

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Portál seriálových postav

Lukáš Vraný

Bakalářská práce

2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lukáš Vraný**
Osobní číslo: **I09312**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Portál seriálových postav**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je návrh a realizace webového portálu o seriálových postavách a jejich obsazení herci s více kolovým hodnocením seriálů, epizod či seriálových postav.

Návrh databázového modelu musí umožnit uspořádání libovolného počtu kol při klání jednotlivých seriálových postav, nominace postav a sledování hlasování jednotlivých uživatelů.

Aplikace bude vyvíjena při využití technologií HTML5, PHP, CSS, XML, Javascript a relačních databází.

Teoretická část se bude zabývat problematikou hlasování na webových stránkách (anonymní vs. neanonymní) a problémy při anonymních anketách z hlediska identifikace uživatele, který již hlasování provedl. Dále bude v teoretické části diskutována problematika vyhodnocení návštěvnosti stránek, tj. prokliků, návštěv, návštěvníků, zobrazení stránek a jedinečných zobrazení stránek.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. SCHAFER, S., M.: HTML, XHTML a CSS ? Bible pro tvorbu www stránek. 4.vyd. Grada, 2009. ISBN 978-80-247-2850-6
2. GUTMAS, A., RETHANS, D., BAKKEN, S., S.: Mistrovství v PHP 5. Computer Press, 2007. ISBN 978-80-251-1519-0
3. HARRINGTON, J., L.: SQL clearly explained. 3.vyd. Elsevier, 2010. ISBN 978-0-12-375697-8
4. HERNANDEZ, J., M., Viescas, L., J.: Myslíme v jazyku SQL: Tvorba dotazů. Grada, 2004. ISBN 978-80-247-0899-7

Vedoucí bakalářské práce:

RNDr. David Žák, Ph.D.

Katedra informačních technologií

Datum zadání bakalářské práce: **16. prosince 2011**

Termín odevzdání bakalářské práce: **11. května 2012**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

Ve Starém Kolíně dne 11. 05. 2012

Lukáš Vraný

Poděkování

Chtěl bych poděkovat vedoucímu mé bakalářské práce panu RNDr. Davidu Žákovi, Ph.D. za cenné rady, připomínky a konzultace při tvorbě této práce. Dále bych chtěl poděkovat své rodině za podporu během studia.

Anotace

Cílem práce je návrh a realizace webového portálu o seriálových postavách a jejich obsazení herci s více kolovým hodnocením seriálů, epizod či seriálových postav. Návrh databázového modelu musí umožnit uspořádání libovolného počtu kol při klání jednotlivých seriálových postav, nominace postav a sledování hlasování jednotlivých uživatelů.

Aplikace bude vyvíjena při využití technologií HTML5, PHP, CSS, XML, JavaScript a relačních databází.

Teoretická část se bude zabývat problematikou hlasování na webových stránkách (anonymní vs. neanonymní) a problémy při anonymních anketách z hlediska identifikace uživatele, který již hlasování provedl. Dále bude v teoretické části diskutována problematika vyhodnocení návštěvnosti stránek, tj. prokliků, návštěv, návštěvníků, zobrazení stránek a jedinečných zobrazení stránek.

Klíčová slova

webový portál, seriálové postavy, MySQL, PHP, HTML, Oracle

Title

The series characters's portal

Annotation

The aim of this thesis is to design and realize portal of serial characters and their cast actors with multi-round rating serials, episodes or serial characters. Database model must allow configuration of any number rounds in each duel of serial characters, nomination of characters and tracking poll for each user.

The application will be developed by using HTML5, PHP, CSS, XML, JavaScript and relational databases.

The theoretical part deals with the issue of voting on websites (anonymous vs. Non-anonymous) and problems connected with anonymous polls in terms of users who (have) already voted. Also there will be discussion about problems with evaluation of site traffic, i.e. click-through, visits, visitors, page views and unique page views.

Keywords

Web portal, series characters, MySQL, PHP, HTML, Oracle

Obsah

Seznam zkratk	8
Úvod	10
1 Hlasování na internetu	11
1.1 Identifikace uživatele.....	11
1.1.1 Příklady identifikace.....	11
1.2 Neanonymní hlasování	13
1.2.1 Vlastní registrace a přihlášení.....	13
1.2.2 Přihlášení přes Facebook	14
1.3 Anonymní hlasování.....	14
1.4 Možnosti ovlivnění hlasování.....	16
1.5 Ankety třetích stran	17
2 Vyhodnocení návštěvnosti	18
2.1 Používané termíny	18
2.2 Heat mapa kliknutí	19
2.3 Heat mapa oční kamery	20
3 Nástroje pro vyhodnocení návštěvnosti	22
3.1 Nejpoužívanější nástroje webové analytiky	22
3.2 Nástroje webové analytiky v České republice.....	24
3.3 Postup při výběru analytického řešení.....	25
4 Analýza webové aplikace	26
4.1 Požadavky.....	26
4.2 Use-case diagram.....	26
4.3 Rich-picture	28
4.4 Activity diagram	29
5 Použité technologie	31
5.1 HTML.....	31
5.2 CSS	31
5.3 PHP.....	32
5.4 JavaScript	32
5.5 Relační databáze	33
5.6 Ostatní.....	33

6	Návrh a tvorba databáze.....	35
6.1	Tabulky.....	35
6.2	Procedury.....	40
6.3	Funkce	46
6.4	Ostatní.....	47
7	Vývoj aplikace.....	48
7.1	Dynamika aplikace	48
7.2	Formuláře	48
7.3	Hesla a bezpečnost.....	49
7.4	Uživatelé.....	50
7.4.1	Registrace	50
7.4.2	Přihlášení a odhlášení	51
7.4.3	Generování výherce soutěže.....	51
7.5	Ročníky.....	51
7.6	Kola	52
7.6.1	Nominace.....	52
7.6.2	Skupiny.....	53
7.7	Souboje	54
7.8	Hlasování	55
7.9	Statistiky a výpisy.....	56
7.10	Získávání informací o postavách	57
7.11	Generování statistik a soubojů.....	57
8	Vzhled a možnosti uživatele.....	59
8.1	Vzhled.....	59
8.2	Nominace.....	59
8.3	Skupiny a vyřazování	60
8.4	Soutěže a tipy na vítěze	61
	Závěr.....	63
	Literatura	64
	Příloha A – ER-diagram	67

Seznam zkratek

IP	Internet Protocol
NAT	Network Address Translation
HTTP	Hypertext Transfer Protocol
TOR	The Onion Routing
PPC	Pay Per Click
SEO	Search Engine Optimization
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
WYSIWYG	What You See Is What You Get
SQL	Structured Query Language
PL/SQL	Procedural Language/ Structured Query Language
URL	Uniform Resource Locator
SHA	Secure Hash Algorithm
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery

Seznam obrázků

Obr. 1 – Heat mapa kliknutí	19
Obr. 2 – Heat mapa oční kamery	20
Obr. 3 – Získání pozornosti čtenáře.....	21
Obr. 4 – Přehled nejpoužívanějších nástrojů	22
Obr. 5 – Závislost použití analytických nástrojů na návštěvnosti	24
Obr. 6 – Podíl jednotlivých nástrojů v ČR	25
Obr. 7 – Use-case diagram	27
Obr. 8 – Rich picture	28
Obr. 9 – Activity diagram.....	29
Obr. 10 – Vývojový diagram generování statistik vyřazovacích soubojů.....	41
Obr. 11 – Vývojový diagram tvorby vyřazovacích soubojů	43
Obr. 12 – Vývojový diagram tvorby skupin a přiřazení postav	44
Obr. 13 – Vývojový diagram tvorby soubojů ve skupině	46
Obr. 14 – Vzhled aplikace	59
Obr. 15 – Způsoby nominace	60
Obr. 16 – Nominace pomocí seriálu	60
Obr. 17 - Ukázka skupin a soubojů	61
Obr. 18 – Ukázka soutěže.....	62
Obr. 19 – Ukázka tipování.....	62

Úvod

Teoretická část práce je zaměřena na problematiku hlasování na internetu. V první části jsou definovány rozdíly mezi anonymním a neanonymním hlasováním a popsány způsoby, jak je možné tato hlasování ovlivnit a jak se proti tomuto ovlivnění bránit. Druhá část teoretické části je zaměřena na problematiku vyhodnocení návštěvnosti internetových stránek tj. prokliků, návštěv, zobrazení stránek atd. Popsány jsou zde také druhy heat map a způsoby, jak lze tyto mapy vytvářet. V závěru teoretické části jsou vyjmenovány nástroje, které lze využít k webové analytice, jejich funkce, možnosti a výběr nejvhodnějšího nástroje pro náš web.

Druhá část bakalářské práce, praktická část, se zabývá vývojem a popisem aplikace. V praktické části jsou nejdříve popsány požadavky na aplikaci a analyzována celá aplikace. Popsány jsou zde použité technologie včetně příkladů jejich použití. Další část je zaměřena na popis návrhu databáze, kde jsou popsány jednotlivé tabulky v databázi, procedury a funkce, které jsou použity. Následně je popsána samostatná aplikace a popis jejích jednotlivých částí. Závěr praktické části je zaměřen na vzhled aplikace, možnosti uživatelů při používání aplikace a návod na práci s aplikací.

1 Hlasování na internetu

Hlasování na internetu je možné rozdělit na anonymní a neanonymní. U obou hlasování je potřeba řešit problém, jak uživateli zamezit vícenásobné hlasování a jak tohoto uživatele identifikovat. Hlasování je možné naprogramovat individuálně nebo využít již hotového řešení.

1.1 Identifikace uživatele

Existuje několik způsobů, jak identifikovat uživatele. Pokud jsou použity samostatně, není těžké je obejít a hlasování ovlivnit. Pokud je použijeme společně, hlasování se stává bezpečnější. Bohužel neexistuje způsob, jak udělat 100% bezpečné hlasování. Každý ze způsobů identifikace se dá podvrhnout. O uživateli můžeme zjistit:

- IP adresu,
- cookies,
- z HTTP_USER_AGENT prohlížeč a operační systém,
- z HTTP_REFERER adresu stránky, která na aktuální odkázala,
- z HTTP_X_FORWARDED_FOR IP proxy serveru a
- pořadí nainstalovaných rozšíření v prohlížeči. (Kebert, 2002)

Při uložení těchto informací do databáze a následném ověření lze celkem spolehlivě identifikovat uživatele. Stále to nebude 100% řešení, ale útočník bude mít mnohem víc práce při skrytí své pravé totožnosti. Na stránce panopticlick.eff.org lze zjistit, jak je nastavení prohlížeče unikátní.

1.1.1 Příklady identifikace

IP adresa

IP adresu lze získat pomocí PHP velice snadno. Je uložena v globální proměnné `$_SERVER["REMOTE_ADDR"]`.

Ukázka PHP kódu, jak získání IP adresy:

```
<?php
$ip = $_SERVER["REMOTE_ADDR"];
echo $ip;
?>
```

Cookies

Aby bylo možné cookies kontrolovat, musí se nejdříve vytvořit. Cookie se vytváří na straně klienta. V PHP k tomu slouží příkaz `SetCookie (název_cookie, hodnota)`. Cookies jsou uložena v proměnné `$_COOKIE["název_cookie"]`.

Ukázka PHP kódu s uložením a načtením cookie:

```
<?php
SetCookie ("hlas", true);
echo "Cookies jménem hlas nastavena na true";
if($_COOKIE["hlas"]==true){
echo "Nelze hlasovat znovu";
}else{
echo "Váš hlas byl odeslán";
}
?>
```

HTTP_USER_AGENT

Z proměnné `$_SERVER['HTTP_USER_AGENT']` lze zjistit uživatelský prohlížeč a také jeho operační systém. Pro vypsání konkrétního prohlížeče nebo operačního systému je potřeba využít regulárních výrazů nebo využít již hotových řešení pro detekci informací z této hlavičky.

Příklad kódu:

```
<?php
$agent = $_SERVER['HTTP_USER_AGENT'];
echo $a;
?>
```

Hlavička může vypadat např. takto:

```
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.19 (KHTML, like
Gecko) Chrome/18.0.1025.162 Safari/535.19
```

HTTP_REFERER

V proměnné `$_SERVER['HTTP_REFERER']` se nachází adresa předchozí stránky, odkud uživatel přišel. Pokud uživatel hlasoval a předchozí stránka není ta, na které se anketa nachází, tak se jedná o podezřelý hlas.

Příklad HTTP_REFERER:

```
<?php
$adresa = $_SERVER['HTTP_REFERER'];
echo "Uživatel přišel ze stránky " . $adresa;
?>
```

HTTP_X_FORWARDED_FOR

Pokud je v \$REMOTE_ADDR IP adresa proxy serveru, tak v \$HTTP_X_FORWARDED_FOR může být adresa počítače, který na proxy server odeslal požadavek.

Příklad HTTP_X_FORWARDED_FOR:

```
<?php
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
echo "Uživatel přišel přes proxy server a jeho adresa je " . $ip;
?>
```

Pluginy v prohlížeči

K detekci pluginů v prohlížeči se používá JavaScript. Je možné použít např. script PluginDetect (Gerds, 2012), který vypíše všechny pluginy v prohlížeči.

Ukázka začátku výpisu tohoto scriptu:

```
Plugin 0: Adobe Acrobat; Adobe PDF Plug-In For Firefox and Netscape
10.1.3; nppdf32.dll; (Acrobat Portable Document Format; application/pdf;
pdf) (Adobe PDF in XML Format; application/vnd.adobe.pdfxml; pdfxml)
(Adobe PDF in XML Format; application/vnd.adobe.x-mars; mars) (Acrobat
Forms Data Format; application/vnd.fdf; fdf) (XML Version of Acrobat
Forms Data Format; application/vnd.adobe.xfdf; xfdf) (Acrobat XML Data
Package; application/vnd.adobe.xdp+xml; xdp) (Adobe FormFlow99 Data File;
application/vnd.adobe.xfd+xml; xfd). Plugin 1: Chrome PDF Viewer; ;
pdf.dll;
```

1.2 Neanonymní hlasování

Neanonymní hlasování je takové hlasování, kdy je známa pravá totožnost hlasujícího uživatele. Zpravidla je vyžadována registrace uživatele, kde uživatel vyplní minimálně svoji mailovou adresu. Dále to může být jméno a příjmení, adresa, telefonní číslo, uživatelské jméno atd. S registrací si autor hlasování může zjistit o uživateli údaje uvedené v kapitole 1.1. Každý hlas v anketě patří konkrétnímu registrovanému uživateli a není možné, aby tento uživatel mohl hlasovat několikrát, pokud mu to autor hlasování nedovolí.

1.2.1 Vlastní registrace a přihlášení

Problém nastává v případě, když si uživatel udělá novou registraci pod jiným jménem. Ověřit pravost jména je nemožné, ale lze ověřit mailovou adresu. Mailová adresa, kterou uživatel zadává, musí být jedinečná. Nemůže se stát, aby dva uživatelé v systému měli stejnou mailovou adresu. Mailovou adresu lze ověřit pomocí mailu s aktivačním odkazem, který obsahuje např. přihlašovací jméno a mail uživatele v metodě GET. Po kliknutí na tento odkaz se daný účet, který je propojen s touto mailovou adresou, aktivuje a tím je mail ověřený a patří uživateli, který provedl registraci.

Odkaz v aktivačním mailu může vypadat například takto:
<<http://webovestranky.cz/dokoncitreg.php?id=1254785>>. Hodnota id může být

zaheshované uživatelské jméno a mail. Pokud uživatel na tento odkaz klikne, přejde na stránku dokoncitreg.php. Na této stránce získá zaheshované uživatelské jméno a mail, a ověří se s hodnotami v databázi. Pokud jsou v pořádku, daný účet se aktivuje.

