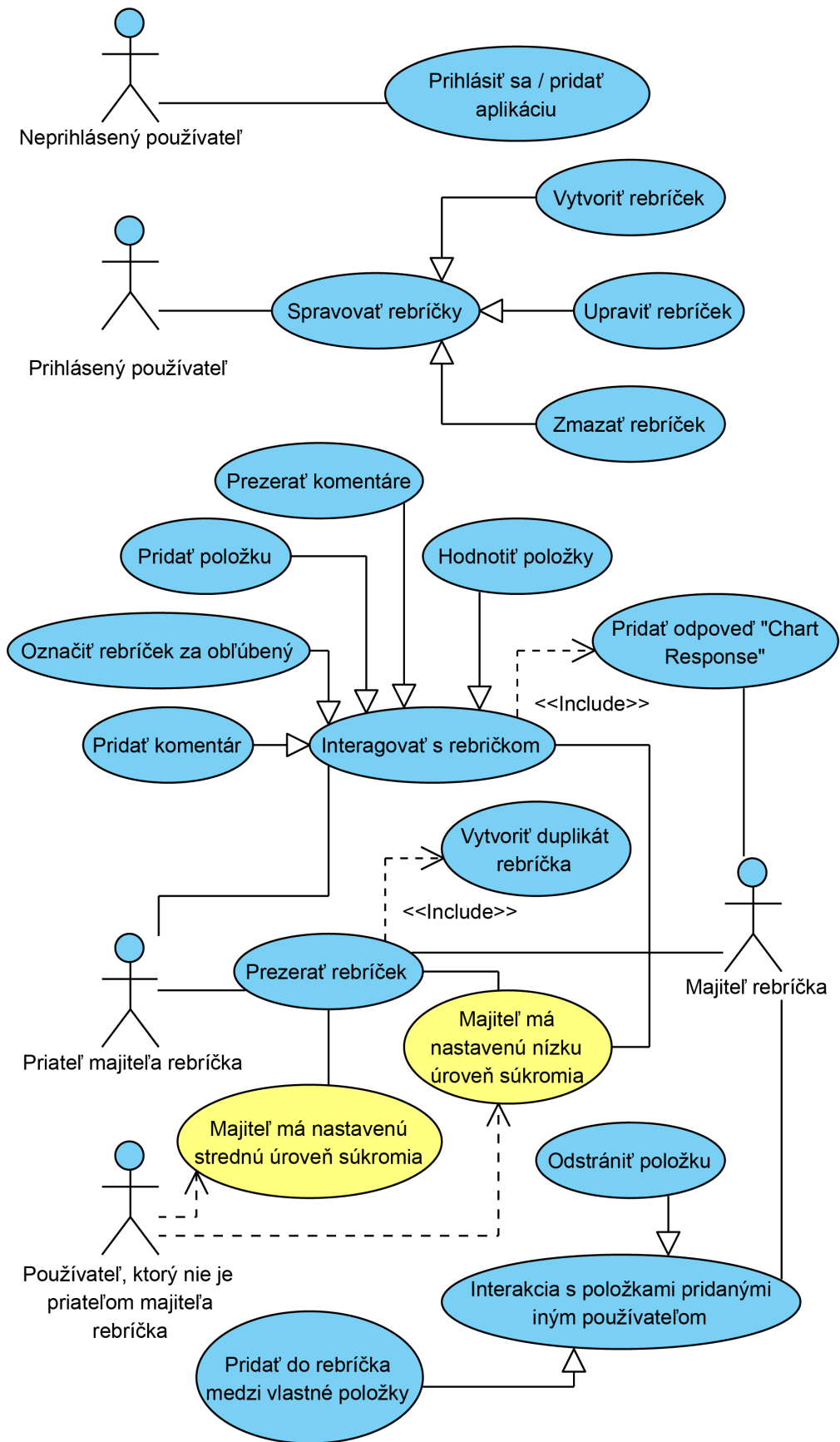
### **Získavanie nových používateľov prostredníctvom requestov**

Aplikácia umožní používateľom prostredníctvom posielania Facebook requestov nabádať svojich priateľov, ktorí ešte nie sú používateľmi aplikácie, aby aplikáciu začali používať.