Toto řešení nezabrání uživateli udělat novou registraci, ale je časově náročnější. Zamezit uživateli, aby si udělal více registrací, je možné stejným způsobem, jako se kontroluje vícenásobné hlasování u anonymního hlasování.

1.2.2 Přihlášení přes Facebook

Další možností je nedělat vlastní registraci a nechat uživatele přihlásit se prostřednictvím svého Facebook účtu. Jako u vlastní registrace nejde zamezit, aby si útočník udělal nový Facebook účet, ale je to další komplikace, která případný útok znepříjemní.

Implementace Facebook přihlášení je možné udělat pomocí PHP nebo javascriptu.

Implementace pomocí PHP probíhá v těchto krocích:

- stažení Facebook PHP SDK,
- registrace webové stránky na Facebooku a získání appId a secretId,
- vložení SDK do aplikace,
- vytvoření třídy a předání jí appId a secretId,
- vygenerování přihlašovacího linku,
- požádání o login URL a
- SDK uloží do session aktuálního uživatele. (Zatomik, 2012)

1.3 Anonymní hlasování

Anonymní hlasování znamená, že před hlasováním není vyžadována registrace uživatele. Jediné informace o uživateli jsou ty, které získáme z HTTP hlaviček. K ověření uživatele se musí použít kombinace více metod, protože pokud se jednotlivé metody použijí samostatně, je toto zabezpečení lehce prolomitelné.

Při zabezpečení například pouze pomocí cookies, uživateli nic nebrání v tom ho vymazat. Někteří uživatelé mají cookies trvale zakázané. V případě IP adresy může být problémem shodná IP adresa více uživatelů. Jedná se buď o neveřejnou adresu, nebo o více počítačů za NATem. Pokud toto nastane, hlasovat může pouze jeden uživatel a ostatní uživatelé nemohou, protože jejich IP adresa je již v systému zaznamenána.

Je také možné využít, stejně jako u neanonymního hlasování, kontrolu prostřednictvím mailu a aktivačního odkazu. Hlas v anketě se stane aktivní až poté, co je potvrzen přes aktivační odkaz. (Lahvička, 2000)

Aby bylo anonymní hlasování bezpečné a bylo možné zamezit vícenásobnému hlasování, musí se použít kombinace více metod ověření. Algoritmus může vypadat např. takto: Automaticky se vyloučí hlasy ze stejné cookies a mailové adresy. Spočítá se zastoupení IP

adresy, ze které hlas přišel, u dané odpovědi a mezi všemi hlasy. „Čím vyšší je zastoupení dané IP adresy u dané varianty vůči zastoupení u ostatních variant, tím je hlas podezřelejší (předpokládá se, že podvodník bude mít zájem zvýhodnit svým hlasováním jednu variantu vůči ostatním). Podobně lze podezřelost určit i na základě hlavičky user-agent a na základě části e-mailu za zavináčem.“ (Lahvička, 2000) Podle vypočtené podezřelosti se stanoví časový rozdíl potřebný mezi jednotlivými hlasy nebo je hlas rovnou odmítnut.

Jak složitý algoritmus lze pro hlasování použít, záleží na tom, jak dobře zabezpečené má být hlasování. Jedná-li se o informativní hlasování na soukromém webu, tak stačí, pokud je použita kontrola např. prostřednictvím IP adresy a cookies. Anketa bude sice relativně lehce ovlivnitelná, ale následky z toho nejsou téměř žádné. Pokud se ale bude jednat o soutěž, tak je pravděpodobné, že se hlasování někdo pokusí ovlivnit. Čím hodnotnější bude výhra, tím větší zabezpečení je potřeba nastavit.

Cílem těchto kontrol je vytvořit co nejkomplikovanější možnost opakovaného hlasování, která by nestála za jejich obejití.

Ukázka kontroly podle cookies, IP adresy a hlavičky X-Forwarded-For: Převzato z (Vrána, 2006)

```
<?php
if (isset($_COOKIE["hlasoval"])) {
    echo "Váš hlas byl již dříve započten.\n";
} else {
    setcookie("hlasoval", $_POST["hlas"], strtotime("+1 month"));
    $forwarded_for = addslashes(preg_replace('~.*\\s*~', '',
$_SERVER["HTTP_X_FORWARDED_FOR"]));
    if (mysql_result(mysql_query("SELECT COUNT(*) FROM anketa_hlasy WHERE
ip = '$_SERVER[REMOTE_ADDR]' AND forwarded_for = '$forwarded_for' AND
datum + INTERVAL 1 HOUR > NOW()"), 0)
|| mysql_result(mysql_query("SELECT COUNT(*) FROM anketa_hlasy WHERE
ip = '$_SERVER[REMOTE_ADDR]' AND datum + INTERVAL 1 HOUR > NOW()"), 0) >=
5) {
        echo "Hlas z této adresy byl již dříve započten.\n";
    } else {
        mysql_query("INSERT INTO anketa_hlasy (hlas, ip, forwarded_for,
datum) VALUES ('$_POST[hlas]', '$_SERVER[REMOTE_ADDR]', '$forwarded_for',
NOW())");
    }
}
?>
```


Kontrola v tomto případě probíhá ve třech krocích:

- pokud existuje cookies, tak hlas nezapočítat,
 - pokud této IP adresy v kombinaci s X-Forwarded-For bylo tuto hodinu hlasováno, tak hlas nezapočítat a
 - pokud z této IP adresy přišlo v poslední hodině 5 hlasů, tak hlas nezapočítat.
- (Vrána, 2006)

Takto nastavená kontrola je již celkem spolehlivá. V omezené míře lze hlasovat několikrát z dynamické IP adresy. (Vrána, 2006)

1.4 Možnosti ovlivnění hlasování

Pokud hlasování využívá metody GET, je možné toto hlasování ovlivnit pomocí vlastní internetové stránky. Do vlastní webové stránky se vloží odkaz s odesláním odpovědi jako src obrázek.

```

```

Obrázek bude na stránce neviditelný a každý návštěvník stránky udělí jeden hlas pro požadovanou odpověď. Tomuto lze zabránit kontrolováním hlavičky HTTP_REFERER nebo místo metody GET použít metodu POST. (Hulán, 2007)

Pokud je hlasování založeno na unikátnosti IP adresy, musí útočník použít veřejný proxy server. Seznam proxy serveru lze nalézt třeba na této stránce www.checkedproxylists.com. Proxy server je prostředník mezi serverem a klientem. Přijímá požadavky od klienta a předává je serveru, vůči kterému vystupuje jako klient. Odpovědi serveru posílá zpět klientovi. (Wikimedia Foundation Inc., 2012) Takové hlasy budou pokaždé z jiné IP adresy. „Pokud provoz prochází přes proxy server, bývá doplněna hlavička *X_FORWARDED_FOR*“. (Vrána, 2006) Takto lze poznat, že byl použit proxy server. Problém je, že tato hlavička se dá snadno podvrhnout.

Změnu proxy serveru lze provést v nastavení každého prohlížeče v kategorii sítě, připojení aj. Je potřeba znát adresu proxy serveru a port serveru. Po nastavení těchto dvou údajů se prohlížeč bude připojovat přes proxy server a v internetu bude vystupovat, jako kdyby byl zadán proxy server.

Další možností jak obejít ochranu unikátní IP adresy je použití technologie TOR. Tato technologie slouží k anonymitě na internetu a je snadno zneužitelná pro tyto druhy útoků. Odesílaná data jsou obalena vrstvami. Každá vrstva je zašifrována jiným klíčem a obsahuje cestu k dalšímu TOR routeru. TOR router dešifruje jednu vrstvu a zjistí kam data dále odeslat. Až zbydou pouze data, poslední TOR router je odešle k požadovanému serveru. Server bude mít informace o posledním TOR routeru a ne o původním odesílateli.

(Čížek, 2009) Takto lze hlasovat několikanásobně, protože IP adresa se bude měnit. Obranou je kontrolovat IP adresu zda není na seznamu TOR adres. (Vrána, 2009)

K použití technologie TOR je potřeba nainstalovat do počítače klienta. Klient je zároveň i TOR router, takže i přes tento počítač bude procházet TOR provoz. V nastavení klienta lze zvolit, aby tento počítač nebyl posledním článkem a nevstupovalo se přes něj do internetu. Pokud je nainstalovaný a spuštěný klient, tak se odchozí pakety zabalí do „slupek“ a jsou nejdříve poslány do sítě TOR. Problém tohoto připojení je, že je pomalé. Kdokoli může být TOR router a jeho připojení nemusí být tak kvalitní, což bude mít za následek pomalý průchod přes TOR router a tím pomalou i celou síť.

1.5 Ankety třetích stran

Anketu není nutné programovat. Je možné si ji nechat vygenerovat a do vlastní webové stránky vložit jenom HTML kód a o obsluhu a zabezpečení se stará poskytovatel ankety. Ani tohle řešení 100% nezaručí bezpečné hlasování a ani sám poskytovatel ankety to nemůže garantovat.

Nejznámější řešení na českém internetu je pravděpodobně od společnosti <blueboard.cz>. Tyto ankety jsou zabezpečeny IP kontrolou a cookies. (Blueboard.cz)

Další možností je stáhnutí již vyhotovených scriptů, HTML kódu a PHP tříd a vložení do webové stránky.

2 Vyhodnocení návštěvnosti

Jedná se o měření, sběr, sledování a vyhodnocování dat za účelem porozumění a optimalizace webu. Lze získat dva druhy informací. Prvním druhem je, co se na webových stránkách děje a druhým, jaké úpravy stránek by pomohly k tomu, aby byly efektivnější. (Wikimedia Foundation Inc., 2012) Díky těmto informacím je možné vyhodnotit např. úspěšnost PPC kampaně nebo SEO.

2.1 Používané termíny

Ve vyhodnocování návštěvnosti se lze setkat s několika termíny, které se tváří jako jasné definovaná a podobná. Jsou ale mezi nimi rozdíly a ve webové analytice mohou znamenat něco jiného, než znamenají normálně.

Prokliky

„Jako proklik se označuje takové kliknutí na odkaz, které uživatele dovede na cílovou stránku.“ (Adaptic, s.r.o.) Po kliknutí na odkaz se může stát, že se uživatel na cílovou stránku nedostane z důvodů jako např. stránka nefunguje, uživatel si to rozmyslí a načítání stránky ukončí nebo načítání stránky trvá příliš dlouho. Proto každé kliknutí nemusí být automaticky i proklik. (Adaptic, s.r.o.)

Tento termín se nejčastěji vyskytuje v oblasti internetové reklamy. *„V této souvislosti bývá odhadováno, že asi 5-10 % kliknutí na reklamu se k cílovému serveru nedostane.“* (Adaptic, s.r.o.) V internetové reklamě se měří míra prokliků, což je poměr mezi počtu prokliků a zobrazení reklamy. (Adaptic, s.r.o.)

Často bývají prokliky a normální kliknutí zaměňovány, protože měřit kliknutí je mnohem jednodušší než proklik. U PPC reklamy, která je často prezentována jako platba za proklik, se často měří pouze počet kliknutí na reklamu. (Adaptic, s.r.o.)

Návštěva

Jedná se o počet jedinečných relací s návštěvníky stránek. Relace má nastavenou platnost, takže po vypršení platnosti se musí vytvořit nová relace a tím vznikne nová návštěva. Pokud např. návštěvník klikne na reklamu vícekrát během stejné relace, jedná se o jednu návštěvu, ale o více prokliků. Dalším příkladem může být situace, kdy uživatel klikne na reklamu a později během jiné relace se vrátí přímo na stránku kliknutím na záložku. V tomto případě se bude jednat o jeden proklik a více návštěv. (Google Inc. , 2012)

Návštěvníci

Jeden návštěvník se nerovná jeden člověk. Spíše platí, že jeden návštěvník se rovná konkrétní prohlížeč na konkrétním počítači. Takže všechny návštěvy z jednoho prohlížeče jsou brány jako jeden návštěvník. Návštěvníci jsou sčítáni za zvolené období. Pokud je zobrazení po dnech, pak je jeden návštěvník započítán pod každý den, kdy web navštívil. Při měsíčním pohledu je započítán pouze jednou za celý měsíc. (Weida, 2011)

Zobrazení a unikátní zobrazení

Zobrazení je definováno jako jedno zobrazení jedné webové stránky. Pokud uživatel na načtené stránce stiskne tlačítko pro opětovné načtení, je to počítáno jako další zobrazení webové stránky. Přejde-li uživatel na jinou stránku a poté se vrátí zpět, je to také počítáno jako další zobrazení.

Unikátní zobrazení je takové zobrazení stránky, které bylo generováno jedním uživatelem během jedné relace. (Google Inc. , 2012) Jeden uživatel může tedy vygenerovat jedno unikátní zobrazení během relace.

Konverzní poměr

Konverzní poměr můžeme také označit jako míra konverze. Tato hodnota je sledována především u elektronických obchodů. Konverzní poměr se spočítá jako podíl počtu konverzí a počtu návštěvníků. Počet konverzí může být např. počet objednávek. (Wikimedia Foundation Inc., 2011)

2.2 Heat mapa kliknutí

Heat mapa kliknutí je vrstva na webové stránce, na které je vidět kam a jak intenzivně uživatelé klikají. Nejlépe heat mapu popíše obrázek (Obr. 1).



Obr. 1 – Heat mapa kliknutí

Zdroj: (Plotěný, 2009)

Mapu lze využít k:

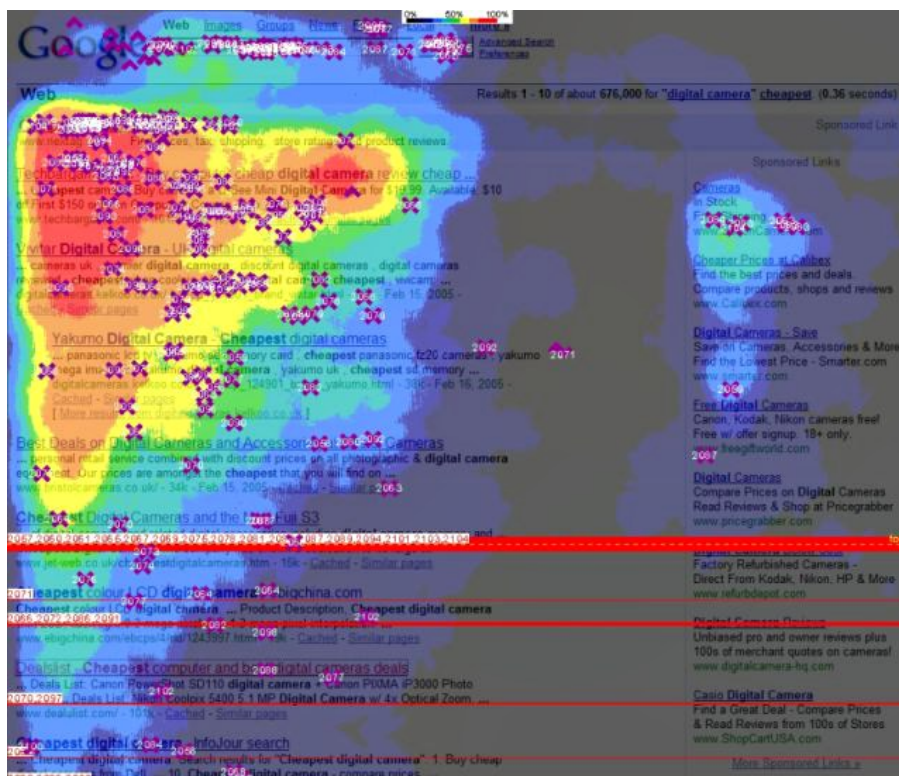
- odhalení přehlížených aktivních prvků na stránce,
- nalezení klíčových prvků vyzývající k akci a
- identifikaci chybějících odkazů. (Plotěný, 2009)

Heat mapu není nutné vytvářet pro každou stránku na webu. Stačí ji znát pro klíčové stránky jako je hlavní strana, stránka produktu nebo stránka s objednávacím formulářem. Ideálním postupem je získat výsledky, stránku podle nich upravit a znovu získat nové výsledky. (Plotěný, 2009)

Implementace je jednoduchá. Stačí pouhá registrace stránky u poskytovatele služby, jako je třeba <crazyegg.com> nebo <clickdensity.com>. Službu je možné využít ve trial verzi nebo za poplatek \$9 měsíčně v případě Crazyegg. Po registraci je nutné do stránky vložit měřicí kód. (Plotěný, 2009)

2.3 Heat mapa oční kamery

Tato služba má podobný výstup jako heat mapa kliknutí, ale jedná se o něco jiného. Tato mapa udává, kam se uživatel na stránce dívá a jaké prvky ho nejvíce zajímají. Na obrázku (Obr. 2) je výsledek hledání ve vyhledávači Google.



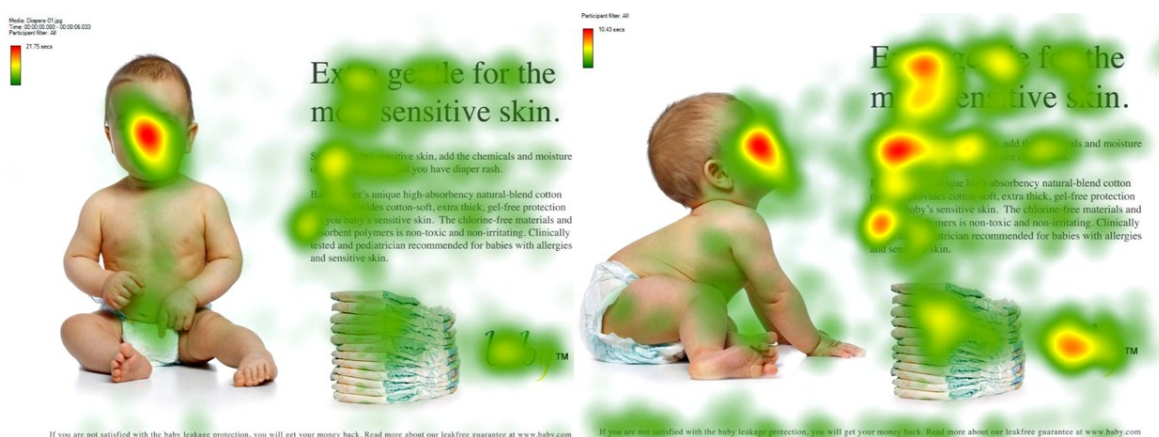
Obr. 2 – Heat mapa oční kamery

Zdroj: (Dobiáš, 2011)

Co nejvíce přitahuje pozornost uživatele v textu:

- titulky,
- podnadpisy,
- tučné písmo,
- hypertextové odkazy,
- obrázky a jejich podtitulky,
- odrážky a
- seznamy. (Dobiáš, 2011)

Pomocí fotografie je možné upoutat čtenářův zrak na konkrétní článek nebo produkt. Pokud je na stránce postava, která se dívá na text, čtenáři se na text podívají také. Na obrázku (Obr. 3) se dítě dívá na čtenáře a největší pozornost čtenáře je zaměřena na obličej. Pokud se ale dítě dívá na článek, tak článek upoutá stejnou pozorností jako obličej dítěte. (Dobiáš, 2011)



Obr. 3 – Získání pozornosti čtenáře

Zdroj: (Dobiáš, 2011)

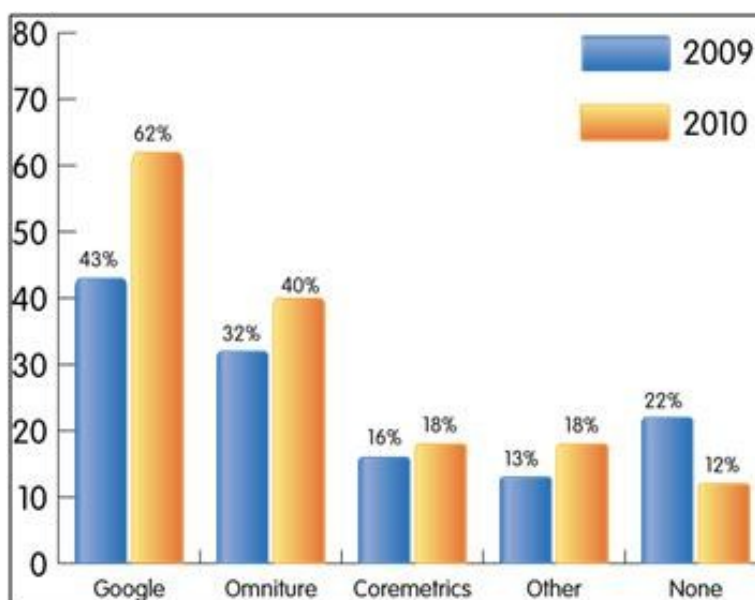
3 Nástroje pro vyhodnocení návštěvnosti

Návštěvnost se může vyhodnocovat pomocí vlastnoručně napsaného kódu, ale je to zbytečné. Na trhu je velké množství nástrojů, které lze použít. Je možné kombinovat nástroje od různých výrobců a tím docílit požadované funkcionality.

3.1 Nejpoužívanější nástroje webové analytiky

Na obrázku (Obr. 4) jsou vidět tři nejpoužívanější analytické nástroje podle studie „*Analytics Market analysis*“ (iPerceptions inc., 2010). Jsou jimi:

- Google Analytics,
- Omniture a
- Coremetrics.



Obr. 4 – Přehled nejpoužívanějších nástrojů

(iPerceptions inc., 2010)

České analytické nástroje jsou např.:

- Toplist.cz,
- eMerito,
- Navrchlu.cz a
- NetMonitor.

Google Analytics

Google Analytics je od společnosti Google Inc. a používání služby je zdarma. Tento nástroj má na trhu dominantní postavení, které potvrzuje i výzkum z roku 2010 zveřejněný na webu < Istobe.com>. Z obrázku (Obr. 5) je patrné, že se Google Analytics využívá u méně rozsáhlých a navštěvovaných internetových stránek. Čím jsou stránky rozsáhlejší a navštěvovanější, tím více jsou nasazovány propracovanější analytické nástroje. U internetových stránek s měsíční návštěvností více jak 5 miliónů unikátních návštěvníků je použití Google Analytics menší než 15%. (Čech, 2010)

Výhodou Google Analytics je snadné použití a především snadná implementace. Implementace je možná např. vložením vygenerovaného scriptu do webových stránek.

Omniure

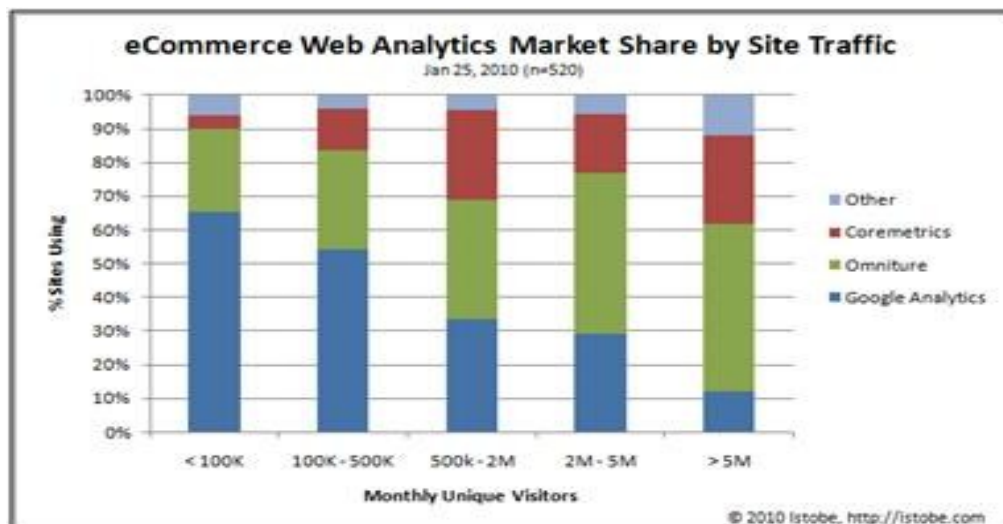
Omniure patří společnosti Adobe Systems Incorporated, která ho koupila v roce 2009 přibližně za 1.8 miliard dolarů. (Adobe Systems Incorporated, 2009) Jedná se o komerční řešení. Na obrázku (Obr. 5) Obr. 5 je vidět, že se jedná o nástroj, který využívají rozsáhlé a hodně navštěvované internetové stránky. U stránek s návštěvností přes 5 miliónů je Omniure nejpoužívanějším řešením. (Čech, 2010)

Podle společnosti Optimics s.r.o. (Optimics s.r.o) jsou hlavní výhody:

- rozsáhlé možnosti customizace,
- integrace s dalšími firemními systémy,
- pokročilá funkcionalita a
- garance dat a technická podpora.

Coremetrics

Coremetrics patří společnosti International Business Machines Corp. (IBM). Stejně jako Omniure se používá pro rozsáhlé internetové stránky s vysokou návštěvností. Na obrázku (Obr. 5) je vidět, že se používá především jako doplněk ostatních řešení, například společně s Omniure.



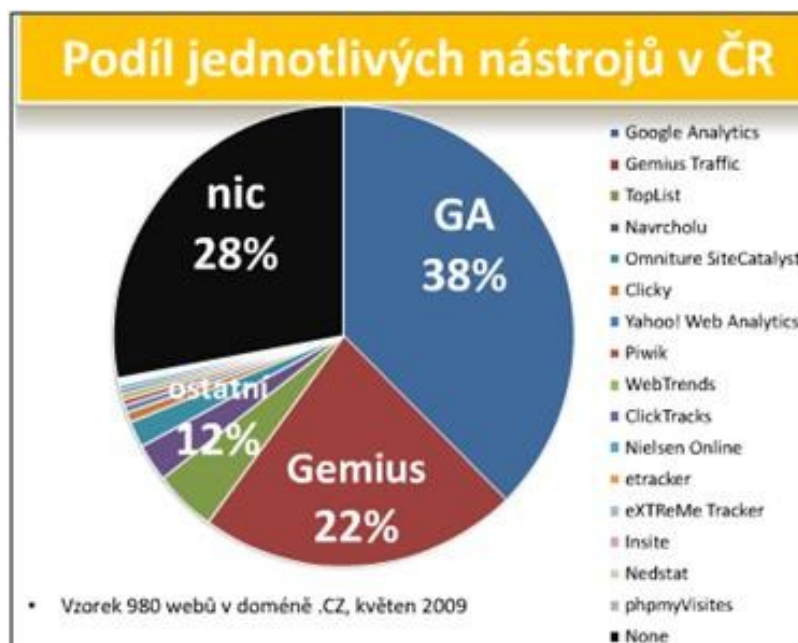
Obr. 5 – Závislost použití analytických nástrojů na návštěvnosti

(Bright, 2010)

3.2 Nástroje webové analytiky v České republice

Výsledky statistického měření za období květen 2009, které bylo provedeno na vzorku 980 webů v doméně CZ, jsou na obrázku (Obr. 6). Z něho vyplývá:

- nejpoužívanější je Google Analytics s 38%,
- silný podíl má české řešení NetMonitor,
- další česká řešení obsadila 3 a 4 místo, ale v součtu mají pouze 7% podíl,
- minimální použití světově nejpoužívanějších řešení jako Omniture a Coremetrics a
- velký podíl webů, které nepoužívají žádná řešení. (Čech, 2010)



Obr. 6 – Podíl jednotlivých nástrojů v ČR

(Čech, 2010)

3.3 Postup při výběru analytického řešení

Odborníci v oblasti webové analytiky Avinsh Kaushik a Evam LaPointe doporučují při výběru vhodného nástroje postupovat v následujících krocích:

- zjištění potřeb firmy a požadavků na analýzu webu,
- zjištění i potřeb jednotlivých pracovníků firmy, pokud tato data potřebují,
- výběr vhodného řešení podle zjištěných potřeb a požadavků,
- návrh základní implementace,
- implementace podle návrhu,
- ladění a customizace nástroje,
- školení zaměstnanců, kteří budou s nástrojem pracovat,
- využívání analytiky a
- případná implementace chybějících funkcionalit nebo propojení s dalšími firemními systémy. (Čech, 2010)

Tento postup lze uplatnit při výběru analytického řešení pro větší společnost. Pro menší společnosti nebo majitele malých webových stránek je důležitá hlavně cena a náročnost na implementaci do stávajícího řešení. Tyto dva požadavky perfektně splňuje Google Analytics a proto je také nejrozšířenější.

4 Analýza webové aplikace

Analýzu aplikace je možné rozdělit do několika částí. Tvůrce aplikace musí vědět, co od ní očekává, a jaké má mít funkce a možnosti. Také je důležité si uvědomit, co bude od aplikace očekávat uživatel a jak s ní bude pracovat.

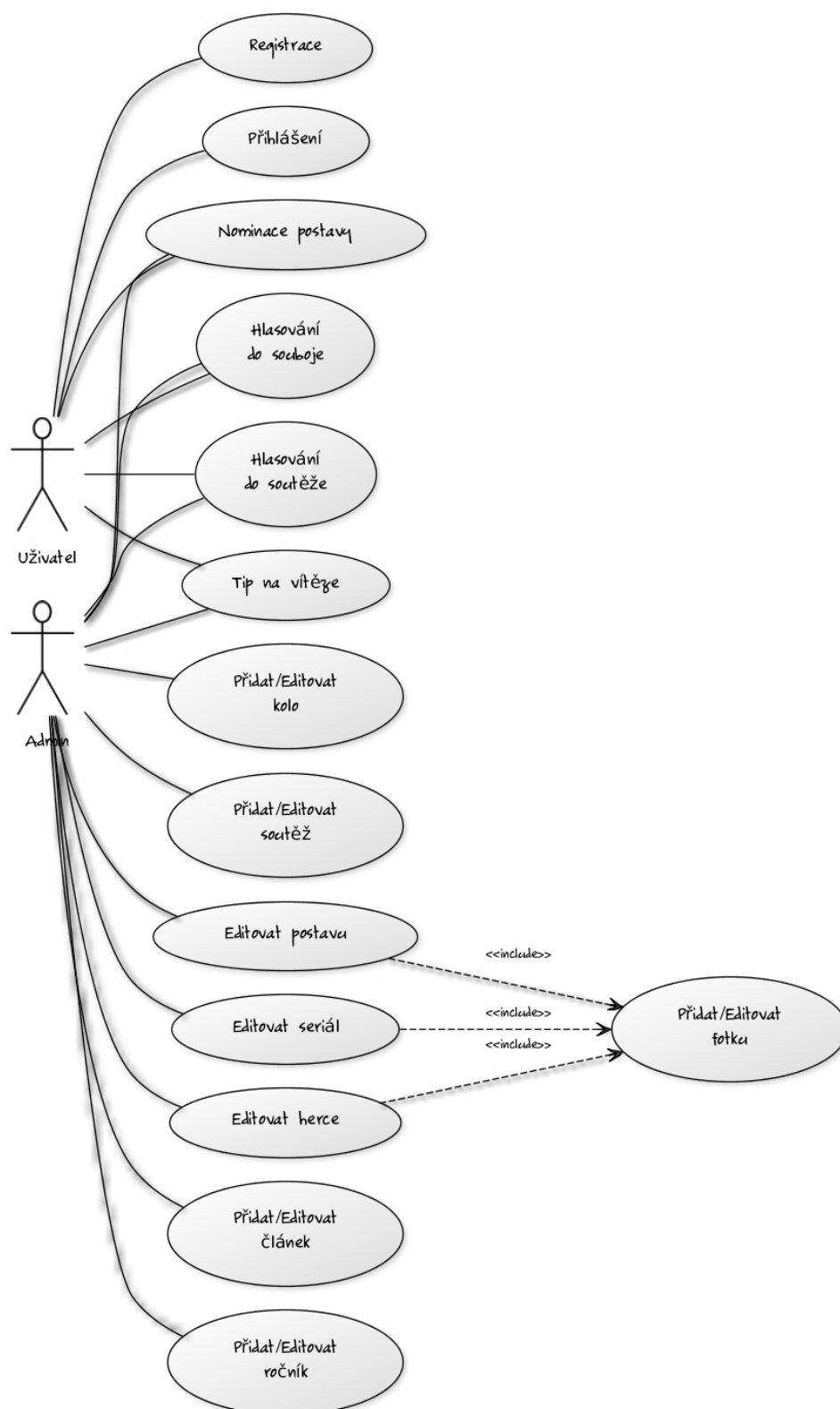
4.1 Požadavky

Aplikace by měla splňovat tyto požadavky:

- vícekolový průběh,
- registrace a přihlášení uživatele,
- možnost hlasování dělit do ročníků,
- bezúdržbový běh celého ročníku,
- bezpečné hlasování,
- tvorbu soutěží,
- přehlednost a
- intuitivnost.