## **3.2 Konceptuálna analýza**

### **3.2.1 Akcie používateľov**

Keďže aplikácia poskytuje široké spektrum funkcionality a interaktivity medzi jednotlivými používateľmi, je dôležité správne navrhnuť optimálne obmedzenia funkcionality pre isté skupiny používateľov. Keďže sa jedná o Facebook aplikáciu, koncept práv by mal byť podobný konceptu samotnej sociálnej siete. Z dôvodu zachovania spokojnosti čo najväčšej skupiny používateľov je možné, niektoré z obmedzení nastaviť individuálne pre každého používateľa. Nasledujúci use-case diagram zobrazuje povolenú funkcionality:

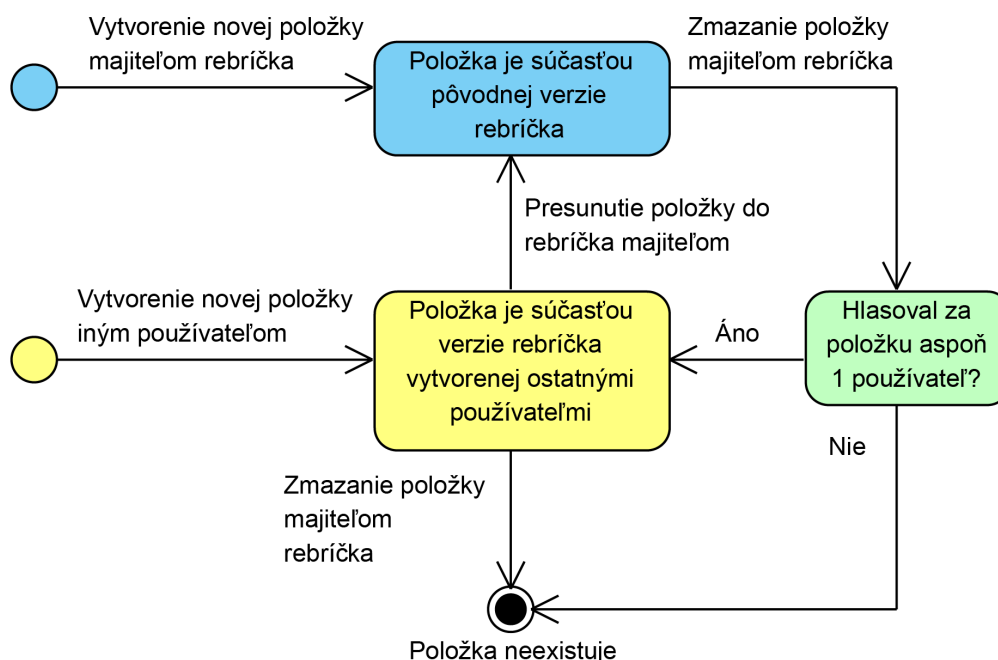


### 3.2.2 Typy položiek

Položky rebríčkov je možné rozdeliť do dvoch kategórií:

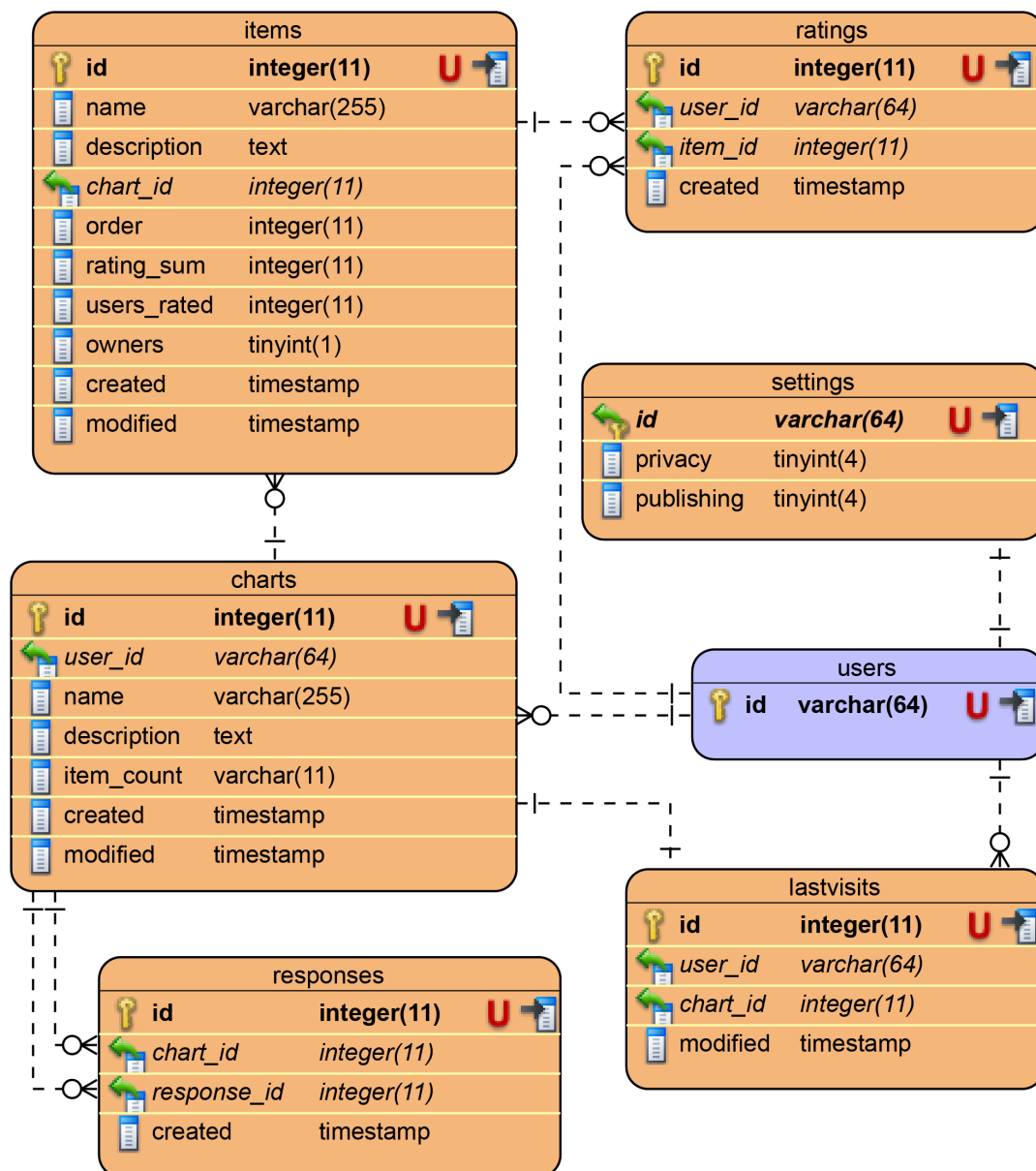
1. Položky vytvorené majiteľom rebríčka
2. Položky pridané do rebríčka ostatnými používateľmi

Majiteľ rebríčka má možnosť preniesť položku pridanú iným používateľom medzi vlastné položky. Za istých okolností je možné docieľiť aj opačný proces, a teda preniesť položku pridanú majiteľom rebríčka medzi položky, ktoré patria do rebríčka podľa ostatných používateľov. Táto situácia nastane v prípade, že majiteľ sa rozhodne zmazať položku, za ktorú hlasoval aspoň jeden používateľ. Nasledujúci stavový diagram zobrazuje životný cyklus položky:



### 3.3 Návrh databázového modelu

Aplikácia bude využívať na ukladanie dát relačnú databázu. Nasledujúci relačno-entitný diagram zobrazuje jednotlivé tabuľky databázy a vzťahy medzi nimi. V diagrame je uvedená tabuľka „users“, ktorá obsahuje informácie o používateľoch. Keďže tieto dáta nie sú spravované samotnou aplikáciou, ale sociálnou sieťou Facebook, táto tabuľka nebude súčasťou databázy. V diagrame je uvedená len kvôli logickosti znázornenia vzťahov. Pre identifikátor používateľa nie je použitý typ Int(11) z dôvodu, že identifikátory, ktoré používa sociálna sieť Facebook pre používateľov sú v mnohých prípadoch dlhšie ako rozsah hodnôt typu Int(11).



### 3.3.1 Popis tabuliek a ich položiek

#### Tabuľka „charts“

Položky `user_id` je cudzím kľúčom k identifikátoru majiteľa rebríčka. V prípade, že je hodnota stĺpca `user_id` nulová, ide o anonymný rebríček. Z dôvodu optimalizácie vykonávania dotazov tabuľka obsahuje stĺpec `item_count`, ktorý reprezentuje počet položiek rebríčka. Túto hodnotu by bolo možné rátať pomocou SQL funkcie `COUNT()`, avšak doba vykonávania SQL dotazu by sa tým predĺžila. Medzi položky rebríčka sa rátajú iba položky pridané majiteľom, teda tie, ktoré majú stĺpec `owners` nastavený na 1 (True).