4.2 Use-case diagram

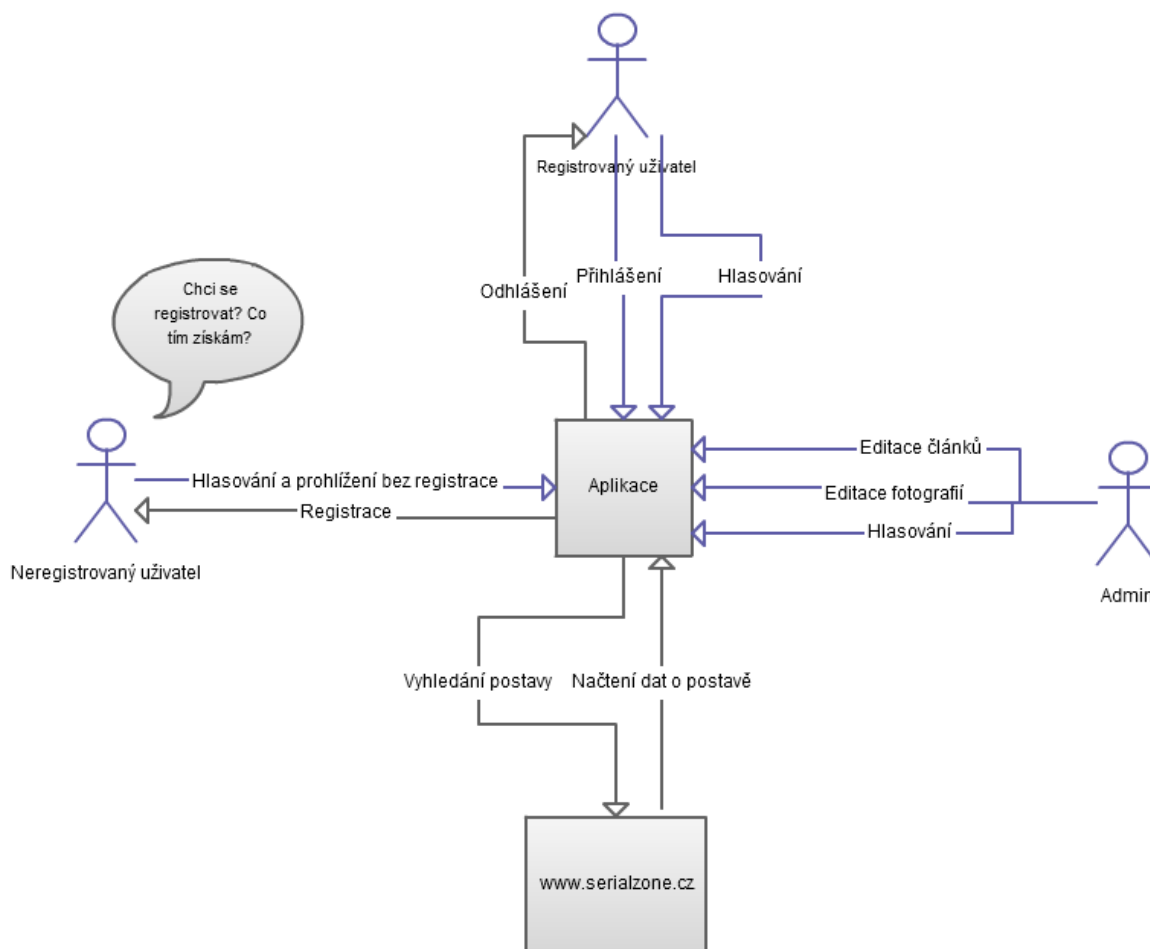
Use-case diagram (Obr. 7) zobrazuje chování celého systému z hlediska jednotlivých uživatelů. Diagram je rozdělen z pohledu uživatele a administrátora. Je vidět, že administrátor může dělat to samé co obyčejný uživatel. Navíc má na starost chod celého systému. Diagram je vytvořen pomocí online nástroje nacházejícího se na stránce <yuml.me>.



Obr. 7 – Use-case diagram

4.3 Rich-picture

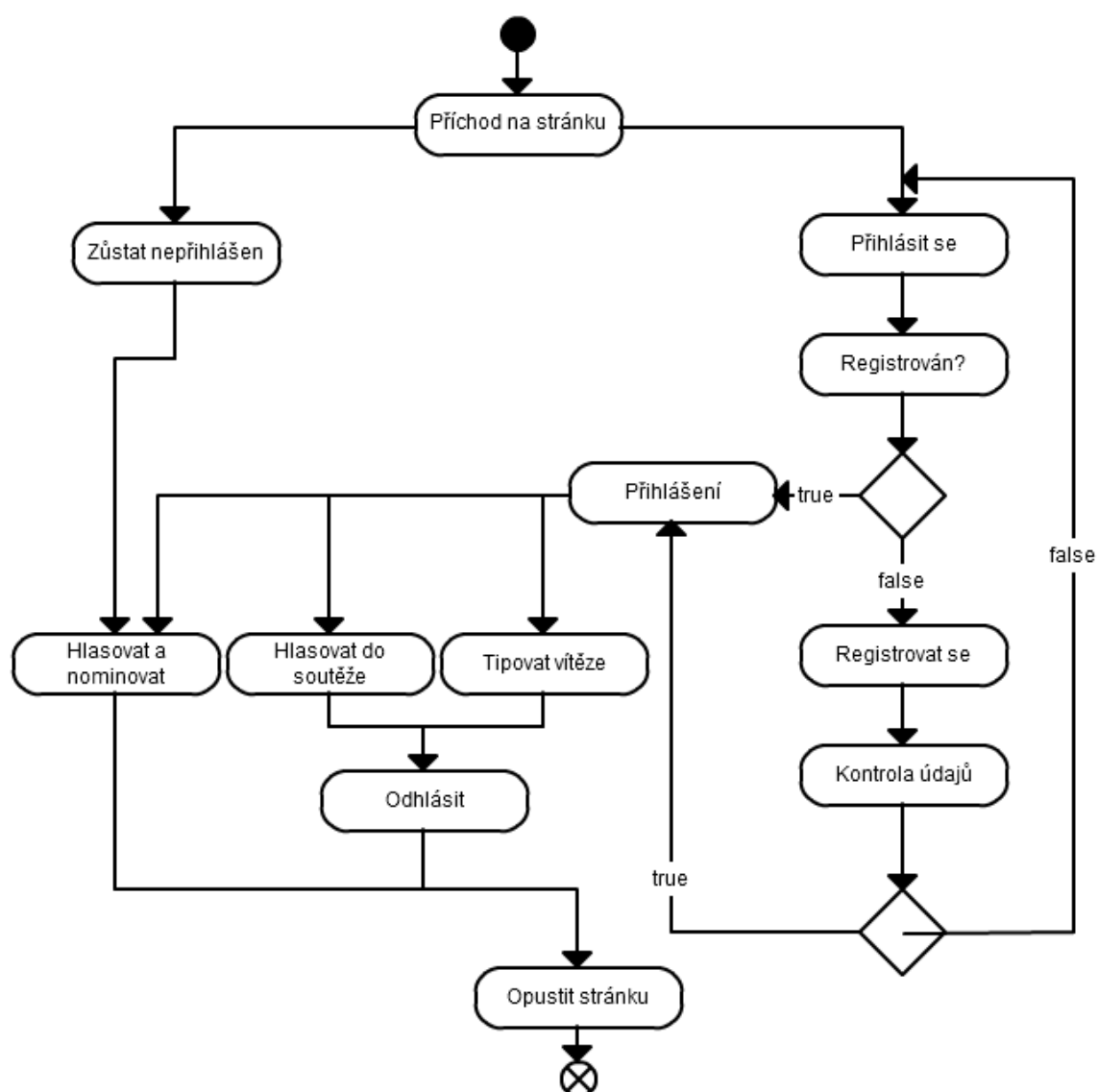
Pomocí diagramu rich picture se načrtne základní běh aplikace. Díky tomu jsou ihned vidět základní informace a otestuje se reálnost návrhu. Také se identifikují a vymezí hranice systému. V rich picture by se neměla objevovat definice systému a konkrétní implementace. Tento diagram nemá žádnou formální definici. Diagram je vytvořen pomocí online nástroje na stránce <creately.com>.



Obr. 8 – Rich picture

4.4 Activity diagram

Activity diagram (Obr. 9) slouží k pochopení chování jednotlivých uživatelů na stránce. Díky tomuto diagramu je vidět, jak se uživatel chová. Uživatelé je možné rozdělit do dvou kategorií; uživatele přihlášené a nepřihlášené. Rozdíl mezi nimi je v možnostech. Přihlášení uživatelé mohou navíc hlasovat do soutěží a tipovat vítěze. Diagram je vytvořen pomocí online nástroje na stránce <creately.com>.



Obr. 9 – Activity diagram

Uživatel, který přijde na stránku, má dvě možnosti. Může se přihlásit, nebo zůstat jako nepřihlášený uživatel. Přihlásit se může jenom v tom případě, je-li na webové stránce registrován. Přihlášený uživatel může proti nepřihlášenému soutěžit ve vypsáných soutěžích a tipovat postavy na vítězství. Před opuštěním stránky se může uživatel odhlásit.

5 Použité technologie

Aplikace je postavena na technologii HTML a CSS. Dále je v aplikaci použit jazyk PHP a JavaScript. Databáze je od společnosti Oracle, konkrétně se jedná o Oracle Database 10g. V aplikaci jsou použity prvky z Nette Framework. Pro HTML parsování je použit PHP Simple HTML DOM Parser.

5.1 HTML

Jedná se o jazyk, který se používá k tvorbě webových stránek. Je definován organizací W3C. Jazyk HTML se skládá z tagů a jejich atributů. Používá se k zobrazení informací pro uživatele a výsledkem je jednoduchý text se základními prvky grafiky. Aby bylo možné výslednou stránku lépe graficky upravit, je nutné použít např. CSS styly. Soubory s HTML kódem mají koncovku .htm nebo .html.

Příklad HTML kódu:

```
<!doctype html>
<html>
  <!-- Takto lze zapsat komentář, který se na stránce nezobrazí -->
  <head>
    <meta charset="kódování">
    <title>Titulek stránky</title>
  </head>
  <!-- tělo dokumentu -->
  <body>
    <h1>Nadpis stránky</h1>
    <p>Tělo dokumentu</p>
  </body>
</html>
```

Stránka se v internetovém prohlížeči zobrazí takto:

Nadpis stránky

Tělo dokumentu

Aplikace je napsána ve verzi HTML 5, které je nejnovější. Hlavní výhody této verze jsou lepší strukturovanost jazyka a lepší podpora videa a audia, které jsou na dnešním internetu velice populární.

5.2 CSS

CSS neboli kaskádové styly slouží ke grafické úpravě webových stránek. CSS nelze použít bez znalosti HTML jazyka. CSS se do českého jazyka překládá jako „kaskádové styly“. Kaskádové protože lze jednotlivé vlastnosti na sebe vrstvit a vždy platí jenom ta poslední. CSS je jako HTML definováno organizací W3C. Velkou výhodou CSS je možnost oddělení vzhledu dokumentu od jeho struktury.

Příklad CSS kódu:

```
body {  
  background-color: white;  
  color: blue;  
  padding: 10px;  
  margin: 10px 30px auto 30px;  
}
```

CSS definuje několik selektorů. Selektory rozhodují, kterých prvků se pravidla týkají. Mezi nejpoužívanější patří:

- prvek,
- potomek,
- třída a
- identifikátor.

5.3 PHP

Jedná se o skriptovací programovací jazyk, který se používá pro programování dynamických webových stránek. PHP je prováděno na straně serveru, tudíž je potřeba podpora z jeho strany. Uživatel uvidí pouze výsledek daného kódu aplikace.

Ukázka PHP kódu:

```
<?php  
  $promenna = "ahoj, světe!";  
  echo $promenna;  
?>
```

PHP je klasický programovací jazyk, jako je například jazyk C nebo Java. Lze v něm používat podmínky, cykly, pole atd.

V aplikaci je použito PHP verze 5.3.

5.4 JavaScript

JavaScript je programovací jazyk, který se zapisuje přímo do HTML. Na rozdíl od PHP se jedná o jazyk, který se provádí na straně klienta. Často je JavaScript zaměňován s Javou, ale Java je samostatný programovací jazyk. Mají pouze podobnou syntaxi. (Janovský, 2012)

Ukázka kódu javascriptu:

```
<script>  
document.write("Hello world");  
</script>
```

V aplikaci je JavaScript použit pro implementaci WYSIWYG editoru.

5.5 Relační databáze

Databáze je postavena na řešení od firmy Oracle. Konkrétně na Oracle Database 10g. S databází se pracuje pomocí jazyka SQL a je možné využít jazyk PL/SQL.

SQL

Jazyk SQL slouží k:

- definici dat,
- získávání dat,
- manipulaci s daty,
- řízení přístupu,
- sdílení dat a
- k dodržení integrity dat. (Žák, 2011)

Aplikace pomocí jazyka SQL komunikuje s databází a pracuje s ní. Příkazy jsou na webové stránce zadávány pomocí jazyka PHP.

PL/SQL

PL/SQL je procedurální nástavbu jazyka SQL od společnosti Oracle. Pomocí jazyka PL/SQL je možné využít konstrukce procedurálního programování. Lze vytvářet podmínky, cykly, kurzory atd. PL/SQL se skládá z bloků.

Příklad struktury bloku:

```
DECLARE
/* Deklarace obsahuje proměnné, typy a lokální subprogramy. */
BEGIN
/* Výkonná sekce */
/* Toto sekce je povinná */
EXCEPTION
/* Zde se zpracovávají chybové události. */
END;
```

V PL/SQL se nejčastěji vytvářejí procedury, které jsou pak volány. Další často používanou možností je vytvoření funkce, která se od procedury liší tím, že vrací hodnotu.

5.6 Ostatní

V aplikaci jsou také využity prvky z následujících technologií.

Nette Framework

Framework je open source a používá se pro tvorbu webových aplikací v PHP 5. Ušetří práci při psaní opakujícího se kódu. „*Framework vám ulehčí práci, budete méně psát, mít přehlednější kód a radost z práce*“. (Nette Foundation, 2012)

Výhody použití Nette:

- šablonovací systém,
- ladící nástroje,
- efektivní databázová vrstva,
- zabezpečení před zranitelnostmi,
- podpora HTML5, AJAX a SEO a
- kvalitní česká dokumentace. (Nette Foundation, 2012)

V aplikaci je Nette implementováno jen částečně a používá se pro tvorbu a obsluhu formulářů.

PHP Simple HTML DOM Parser

Pomocí tohoto nástroje lze stáhnout HTML kód ze zadané stránky a následně vybrat požadované informace či celé tagy. Nástroj je napsaný v PHP. Pro potřeby této aplikace toto řešení vyhovuje. Parser je poměrně rozsáhlý a má spoustu funkcí, ale práce s ním je velice intuitivní a snadná.

6 Návrh a tvorba databáze

Kvalitně navržená databáze je prvním důležitým krokem při vývoji aplikace, která svá data do databáze ukládá. Databáze je navržena ve formě fyzického modelu a splňuje třetí normální formu (3NF). Další důležitou vlastností je dodržení integrity dat v databázi.

Kompletní ER- diagram viz. Příloha A.

6.1 Tabulky

V této kapitole se nachází kompletní popis tabulek a atributů v databázi.

Id_zmeny ve všech tabulkách je cizí klíč tabulky *Upraveno* a v tabulkách se nachází z důvodu zaznamenání posledního změny řádku a informace o vzniku řádku. V dalším popisu nebude tento cizí klíč uváděn, protože jeho funkce je stále stejná.

Postavy

V tabulce se nacházejí všechny postavy v systému. *Id_serialu* určuje, v jakém seriálu postava hraje. *Id_herce* udává, kdo postavu hraje.

- *id_postavy* (**Number(4,0)**), primární klíč.
- *id_serialu* (**Number(4,0)**), cizí klíč z tabulky *Seriály*.
- *id_herce* (**Number(4,0)**), cizí klíč z tabulky *Herci*.
- *jmeno_postavy* (**Varchar2(50)**).

Seriály

Tabulka obsahuje všechny seriály v systému, ve kterém alespoň jedna postava hrála.

- *id_serialu* (**Number(4,0)**), primární klíč.
- *nazev_cesky* (**Varchar2(50)**), český název seriálu, pokud neexistuje tak originální název.
- *nazev_original* (**Varchar2(50)**), originální název seriálu.
- *datum_startu* (**Number(4,0)**), rok kdy se seriál začal vysílat.
- *datum_konce* (**Number(4,0)**), rok kdy seriál skončil.

Herci

Všichni herci, kteří hráli nebo namluvili alespoň jednu postavu, jsou v této tabulce.

- *id_herce* (**Number(4,0)**), primární klíč.
- *jmeno_herce* (**Varchar2(40)**), křestní jméno herce.
prijmeni_herce (**Varchar2(60)**), příjmení herce a případně prostřední jméno.
- *odkaz_na_biografii* (**Varchar2(200)**), obsahuje URL odkaz na biografii herce.

Reg_uzivatel

V této tabulce jsou všichni uživatelé, kteří se v aplikaci registrovali a mají zde uvedeny všechny potřebné údaje k přihlášení. *id_role_uzivatele* určuje jakou roli uživatel má.

- *id_reg_uzivatele* (Number(5,0)r), primární klíč.
- *id_role_uzivatele* (Number(2,0)), cizí klíč z tabulky Role_uzivatele.
- *login* (Varchar2(60)), přihlašovací jméno uživatele.
- *heslo* (Raw(60)), zahashované heslo uživatele.
- *mail* (Varchar2(60)), mail uživatele.

Souboje

Tabulka obsahuje všechny naplánované souboje, aktuální souboje a i všechny proběhlé souboje. *Id_kola* určuje, v jakém kole souboj probíhá. *Id_postavy_domaci* a *id_postavy_hosti* říká, jaké postavy se souboje účastní.

- *id_souboje* (Number(4,0)), primární klíč.
- *id_kola* (Number(3,0)), cizí klíč z tabulky Kola.
- *id_postavy_domaci* (Number(4,0)), cizí klíč z tabulky Postavy.
- *id_postavy_hosti* (Number(4,0)), cizí klíč z tabulky Postavy.
- *zacatek_souboje* (Date), den začátku souboje.
- *konec_souboje* (Date), den konce souboje.

Rocnik

V tabulce jsou všechny minulé ročníky a i všechny budoucí naplánované ročníky. Jeden ročník je rozdělen na několik kol a jednotlivá kola obsahují souboje, které jsou k těmto kolům přiřazena. *id_rocniku* je jedinečný identifikátor ročníku a je přiřazován postupně, takže také udává, o kolikátý se jedná ročník.

- *id_rocniku* (Number(3,0)), primární klíč.
- *nazev_rocniku* (Varchar2(60)).
- *datum_zacatku* (Date), den začátku ročníku.
- *datum_konce* (Date), den konce ročníku.
- *postup* (Number(3,0)), udává počet postav, které postoupí z nominace do hlavní části.

Kola

V této tabulce se nacházejí kola, ze kterých jsou ročníky složeny. Do jakého ročníku dané kolo spadá, určuje atribut *id_rocniku*.

- *id_kola* (Number(3,0)), primární klíč.
- *id_rocniku* (Number(3,0)) cizí klíč z tabulky Rocnik.
- *nazev_kola* (Varchar2(60)) doporučená jména jsou nominace, skupiny, čtvrtfinále aj.

- *zacatek_kola* (Date), den začátku kola.
- *konec_kola* (Date), den konce kola.
- *postup_skupin* (Number(2,0)), udává počet skupin nebo počet vyřazovacích soubojů

Foto

V tabulce jsou všechny fotky, které jsou v aplikaci použity.

- *id_fotky* (Number(4,0)), primární klíč.
- *url_fotky* (Varchar2(300)) URL adresa fotky.
- *popisek_fotky* (Varchar2(300)).

Fotky_serialu, Fotky_postavy, Fotky_hercu

Tři tabulky, které mají stejnou funkci a liší se pouze v druhé tabulce, kterou spojují s tabulkou Foto.

- *id_fotky* (Number(4,0)), cizí primární klíč z tabulky Foto.
- *id_****** (Number(4,0)) cizí klíč z tabulky Postavy, Herci nebo Serialy.

Clanky_hlavni

V této tabulce jsou uloženy všechny články, které jsou na úvodní straně aplikace.

- *id_clanku* (Number(4,0)), primární klíč.
- *nadpis* (Varchar2(1000)).
- *text* (Clob), celý text článku.

Upraveno

Zde jsou uloženy všechny informace o tom, kdy byl jaký řádek vytvořen a kým, a také kdy a kým byl naposledy upraven. *id_reg_uzivatele_insert* udává id uživatele, který řádek vložil a *id_reg_uzivatele_update* zase id uživatele, který řádek naposledy upravil.

- *id_zmeny* (Number(5,0)), primární klíč.
- *id_reg_uzivatele_insert* (Number(5,0)) cizí klíč z tabulky Reg_uzivatel.
- *id_reg_uzivatele_update* (Number(5,0)) cizí klíč z tabulky Reg_uzivatel.
- *vloženo* (Date), den vložení řádku.
- *updatovano* (Date), den poslední úpravy řádku.

Role_uzivatelu

Obsahuje všechny role, do kterých je možné uživatele zařadit.

- *id_role* (Number(2,0)), primární klíč.
- *nazev_role_uzivatele* (Varchar2(40)) např. administrátor.

Souteze

V této tabulce jsou všechny soutěže se všemi potřebnými údaji. Výherce soutěže se zjistí z tabulky Hlasy_do_souteze. *Id_rocniku* udává, pod jaký ročník daná soutěž spadá.

- *id_souteze* (Number(2,0)), primární klíč.
- *id_rocniku* (Number(3,0)) cizí klíč z tabulky Rocnik.
- *nazev_souteze* (Varchar2(60)).
- *zacatek* (Date), den začátku soutěže.
- *konec* (Date), den konce soutěže.
- *otazka* (Varchar2(500)), den konce soutěže.
- *odpovedA* (Varchar2(30)), první možná odpověď.
- *odpovedB* (Varchar2(30)), druhá možná odpověď.
- *odpovedC* (Varchar2(30)), třetí možná odpověď.
- *odpovedD* (Varchar2(30)), čtvrtá možná odpověď.
- *spravna_odpoved* (Varchar2(1)), písmeno odpovědi, která je správná.

Hlasy_do_souteze

V tabulce jsou zaznamenávány odpovědi registrovaných uživatelů do jednotlivých soutěží. *Id_souteze* a *id_reg_uzivatele* udává, do jaké soutěže odpověď patří, a kdo odpověď poslal.

- *id_hlasu_do_souteze* (Number(4,0)), primární klíč.
- *id_souteze* (Number(2,0)) cizí klíč z tabulky Souteze.
- *id_reg_uzivatele* (Number(5,0)) cizí klíč z tabulky Reg_uzivatel.
- *odpoved* (Varchar2(1)), písmeno odpovědi (A,B,C,D).

Tipy_na_viteze

Primárním klíčem této tabulky jsou dva cizí klíče. První je *id_reg_uzivatele* z tabulky Reg_uzivatel a druhým je *id_postavy* z tabulky Postavy. Cílem tabulky je zaznamenávat tipy jednotlivých uživatelů na vítěze. Uživatel může v rámci jednoho ročníku tipovat jednu postavu pouze jednou. Pomocí atributu *id_rocniku* je možné zjistit, do kterého ročníku tip patří.

- *id_reg_uzivatele* (Number(4,0)), cizí primární klíč z tabulky Reg_uzivatel.
- *id_postavy* (Number(5,0)) cizí primární klíč z tabulky Postavy.
- *id_rocniku* (Number(3,0)) cizí klíč z tabulky Rocnik.

Akt_nominace

Tato tabulka má na starosti zaznamenávání všech nominací od uživatelů podle jednotlivých ročníků. Také je zde uložena IP adresa uživatele, který nominaci provedl.

- *id_akt_nominace* (Number(5,0)), primární klíč.
- *id_reg_uyivatele* (Number(5,0)) cizí klíč z tabulky Reg_uzivatel.
- *id_rocniku* (Number(3,0)) cizí klíč z tabulky Rocnik.

- *id_postavy* (Number(4,0)), Cizí klíč z tabulky Postavy.
- *ip_adresa* (Varchar2(50)), IP adresa nominujícího uživatele.

Logy_hlasu

V této tabulce jsou uloženy všechny hlasy do soubojů.

- *id_logu* (Number(8,0)), primární klíč.
- *id_souboje* (Number(4,0)) cizí klíč z tabulky Souboje.
- *id_reg_uzivatele* (Number(5,0)) cizí klíč z tabulky Souboje.
- *pro_koho* (Date), pokud je hlas pro domácího, tak 1, pokud pro hostujícího, tak 2.
- *cas_hlasu* (Date), kdy byl hlas zaznamenán.
- *ip_adresa* (Varchar2(50)), z jaké IP adresy byl hlas odeslán.

Fun_weby

Záznamy o fanouškovských webech k seriálům.

- *id_webu* (Number(3,0)), primární klíč.
- *id_serialu* (Number(4,0)) cizí klíč z tabulky Serial.
- *nazev* (Varchar2(60)) název fun webu.
- *url_adresa* (Varchar2(50)), URL adresa fun webu.

Skupiny

Tabulka obsahuje skupiny, které jsou v daném ročníku a v daném kole.

- *id_skupiny* (Number(3,0)), primární klíč.
- *id_rocniku* (Number(3,0)) cizí klíč z tabulky Rocnik.
- *nazev_skupiny* (Varchar2(100)) název skupiny, zpravidla A, B, C atd.
- *id_kola* (Number(3,0)), cizí klíč z tabulky Kola.
- *postupuje* (Number(2,0)), počet postav, které ze skupiny postoupí.

Postavy_ve_skupine

Tato tabulka přiřazuje postavy ke skupinám. Primárním klíčem jsou dva cizí klíče.

- *id_skupiny* (Number(4,0)), primární klíč.
- *id_postavy* (Number(3,0)), primární klíč.

Statistiky_postav

V této tabulce jsou záznamy o statistikách k postavám za jednotlivé ročníky. Statistiky jsou aktualizovány po každém kole v ročníku.

- *id_statistiky* (Number(5,0)), primární klíč.
- *id_rocniku* (Number(3,0)) cizí klíč z tabulky Rocnik.
- *id_postavy* (Number(4,0)), cizí klíč z tabulky Postavy

- *celkovy_pocet_hlasu* (Number(6,0)), celkový počet hlasů v rámci ročníku.
- *celkem_vyher* (Number(2,0)), celkový počet výher v ročníku.
- *celkem_proher* (Number(2,0)), celkový počet proher v ročníku.
- *celkem_remiz* (Number(2,0)), celkový počet remíz v ročníku.
- *počet_nominaci* (Number(4,0)), celkový počet nominací v ročníku.
- *vyher_ve_skupine* (Number(2,0)), celkový počet výher ve skupině v ročníku.
- *remiz_ve_skupine* (Number(2,0)), celkový počet remíz ve skupině v ročníku.
- *proher_ve_skupine* (Number(2,0)), celkový počet proher ve skupině v ročníku.
- *posledni_update* (Date), čas posledního updatu záznamu.

6.2 Procedury

Procedura je programová jednotka, jejíž příkazy jsou uloženy v databázi. Uložené procedury jsou přeloženy pouze jednou a při jejich volání se zasílají pouze informace o volání a hodnotě parametrů.

V databázi jsou procedury, které mají na starosti tvorbu soubojů a generování statistik.

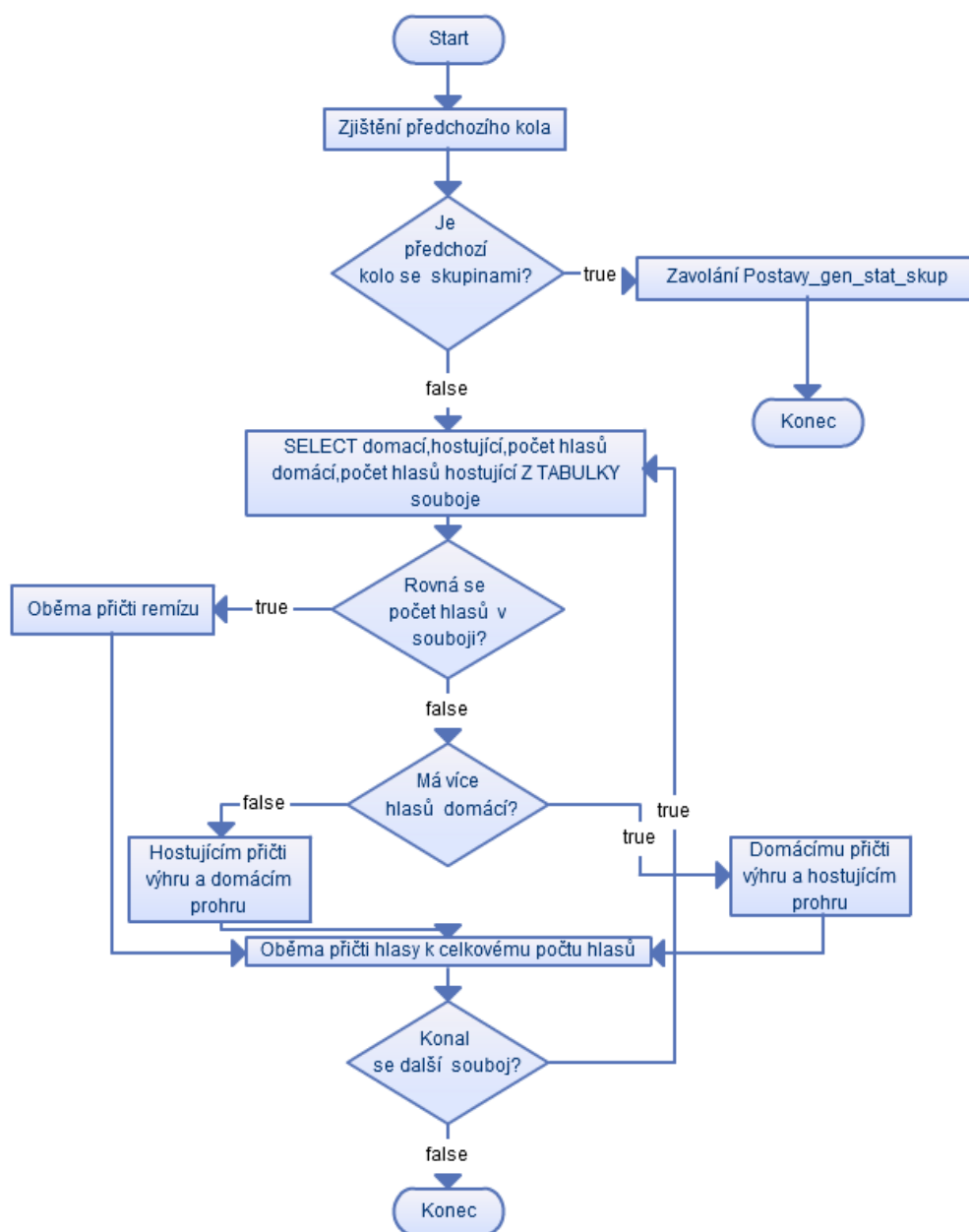
Postavy_gen_stat_pav

Tato procedura je volána při generování statistik ve vyřazovací části. Pokud je minulé kolo také vyřazovací, proběhne tato procedura, ale pokud jsou v minulém kole skupiny, je zavolána procedura *postavy_gen_stat_skup*. Ke zjištění předchozího kola je volána funkce *postavy_zjisti_pred_kola*. Po ověření minulého kola jsou v cyklu volány jednotlivé souboje pomocí kurzoru. U každého souboje se zjistí počet hlasů a vyhodnotí vítěz. Na konci cyklu se zavolá UPDATE na tabulku statistiky_postav. Jak procedura funguje je vidět na vývojovém diagramu (Obr. 10).