### Tabuľka „items“

Položka `owners` má hodnotu 1 (True) v prípade, že položka bola pridaná majiteľom rebríčka. Ak bola pridaná iným používateľom, má hodnotu 0 (False). Stĺpec `rating_sum` obsahuje súčet všetkých hlasovaní za danú položku. Stĺpec `users_rated` je použitý z dôvodu optimalizácie a reprezentuje počet používateľov, ktorý hlasovali za danú položku. Z hodnôt `rating_sum` a `users_rated` je možné vypočítať priemerné hodnotenie položky.

### Tabuľka „ratings“

Táto tabuľka zámerne neobsahuje stĺpec, ktorý by obsahoval číselnú hodnotu hlasu. Vďaka tomu vo väčšine prípadov nie je možné zistiť ako ktorý používateľ hlasoval. Jedinou dôvodom existencie tejto tabuľky je to, aby bolo možné implementovať funkcionality, ktorá zabráni používateľovi hlasovať viackrát za jednu a tú istú položku.

### Tabuľka „settings“

Stĺpec `privacy` reprezentuje úroveň súkromia nastaveného pre daného používateľa. Možné hodnoty sú položky: 0 - nízka, 1 - stredná, 2 - vysoká. Stĺpec `publishing` určuje spôsob, akým šíriť informácie prostredníctvom sociálnych kanálov. Možné hodnoty: 0 - nešíriť, 1 - pred šírením sa spýtať používateľa, 2 - šíriť automaticky.

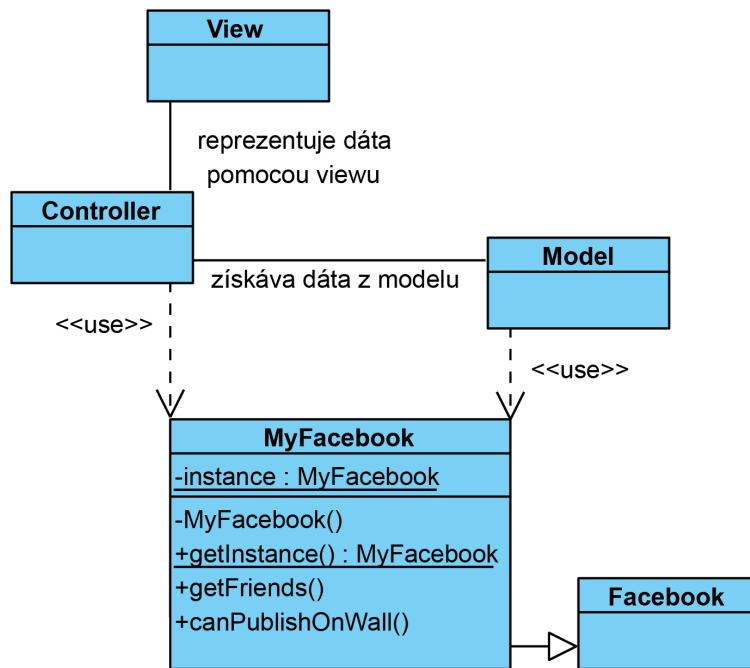
### Tabuľka „responses“

Keďže každý rebríček môže mať viacej rebríčkov ako „Chart response“ a súčasne každý rebríček môže byť „Chart response“ inému rebríčku, rámci tabuľky `charts` existuje vzťah typu „Has many and belongs to many“. Táto tabuľka teda slúži ako spájajúca tabuľka pre „HABTM“ vzťah. Stĺpec `chart_id` reprezentuje identifikátor rebríčka, ku ktorému sa vzťahuje odpoveď a stĺpec `response_id` je indentifikátorom rebríčka, ktorý je odpoveďou.