SELECT v kurzoru vrátí id domácí a hostující postavy. Dva vnořené dotazy spočítají počet *id_logu* v tabulce *logy_hlasu*, kde se *id_souboje* rovnají, v první dotazu SELECT pro domácí postavu a ve druhém pro hostující postavu. Vzhledem k tomu, že se výpočet statistik provádí po skončení kola, budou zjištěné výsledky pro předchozí kolo.

Kurzor:

```
CURSOR c1 IS SELECT
    DISTINCT "id_postavy_domaci",
    "id_postavy_hosti",
    (SELECT COUNT("id_logu") FROM "logy_hlasu" WHERE "id_souboje" =
a."id_souboje" AND "pro_koho" = 1) as "domaci",
    (SELECT COUNT("id_logu") FROM "logy_hlasu" WHERE "id_souboje" =
a."id_souboje" AND "pro_koho" = 2) as "hosti"
FROM "souboje" a
LEFT JOIN "logy_hlasu" ON a."id_souboje" = "logy_hlasu"."id_souboje"
WHERE "id_kola" = idPred;
```



Obr. 10 – Vývojový diagram generování statistik vyřazovacích soubojů

Postavy_gen_stat_skup

Procedura pracuje velice podobně jako *postavy_gen_stat_pav*. Rozdíl je v tom, jaké atributy upravuje v tabulce statistiky_postav. Algoritmus je stejný. Místo aktualizace statistik z vyřazovacích soubojů aktualizuje statistiky ze skupin.

Postavy_generuj_soub_pav

Tato procedura má na starost vytváření soubojů ve vyřazovací části. Algoritmus lze rozdělit na dvě části podle předchozího kola. Je rozdíl, jestli předchozí kolo bylo

vyřazovací, nebo se jednalo o skupiny. Práce procedury je vidět na vývojovém diagramu (Obr. 11).

Tvorba soubojů ze skupin je tvořena ze dvou vnořených cyklů do sebe. První z cyklů postupně prochází skupiny. V tomto cyklu se vybere skupina i a skupina $i+1$. Souboje budou vytvořeny z těchto dvou skupin. Cyklus končí, když se dostane na předposlední skupinu v pořadí. Druhý cyklus je vnořen v prvním a v jeho těle se vybírají postavy, které budou spojeny do souboje. Počet opakování je podle počtu postav, které ze skupin mají postoupit. Uvnitř cyklu jsou dva dotazy SELECT, které vybírají postavy. První SELECT vybírá z první skupiny od vrchu podle počtu bodů, a druhý SELECT z druhé skupiny od spodu také podle počtu bodů. Z těchto dvou vybraných postav se vytvoří souboj do vyřazovací části. V dalším opakování se vybírá podle stejného pravidla, jenom se přeskočí již vybrané postavy. Tímto způsobem se spárují všechny postavy a projdou se všechny skupiny.

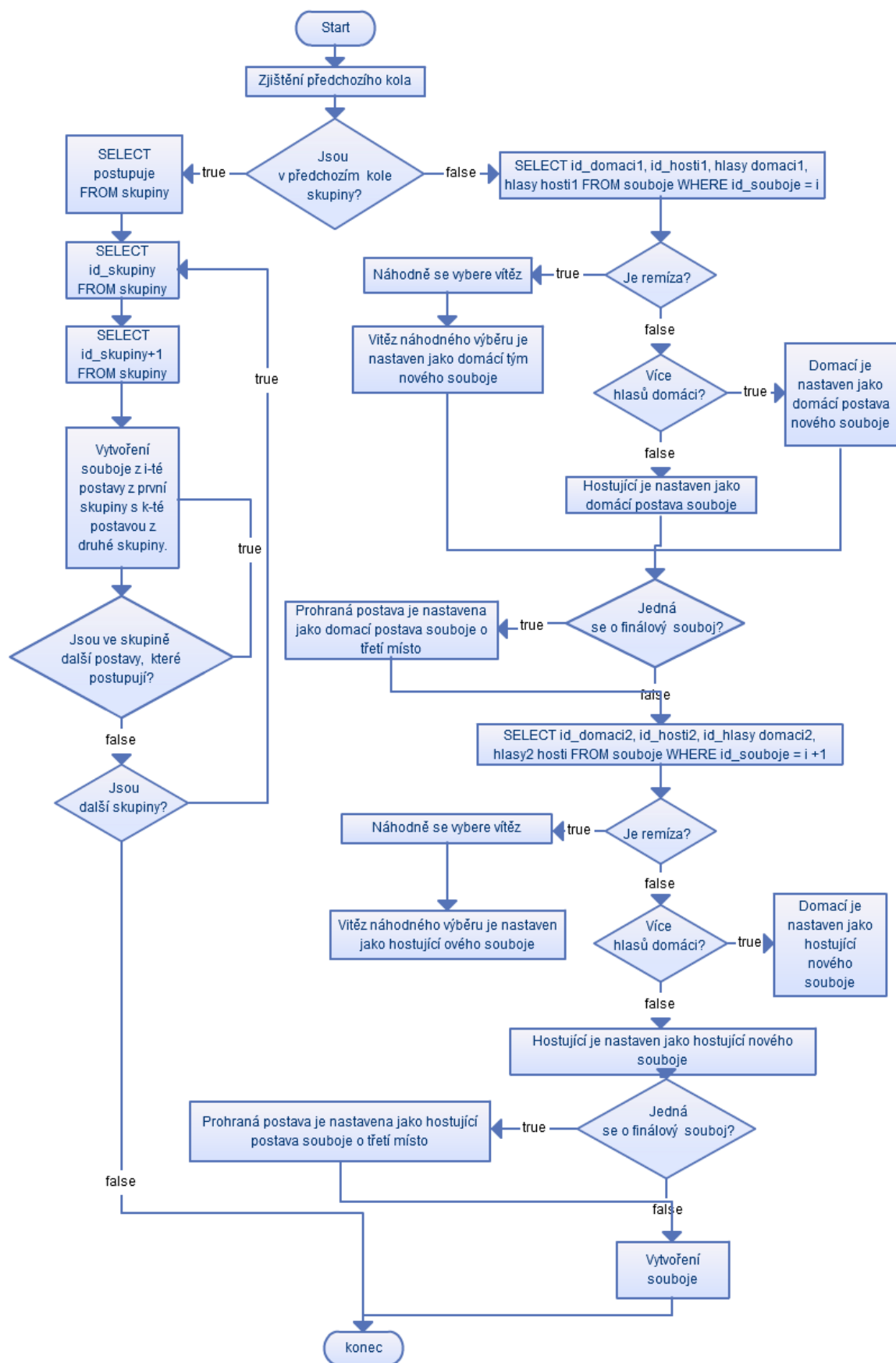
SELECT pro výběr postavy ze skupiny:

```
SELECT "id_postavy" INTO idPosD FROM
(SELECT ("vyher_ve_skupine"*3+"remiz_ve_skupine"), a.*, row_number()
over(ORDER BY ("vyher_ve_skupine"*3+"remiz_ve_skupine") DESC) rn FROM
"postavy_ve_skupine" a
JOIN "postavy" ON a."id_postavy" = "postavy"."id_postavy"
JOIN "statistiky_postav" ON a."id_postavy" =
"statistiky_postav"."id_postavy"
WHERE "id_skupiny" = skup1 AND "id_rocniku" = idRoc) WHERE rn=k;
```

Tvorba soubojů z vyřazovací části je složena z jednoho cyklu a ze dvou hlavních dotazů SELECT. Cyklus prochází souboje a končí, až se dostane na předposlední souboj. Uvnitř cyklu jsou dva dotazy SELECT, které vybírají souboj i a souboj $i+1$. SELECT vybere id obou postav a počet hlasů získaných v tomto souboji. Následně je v několika podmínkách vybrán vítěz souboje, a pokud nastala remíza, je vítěz vybrán náhodně. Také je ověřeno, zda se nejedná o poslední kolo ročníku, čili finále, a pokud ano, do proměnné se uloží i poražená postava tohoto souboje kvůli tvorbě souboje o třetí místo. To samé se provede i pro druhý souboj vybraný druhým dotazem SELECT. Z těchto dvou soubojů je vytvořen jeden nový a případně i souboj o třetí místo.

SELECT pro výběr souboje:

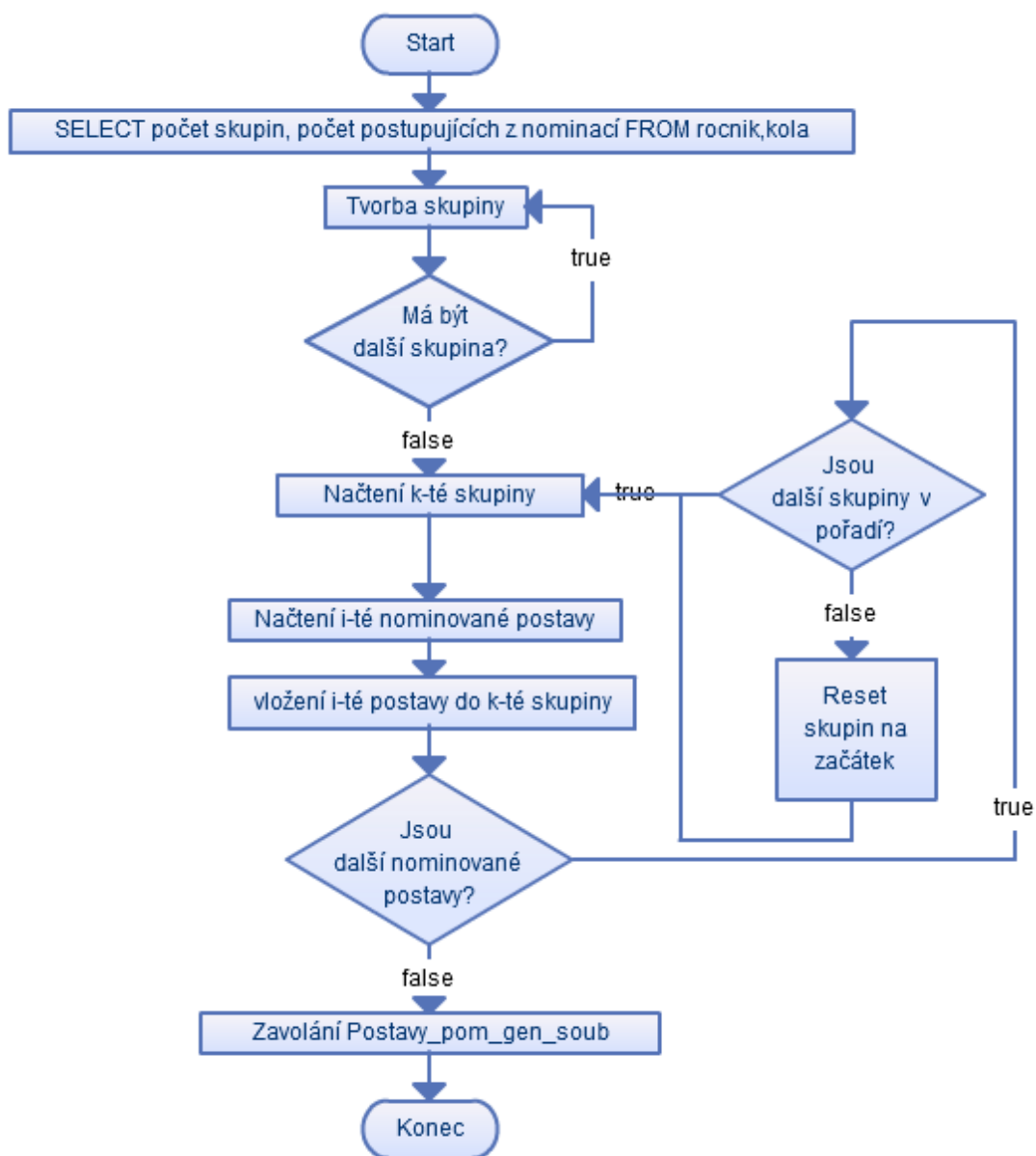
```
SELECT DISTINCT "id_postavy_domaci", "id_postavy_hosti",
(SELECT COUNT("id_logu") FROM "logy_hlasu" WHERE a."id_souboje" =
"logy_hlasu"."id_souboje" and "pro_koho" = 1) as "domaci",
(SELECT COUNT("id_logu") FROM "logy_hlasu" WHERE a."id_souboje" =
"logy_hlasu"."id_souboje" AND "pro_koho" = 2) AS "hosti"
INTO idPosD, idPosH, POCDOM, POCHOS
FROM (SELECT b.*, row_number() over(ORDER BY "id_souboje") rn FROM "souboje"
b
where "id_kola" = idPred) a
LEFT JOIN "logy_hlasu" ON a."id_souboje" = "logy_hlasu"."id_souboje"
WHERE rn=j;
```



Obr. 11 – Vývojový diagram tvorby vyřazovacích soubojů

Postavy_generuj_soub_skup

Úkolem této procedury je vytvoření skupin pro postavy a přiřazení postav do nově vygenerovaných skupin. V první části procedury je cyklus, ve kterém jsou vytvořeny skupiny. Skupiny jsou pojmenovány podle abecedy. Ve druhé části se nacházejí dva vnořené cykly do sebe, jejichž úkolem je do nově vygenerovaných skupin přiřadit postupující postavy z nominační části. Ve vnitřním cyklu se procházejí skupiny, a ve vnějším postavy. Postavy jsou roztrženy tak, aby se nejúspěšnější postavy v nominaci nepotkaly hned v základních skupinách. Diagram procedury je na obrázku (Obr. 12).

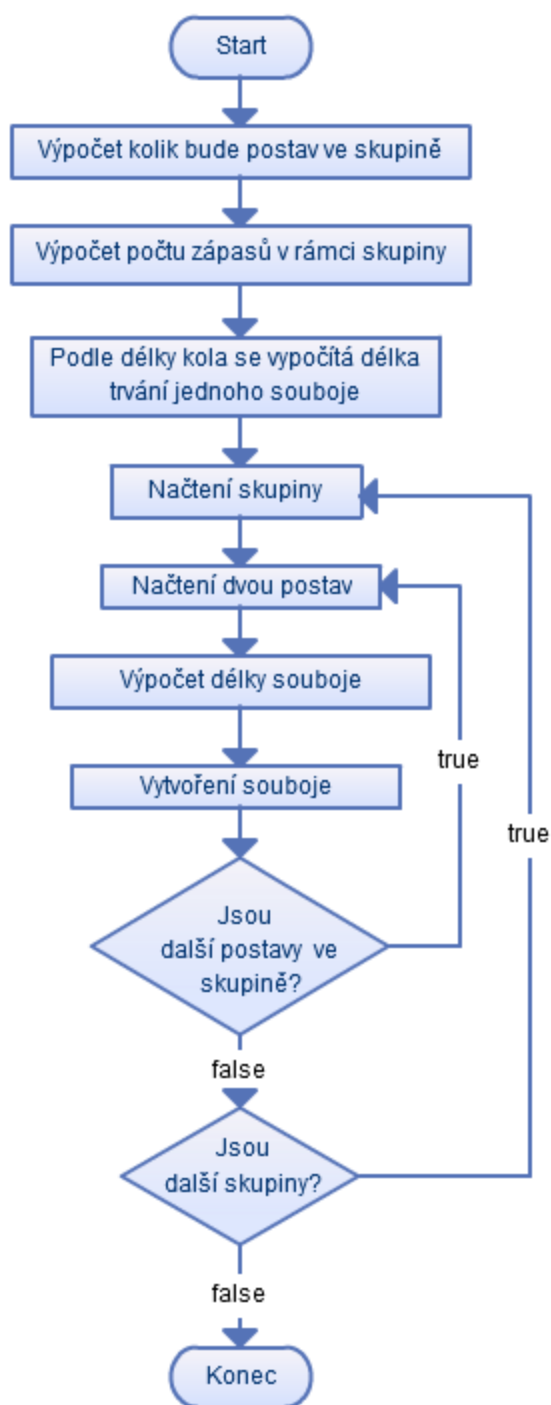


Obr. 12 – Vývojový diagram tvorby skupin a přiřazení postav

Postavy_pom_gen_soub

Tato procedura je volána z procedury *Postavy_generuj_soub_skup* a má za úkol vytvořit souboje v rámci skupin. V první části jsou vypočítány všechny potřebné informace, například počet postav ve skupině a počet možných kombinací mezi jednotlivými postavami. Výpočet je proveden pomocí kombinatoriky a vydělený dvěma, protože se souboje účastní dvě postavy. Výsledkem je počet podkol, na které bude kolo se skupinami rozděleno, aby současně neběžely dva souboje se stejnými postavami. Není možné, aby jedna postava byla ve dvou soubojích současně. Také se spočítá, jak dlouhá budou tato podkola, aby se všechna vešla do kola se skupinami.

Samotná tvorba soubojů je složena ze třech vnořených cyklů. První z nich postupně prochází všechny skupiny. V těle druhého a třetího cyklu je výběr dvou postav, ze kterých se vytvoří souboj. Po tomto výběru se přes několik podmínek nastaví den začátku souboje a den konce souboje. Na střídačku se nastavují data od začátku kola a od konce kola. To je z důvodu, že postavy jsou ze skupiny vybírány postupně, a ve stejném okamžiku by byly dvě postavy ve více soubojích. Diagram procedury je na obrázku (Obr. 13).



Obr. 13 – Vývojový diagram tvorby soubojů ve skupině

6.3 Funkce

Funkce mají stejné vlastnosti jako procedury a chovají se také stejně. Rozdíl je v tom, že funkce má návratnou hodnotu.

Postavy_factorial_calc

Tato funkce vrací faktoriál předaného čísla.

Funkce vytvořena podle (Rajavu, 2008).

```
CREATE OR REPLACE
FUNCTION postavy_factorial_calc(p_MyNum NUMBER)
RETURN NUMBER AS
BEGIN
    IF p_MyNum = 1 THEN
        RETURN 1;
    ELSE
        RETURN(p_MyNum * postavy_factorial_calc(p_MyNum-1));
    END IF;
END postavy_factorial_calc;
```

Postavy_zjisti_pred_kola

Funkce zjistí *id_kola*, které bylo před aktuálním kolem. Nejdříve zjistí aktuální kolo, poté pomocí kurzoru vypíše všechna kola v ročníku seřazená podle začátku a v cyklu zkontroluje, jestli další vypsané kolo není aktuální kolo. Pokud ano, cyklus skončí a vrátí id předchozího kola.

6.4 Ostatní

V databázi jsou také použity sekvence. Sekvence jsou automatické čítače, které se podle nastavených hodnot inkrementují. Jsou použity pro nastavování hodnot primárních klíčů id u většiny tabulek. Je tím zamezeno duplicitě primárních klíčů, protože nikdo nemůže id do tabulky přidávat ručně. V některých případech, ale nejsou vhodné a id je nastavováno ručně.

7 Vývoj aplikace

Tato kapitola se zabývá funkcemi aplikace z pohledu vývoje a způsoby, jak jsou tyto funkce vyřešeny.

7.1 Dynamika aplikace

Aplikace je postavena dynamicky a je skládána z jednotlivých částí. Hlavička je stále stejná, ale menu se nahrávají různá. Záleží, jestli je uživatel přihlášen a jestli se jedná o administrátora či běžného uživatele. Obsah se mění také dynamicky podle toho, na co uživatel klikne.

Menu

Jsou tři druhy menu a několik druhů podmenu. Menu jsou základní, pro přihlášené uživatele a pro administrátory. Každé se nahrává podle session jménem user a podle hodnot, které jsou v ní uloženy. Pokud session neexistuje, je nahráno základní menu.

Podmenu jsou nahrána podle toho, kde a jakou stránku uživatel otvírá. Kontrola, jestli má být podmenu nahráno, je podle metody GET. Vzhledem k tomu, že je podmenu spojeno s konkrétní stránkou a tím pádem s konkrétní metodou GET, dá se lehce zjistit, kdy a jaké podmenu nahrát.

Hlavní obsah

Jaká stránka má být nahrána se určí podle hodnoty v metodě GET['co']. První se v podmínkách kontroluje, jestli je nastavena nějaká konkrétní metoda GET, a pokud ano, je přímo nahrána konkrétní stránka. Pokud metoda GET neodpovídá žádné podmínce, je kontrolována metoda GET['co']. Podle této metody se nahraje v podmínce switch požadovaná stránka. Uvnitř podmínky switch se nachází druhá podmínka switch, která má za úkol po kontrole přihlášení a kontrole zda je uživatel administrátor, nahrát stránku z administračního rozhraní, pokud je vyžadována.

7.2 Formuláře

Formuláře jsou v aplikaci vytvořeny pomocí Nette Framework. Výhody tohoto řešení jsou snadná implementace a obsažené základní bezpečnostní prvky v Nette. Validace dat je na straně serveru i klienta. Na straně klienta je kontrola provedena pomocí JavaScriptu. Nette formuláře kontrolují vstupní data – odfiltrují kontrolní znaky, kontrolují validitu UTF-8 kódování nebo také jestli nejsou položky v select boxech podvržené. (Nette Foundation, 2012)

Jednoduchý formulář lze vytvořit například takto: (Převzato z (Nette Foundation, 2012))

```
require 'Nette/loader.php';
use Nette\Forms\Form;
$form = new Form;
$form->addText('name', 'Jméno:');
$form->addPassword('password', 'Heslo:');
$form->addSubmit('send', 'Registrovat');
echo $form; // vykreslí formulář
```

Práce s Nette formuláři je velice jednoduchá a intuitivní. Programátorovi ušetří spoustu času, protože nemusí psát kontroly a validace. Všem formulářům je možné jednoduše přidávat vlastní validační pravidla. Lze například přidat, že políčko musí být vyplněno a toto pravidlo je hned na straně uživatele zkontrolováno.

Ve všech vstupních formulářích získá focus textové políčko i v případě, že uživatel klikne místo do textového políčka na text.

Data z formulářů lze jednoduše získat pomocí příkazu:

```
$values = $form->getValues();
```

Odeslání formuláře je kontrolováno pomocí:

```
$form->isSuccess()
```

7.3 Hesla a bezpečnost

Zabezpečení webové aplikace patří k jejím nejdůležitějším prvkům. Pokud po uživateli nejsou požadovány citlivé údaje, jako je např. rodné číslo nebo číslo kreditní karty, nejsou nároky na bezpečnost vysoké.

Na bezpečnost se lze dívat i z jiného pohledu. Kromě zabezpečení dat uživatelů je třeba také zabezpečit webové stránky proti útokům na stránku samotnou.

Hesla

Aplikace používá pro zašifrování hesla algoritmus SHA1 a tzv. „solení“. K heslu je připojeno zašifrované uživatelské jméno pomocí SHA1 a tento celý řetězec je dále zašifrován také pomocí tohoto algoritmu. Takto zašifrované heslo je uloženo v databázi.

Postup šifrování hesla:

```
$password = sha1($values->password . sha1($values->name));
```

Bezpečnost aplikace

Aplikace má ošetřené vstupy automaticky pomocí Nette formulářů. Ty mají implementovanou ochranu proti Cross-Site Scripting (XSS) i proti Cross-Site Request Forgery (CSRF).

XSS je útok na webovou stránku prostřednictvím neošetřených vstupů. Útočník může do stránky podstrčit vlastní kód, a tím může stránku pozměnit, nebo získat citlivé údaje o uživateli. (Nette Foundation, 2012)

CSRF se snaží přimět uživatele navštívit stránku, která skrytě vykoná útok na aplikaci, ve které je uživatel přihlášen. Např. lze pomocí tohoto útoku smazat nebo pozměnit článek, aniž by si toho uživatel všiml. (Nette Foundation, 2012)

Obrana databáze proti SQL injection je zajištěna pomocí funkce `oci_bind_by_name`, která je použita pro každý uživatelský vstup do databáze.

Ukázka `oci_bind_by_name`:

```
$stid = oci_parse($this->db, 'SELECT * FROM "reg_uzivatel" WHERE "login" = :e AND "heslo" = hextoraw(:p)');  
  
oci_bind_by_name($stid, ':e', $values->name);  
oci_bind_by_name($stid, ':p', $password);
```

7.4 Uživatelé

Uživatelé aplikace se mohou na internetových stránkách registrovat. Při každé další návštěvě se přihlašují a získávají tím výhody, které nepřihlášení uživatelé nemají.

7.4.1 Registrace

Uživatel má možnost se v aplikaci registrovat. Je po něm vyžadováno uživatelské jméno, heslo a mail. Uživatelské jméno a heslo musí být unikátní. V podmínce je zkontrolováno nejdříve uživatelské jméno a poté mail. Pokud ani jedna hodnota v databázi není, tak je vytvořen nový účet a uživatel se může přihlásit.

Ve formuláři musí uživatel vyplnit heslo pro kontrolu dvakrát a hesla se musejí shodovat. Kontrolu zajišťuje pravidlo přidané k inputu. Input je typu password, takže vše co je do něj napsáno, je automaticky převedeno na hvězdičky. Heslo musí být dlouhé minimálně 4 znaky a tuto podmínku zajišťuje další pravidlo přidané k inputu.

Pravidla a podmínky při zapisování hesla:

```
$form->addPassword('password', 'Vytvořte heslo:');  
->addRule(Form::FILLED, 'Zadejte prosím své heslo');  
->addRule(Form::MIN_LENGTH, 'Heslo musí mít alespoň %d znaky', 4);  
$form->addPassword('password2', 'Potvrďte heslo:');  
->addRule(Form::FILLED, 'Zadejte prosím své heslo ještě jednou');  
->addRule(Form::EQUAL, 'Hesla se neshodují', $form['password']);
```

Input pro mailovou adresu má přidané pravidlo, že se má očekávat mailová adresa. Zadaný řetězec je zkontrolován javascriptem a pokud formát nesouhlasí, je uživatel hned upozorněn.

7.4.2 Přihlášení a odhlášení

Uživatel zadá přihlašovací údaje, které jsou následně ověřeny s daty v databázi. Pokud je v databázi řádek, který obsahuje uživatelské jméno a shoduje se i heslo, je vytvořena session jménem user do které jsou uloženo tři hodnoty. True hodnota, že je uživatel přihlášen, uživatelské jméno a id uživatele. Po vytvoření session je zkontrolováno, zda je přihlašovaný uživatel administrátor. Pokud ano, je do session přidána hodnota true a pokud ne tak, hodnota false. Je-li session vytvořená, je uživatel přihlášen a přesměrován na úvodní stranu jako přihlášený uživatel.

Odhlášení uživatele je vyřešeno pomocí postupného unset na session user a následné zrušení session pomocí session_destroy() a přesměrování na hlavní stránku. Na hlavní stránce se uživateli načte menu pro nepřihlášeného uživatele, protože session byla zrušena.

7.4.3 Generování výherce soutěže

Výherce soutěže je generován pomocí SQL. Tato funkce není automatická a je nutné stisknout tlačítko generování. Výherce je vybrán ze všech uživatelů, kteří soutěžili, a jejichž odpověď na otázku byla správná. Algoritmus využívá funkci *dbms_random.value* podle které jsou výsledky seřazeny a následně je vybrán první řádek takto seřazených výsledků.

SQL dotaz pro vylosování výherce:

```
SELECT "login" FROM
( SELECT "login" FROM "hlasy_do_souteze"
JOIN "souteze" USING("id_souteze")
JOIN "reg_uzivatel" USING("id_reg_uzivatele")
WHERE "id_souteze" = :idSout
AND "odpoved" LIKE "spravna_odpoved"
ORDER BY dbms_random.value )
WHERE rownum = 1
```

7.5 Ročníky

Ročník může vytvářet pouze administrátor. U ročníku je potřeba zadat od kdy do kdy pobeží, ročníky se nemůžou krýt. Kontrola probíhá pomocí dotazu SELECT, který vypíše všechny ročníky v databázi, a zadaná data jsou kontrolována pomocí podmínky vůči těmto vypsáním ročníkům. Pokud je datum v pořádku, je ročník vložen do databáze.

Datum musí být zapsáno ve správném formátu, který je d.m.rrrr. O kontrolu platnosti data se stará metoda *platne_datum*, která datum kontroluje pomocí regulárního výrazu a vrací false nebo true podle úspěchu kontroly. Pokud je datum ve špatném formátu vrací také false.

Kontrola platnosti data: (Vrána, 2006)

```
preg_match('~^([1-9]|19|[12][0-8]|29(=?\\.[^2]|2\\.\\.([02468][048]|13579)[26])00|[0-9]{2}(0[48]|2468[048]|13579)[26]))|30(=?\\.[^2])|31(=?\\.[13578][02]?\\\\.\\.))\\.[1-9]|1[012])\\.([0-9]{4})$~D', $datum);
```

7.6 Kola

Kola lze vytvářet v rámci ročníků. Kolo nelze vytvořit, aniž by patřilo do některého ročníku. O výpis kol se stará metoda *vlozitRadky* ve třídě *Kola*. Tato metoda má dva parametry. Prvním parametrem je *\$admin*, který udává, jestli je metoda volána z administrační části nebo z uživatelské. Druhý parametr předává id ročníku, který je aktuální. Pokud je metoda volána z administrační části, je na konec přidán odkaz na editaci kola a na začátek id kola. Také je kontrolováno aktuální datum, a pokud je kolo aktuální, je do tagu vloženo id aktRadek. Toto id je v CSS zvýrazněno červenou barvou.