## 3.4 Objektový návrh a popis tried

### 3.4.1 MVC

Facebook aplikácia bude implementovať návrhový vzor MVC, ktorého cieľom je oddeliť dátovú vrstvu (Model) od prezentačnej (View) a od riadiacej logiky (Controller). Nasledujúci diagram zobrazuje základné triedy aplikácie:



Súčasťou diagramu je trieda Facebook, ktorá tvorí PHP SDK poskytnuté sociálnou sieťou Facebook. Trieda MyFacebook je odvodená od triedy Facebook a rozširuje jej funkcionality o cachovanie získaných dát. Okrem toho implementuje návrhový vzor Singleton, ktorý zabezpečuje existenciu maximálne jednej inštancie triedy v rámci celej aplikácie a jednoduchý prístup k tejto inštancii z ktoréhokoľvek miesta v aplikácii.

### 3.4.2 Controlléri

Controlléri sú jediným vstupným bodom aplikácie. Ich verejné metódy reprezentujú jednotlivé akcie, ktoré je možné v aplikácii vykonávať zásahom používateľa.

Väčšina názvov controllérov použitých v aplikácii a ich akcií je jednoznačná a nie je potrebné písať ich popis. Pre niektoré to však úplne jednoznačné nie je a preto ich stručný popis uvediem.

#### Realtime Controller

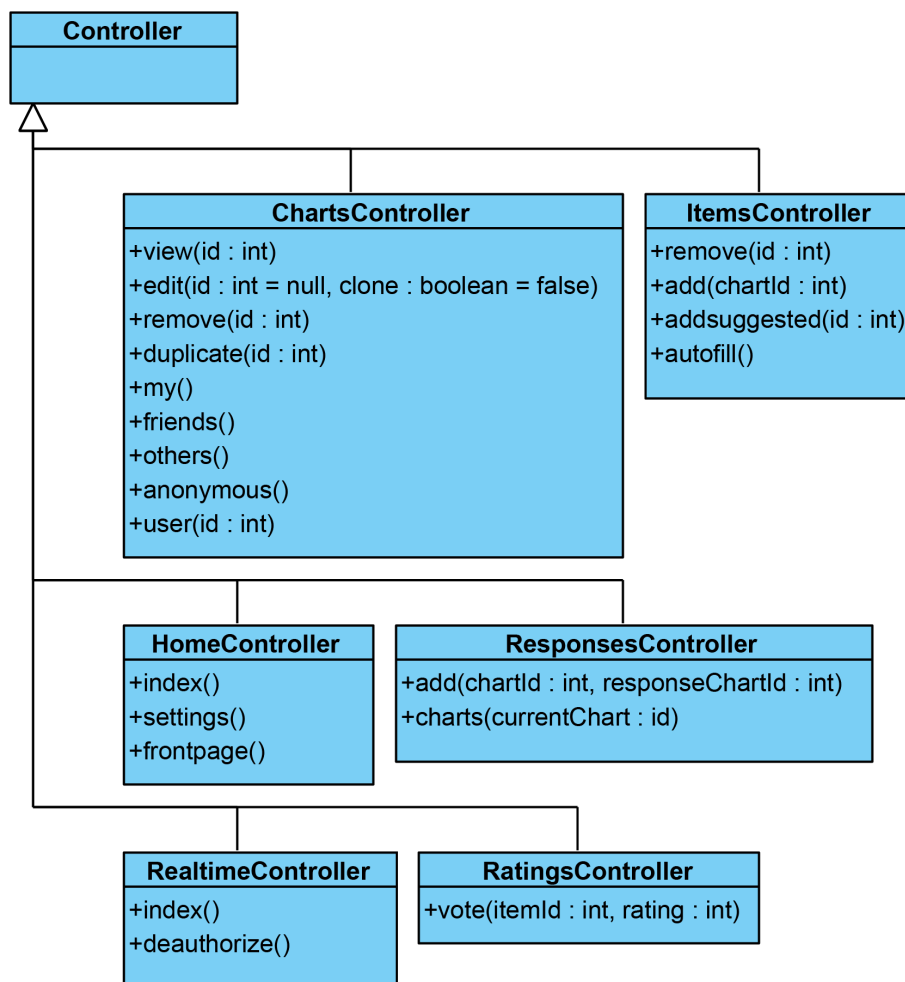
„RealtimeController“ slúži na aktualizáciu cachovaných údajov na základe podnetu zo sociálnej siete Facebook prostredníctvom technológie „RealTime updates“. Akcia index sa stará o prijatie požiadavky o zmene údajov (a následné premazanie časti vyrovnávacej pamäte). Akcie deauthorize má na starosti zmazanie dát používateľa, ktorý si zo svojho Facebook konta odstránil aplikáciu.

## Ostatné nejasné akcie

Akcia „charts“ controlléra Responses slúži na získanie zoznamu rebríčkov, ktoré je možné pridať ako „Chart Response“ k rebríčku špecifikovanému parametrom akcie. Akcia „charts“ index controlléra Home zobrazuje úvodnú stránku prihláseného používateľa s prehľadom noviniek. Vstupný bod pre neprihlásených používateľov tvorí akcia frontpage, ktorá umožňuje prihlásenie a autorizáciu aplikácie prostredníctvom sociálneho pluginu „Login Button“.

## Diagram

Nasledujúci diagram zobrazuje controlléri potrebné pre funkcionality aplikácie a ich akcie:



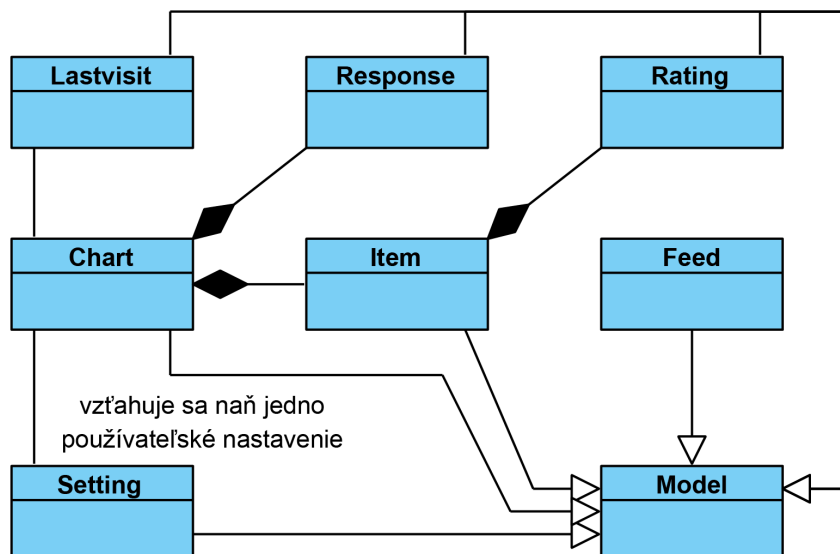
### 3.4.3 Modely

Vo väčšine prípadov platí pravidlo, že každý model reprezentuje práve jednu tabuľku z databázy. Stará sa o jej aktualizáciu, o validáciu dát a o získavanie dát z

databázy. V prípade závislosti na iných tabuľkách, používa modely iných tabuliek pre zabezpečenie konzistentnosti dát. Špeciálnym prípadom je model „Feed“, ktorý nepracuje s konkrétnou tabuľkou a sprostredkúva jednotné rozhranie k najnovším dátam z rôznych tabuliek (Charts, Items, Responses a Ratings).

## Diagram

Nasledujúci diagram zobrazuje všetky modely a vzťahy medzi nimi (sú úzko späté so vzťahmi medzi tabuľkami):



# 4. Implementácia

## 4.1 Použité technológie

Na implementovanie aplikácie bude použitý skriptovací jazyk PHP 5, ktorý poskytuje dostačujúce prostriedky. Aplikácia bude používať MySQL relačnú databázu na ukladanie dát. Z dôvodu použitia Facebook SDK, v ktorom je implementované vykonávanie API volaní prostredníctvom knižnice cURL (Client URL Library), je nutné, aby bola na serveri prevádzkujúcom aplikáciu nainštalovaná.

Aplikácia bude umiestnená na Apache webserveri a bude vyžadovať modul `mod_rewrite`, ktorý poskytuje vytváranie prepisovacích pravidiel pre URL. Vďaka tomuto funkcionalite bude aplikácia používať SEF URL (Search Engine Friendly URL).

PHP skripty budú generovať XHTML dokumenty (podľa štandardu 1.0 Transitional), ktoré budú naformátované pomocou CSS štýlov. V aplikácií budú vo veľkej miere použité technológie JavaScript a AJAX na dynamické menenie obsahu, vďaka ktorým bude aplikácia pôsobiť interaktívnejšie a lepšie zapadne do rozhrania sociálnej siete Facebook, ktoré taktiež tieto technológie využíva.

### 4.1.1 Použité knižnice a frameworky