Přidání kola probíhá velice podobně jako přidání nového ročníku. Rozdíl je ve větší náročnosti na ověření data – jestli nezasahuje do jiného kola a jestli datum nepřesahuje hranice ročníku. Pomocí dotazu *SELECT* jsou vypsána všechna kola v rámci ročníku. Poté jsou v cyklu postupně načítány všechny začátky a konce kol, které jsou postupně kontrolovány se zadanými daty v metodě *overeniDataKola*. V této metodě je nejdříve zkontrolováno, jestli začátek kola není déle než konec kola. Poté je zkontrolováno, jestli kolo nezasahuje do jiného kola.

Můžou být tři druhy kol; nominace, skupiny nebo vyřazovací kolo. Každé kolo probíhá jiným způsobem.

7.6.1 Nominace

Nominace jsou vždy na začátku ročníku. V tomto kole se rozhoduje, které postavy se dostanou do hlavní části. Nominovat postavy lze, pokud je aktuální kolo „nominace“. Tato kontrola se provádí pomocí jednoduchého dotazu na databázi.

```
SELECT "id_kola" FROM "kola" WHERE SYSDATE BETWEEN "zacatek_kola" AND "konec_kola" AND "nazev_kola" LIKE 'nominace'
```

Pokud tato kontrola proběhne úspěšně, je možné dostat se do nominací a nominovat postavy.

O získávání informací o postavách je kapitola 7.9. Tato část je zaměřena na ověřování a ukládání těchto dat do databáze.

Po vybrání postavy uživatelem a kliknutí na tlačítko „Nominuj“ je zavolána metoda *nominuj*. V ní jsou posbírané informace správně roztříděny a hlavně rozkouskovány, aby mohly být správně uloženy do databáze. Např. jméno a příjmení je rozděleno na samostatné jméno a samostatné příjmení. Takto jsou zpracovány všechny informace.

```

$z1 = strpos($nomSerial, "(");
$z2 = strpos($nomSerial, ")");
$serCesky = substr($nomSerial, $z2 + 4);
$serOrig = substr($nomSerial, 0, $z1 - 1);
$rokZ = substr($nomSerial, $z1 + 1, 4);
$rokK = substr($nomSerial, $z2 - 4, 4);
if ($rokK == "????") {
$rokK = NULL;

```

Po této úpravě je zavolána metoda *vlozDoDb*, do které jsou potřebné informace o postavě předány parametrem. V této metodě proběhne kontrola, jestli není postava v databázi a pokud je, získá se její id. Pokud postava není v databázi, je do databáze vložena. Vkládá-li se nová postava do databáze, je nutné ověřit, zda seriál a herec již v databázi nejsou. To se provádí zavoláním metod *overeniSerialu* a *overeniHerce*. Pokud zde nejsou, jsou v databázi vytvořeny a jejich id se vrátí do metody *vlozDoDb*. Pokud v databázi jsou, tak je pouze získáno jejich existující id. S těmito všemi id je možné do databáze vložit novou postavu a získat její id.

Po ověření, jestli uživatel může pro tuto postavu hlasovat, je možné zapsat nominaci. Hlasovat se může jednou denně. Kontrola je buď podle id přihlášeného uživatele, nebo podle IP adresy nepřihlášeného. SELECT nad tabulkou *akt_nominace* vypíše záznamy vložené tímto uživatelem a datum posledního vloženého hlasu. Není-li žádný záznam z tabulky *akt_nominace* vypsán, může být vložena nová nominace. V případě, že záznam existuje, je zkontrolováno datum vložení a shoda s aktuálním datem. Pokud se data shodují, je nominace odmítnuta.

SELECT pro výpis nominací podle IP:

```

SELECT to_char("vlozeno", 'DD.MM.YYYY') AS "vlozeno"
FROM "akt_nominace" JOIN "upraveno" USING("id_zmeny")
WHERE "id_postavy" = :pos
AND "id_reg_uzivatele" IS NULL
AND "ip_adresa" LIKE :ip
AND "id_rocniku" = :idRoc
ORDER BY "vlozeno" DESC'

```

Kontrola data poslední nominace:

```

$vlozeno = new DateTime($row->vlozeno);
$vlozeno->add(new DateInterval('PT24H'));
$datum = new DateTime();
if ($datum > $vlozeno) {
$overeni = 1;}

```

7.6.2 Skupiny

Kolo se skupinami zpravidla bývá jedno. Jedná se o druhé kolo v ročníku. Počet skupin je nastaven při vytváření kola a je závislý na počtu postav, které do tohoto kola postoupí z nominací.

Pokud je toto kolo aktuální, jsou na stránce „Souboje“ vypsány postavy, které jsou v aktuální skupině a pod skupinou jsou všechny souboje, které patří k této skupině.

Výpis řádku do tabulky skupiny zajišťuje metoda *vlozitRadky* ve třídě Souboje. SQL dotaz vypíše všechny postavy, které patří do vybrané skupiny v aktuálním ročníku. K postavám jsou připojeny jejich statistiky, fotky a seriál v jakém hrají. Výsledek dotazu je seřazený podle bodového hodnocení.

SQL dotaz ve funkci *vlozitRadky*:

```
SELECT DISTINCT "url_fotky", "jmeno_postavy", "nazev_original",
"vyher_ve_skupine", "remiz_ve_skupine", "proher_ve_skupine",
("vyher_ve_skupine"*3+"remiz_ve_skupine"*1) AS "body"
FROM "postavy"
LEFT JOIN "fotky_postavy" USING("id_postavy")
LEFT JOIN "foto" USING("id_fotky")
LEFT JOIN "postavy_ve_skupine" USING("id_postavy")
LEFT JOIN "skupiny" USING("id_skupiny")
LEFT JOIN "statistiky_postav" USING("id_postavy")
JOIN "serialy" USING("id_serialu")
JOIN "rocnik" ON "statistiky_postav"."id_rocniku" = "rocnik"."id_rocniku"
WHERE "id_skupiny" = :id
AND SYSDATE BETWEEN "datum_zacatku" AND "datum_konce"
ORDER BY "body" DESC, "vyher_ve_skupine" DESC, "remiz_ve_skupine"
DESC, "proher_ve_skupine" ASC;
```

7.7 Souboje

O výpis soubojů se stará metoda *vypsatZapasy*. Tato metoda má tři parametry. Prvním parametrem je *\$id*, který udává pro jakou skupinu vypsát souboje. Pokud je null, souboje nejsou ve skupině. Druhým parametrem je *\$min*, který značí, jestli se mají vypsát aktuální zápasy nebo zápasy minulé. Posledním parametrem je *\$hlasPred*, ve kterém je informace o úspěšnosti minulého hlasování.

Podle prvních dvou parametrů se vybere jeden ze tří dotazů SELECT, který vypíše správné souboje z databáze.

Ukázka jednoho SELECTu:

```
SELECT "id_souboje",d."jmeno_postavy" AS "domaci",h."jmeno_postavy" AS
"hosti",d."id_postavy" AS "id_dom",h."id_postavy" AS "id_hos",
s."nazev_original" AS "serDom",s2."nazev_original" AS "serHos",
f2."url_fotky" AS "foto_domaci",f4."url_fotky" AS "foto_hosti",
to_char("zacatek_souboje", 'FMDD.MM.YYYY') AS "zacatek_souboje2",
to_char("konec_souboje", 'FMDD.MM.YYYY') AS "konec_souboje2"
FROM "souboje"
JOIN "kola"USING("id_kola")
JOIN "rocnik" USING("id_rocniku")
JOIN "postavy" d ON "id_postavy_domaci" = d."id_postavy"
JOIN "postavy" h ON "id_postavy_hosti" = h."id_postavy"
JOIN "postavy_ve_skupine" a ON d."id_postavy" = a."id_postavy"
JOIN "serialy" s ON d."id_serialu" = s."id_serialu"
JOIN "serialy" s2 ON h."id_serialu" = s2."id_serialu"
LEFT JOIN "fotky_postavy" f1 ON d."id_postavy" = f1."id_postavy"
LEFT JOIN "foto" f2 ON f1."id_fotky"=f2."id_fotky"
LEFT JOIN "fotky_postavy" f3 ON h."id_postavy" = f3."id_postavy"
LEFT JOIN "foto" f4 ON f3."id_fotky"=f4."id_fotky"
WHERE SYSDATE BETWEEN "datum_zacatku" AND "rocnik"."datum_konce"
AND "id_skupiny" = :id
AND "id_kola" = (SELECT "id_kola" FROM "kola"
JOIN "rocnik"USING("id_rocniku")
WHERE "nazev_kola" LIKE 'skupiny'
AND SYSDATE BETWEEN "datum_zacatku" AND "datum_konce")
ORDER BY "zacatek_souboje2" ASC;
```

Poté jsou jednotlivé souboje postupně vypsaný. Pokud je nastavena metoda GET['idSoub'] a id souboje se shoduje s aktuálně vypisovaným soubojem, je k souboji připojen titulek, zda bylo hlasování úspěšné či nikoliv. Toto je rozhodnuto podle hodnoty v *\$hlasPred*.

Dalším krokem je ověření, jestli se u souboje má vypsat tlačítko pro hlasování. K tomu slouží stejná metoda, která slouží k samotnému hlasování. Jedná se o metodu *hlasuj*. Před zapsáním hlasu je v této metodě ověřeno, jestli uživatel může hlasovat a tohoto ověření využívá i metoda pro výpis soubojů. Pokud uživatel nemůže hlasovat, tlačítko se mu automaticky skryje. Tlačítko je také skryté, pokud souboj ještě není spuštěn.

7.8 Hlasování

Hlas se do databáze posílá pomocí metody *hlasuj*. Metoda v parametrech dostane id souboje, pro koho je hlas určen. Poslední parametr je pomocný a používá ho metoda pro výpis soubojů. Pokud je tento parametr false nebo null, jedná se o skutečné hlasování a ne pouze o kontrolu z metody *vypsatZapasy*.

Prvním krokem této metody je načtení posledních hlasů tohoto uživatele z databáze podle IP adresy nebo podle id uživatele. Poté, stejně jako v nominacích, se zkontroluje, jestli uživatel může hlasovat nebo jestli už v tento den hlasoval. Jestliže hlasovat může, je tento hlas vložen do tabulky *logy_hlasu*.

INSERT hlasu:

```
INSERT INTO "logy_hlasu"  
("id_logu","id_souboje","id_reg_uzivatele","pro_koho","cas_hlasu","ip_adr  
esa","id_zmeny")  
VALUES ("seq_logy".nextval,:soub,:uz,:pro,sysdate,:ip,:idZmen;
```

7.9 Statistiky a výpisy

Vypsání všech herců a seriálů v databázi obstarávají metody *vlozitHerce* a *vlozitSerial*. Tyto metody jsou podobné a velice jednoduché.

SELECT pro získání herců z databáze:

```
SELECT DISTINCT "id_herce", "jmeno_herce", "prijmeni_herce",  
"odkaz_na_biografii", "url_fotky"  
FROM "herci"  
LEFT JOIN "fotky_hercu" USING ("id_herce")  
LEFT JOIN "foto" USING ("id_fotky");
```

Výpis postav z databáze je složitější. Metoda pro výpis je volána ze dvou míst. Prvním místem je administrační rozhraní, ve kterém je výpis kdykoliv včetně počtu nominací. Druhým místem je uživatelské rozhraní, ve kterém nejsou přístupné počty nominací během kola nominace. Z administračního rozhraní je potřeba ke každé postavě připojit odkaz na editaci postavy.

SQL dotaz pro výpis postav v aktuálním ročníku:

```
SELECT "jmeno_postavy", COUNT("akt_nominace"."id_postavy") AS "pocet",  
"postup", "vlozeno", "nazev_original", "url_fotky",  
"postavy"."id_postavy", "id_rocniku"  
FROM "akt_nominace"  
JOIN "postavy" ON "akt_nominace"."id_postavy" = "postavy"."id_postavy"  
JOIN "rocnik" ON "akt_nominace"."id_rocniku" = "rocnik"."id_rocniku"  
JOIN "upraveno" ON "akt_nominace"."id_zmeny" = "upraveno"."id_zmeny"  
JOIN "serialy" ON "postavy"."id_serialu" = "serialy"."id_serialu"  
LEFT JOIN "fotky_postavy" ON "postavy"."id_postavy" =  
"fotky_postavy"."id_postavy"  
LEFT JOIN "foto" ON "fotky_postavy"."id_fotky" = "foto"."id_fotky"  
WHERE "akt_nominace"."id_rocniku" = (SELECT "id_rocniku" FROM "rocnik"  
WHERE SYSDATE BETWEEN "rocnik"."datum_zacatku" AND  
"rocnik"."datum_konce")  
GROUP BY "jmeno_postavy", "postup", "vlozeno", "nazev_original",  
"url_fotky", "postavy"."id_postavy", "id_rocniku"  
ORDER BY COUNT("id_postavy") DESC, "vlozeno" ASC;
```

Dále jsou ve výpisu vyznačeny postupující seriály. Těmto seriálům je do tagu přidáno id postavyPostup a v CSS je toto id nastaveno jako tučné. Dalším krokem je kontrola fotky. K postavám bez fotografie je připojena univerzální fotografie. Dále se rozhodne, jestli je po nominacích a pokud ano, jsou k postavám připojeny statistiky z tabulky

statistiky_postav. Je-li metoda volána z administrační části, je na konec připojen odkaz pro editaci postavy.

7.10 Získávání informací o postavách

Informace o postavách jsou čerpány z internetových stránek <www.serialzone.com>. K načtení HTML z jiných stránek je použit Simple_HTML_DOM_Parser. Stránky s konkrétním seriálem jsou načteny pomocí tohoto příkazu:

```
$html = file_get_html('http://www.serialzone.cz/hledani/?co="' . $serial . '"');
```

Z této stránky si aplikace stáhne potřebné informace o seriálu – český název, originální název, rok začátku a konce.

Kód, který hledá požadované údaje:

```
foreach ($html->find('div[class=col4in]', 0)->find('ul') as $ul) {  
    foreach ($ul->find('li') as $li) {  
        foreach ($li->find('a') as $odkazy) {  
            $ser = $li->plaintext;}}}
```

Nejdříve se najde na stránce div, který je třídy col4in. V tomto divu se najde tag ul. Pro každý ul tag se najde li tag a pro každý li tag se najde tag a. Z tagu se pomocí plaintext extrahuje všechny text, ze kterého se pomocí funkcí pro práci s řetězcem najdou všechny potřebné informace o seriálu. Tag a se použije pro načtení stránky s konkrétním seriálem a stejným způsobem se z této stránky načtou zbylé informace o postavě a herci.

7.11 Generování statistik a soubojů

O samotné generování soubojů a statistik se starají procedury v databázi. Je ale potřeba zajistit jejich spuštění. O to se starají metody ve třídě SoubojeTvorba. O stavu informuje session jménem „souboje“. Kontrola probíhá při každém načtení stránky. Pokud tato session není nastavena, je zavolána funkce *overeníSouboju*. Nastavená session má platnost 5 minut a po vypršení této platnosti je session zrušena

V metodě *overeníSouboju* je nejdříve spuštěn SQL dotaz, která vrátí jméno aktuálního kola. Pokud dotaz nevrátí nic, znamená to, že žádné kolo neběží a není potřeba nic generovat. V případě, že vrátí kolo se jménem „Nominace“ tak se také neprovádí žádné generování. Vráti-li ale něco jiného, je potřeba SQL dotazem ověřit, jestli už nejsou souboje vygenerovány.

SQL dotaz na ověření existence soubojů:

```
SELECT COUNT("id_souboje") AS "res"
FROM "souboje"
JOIN "kola" USING ("id_kola")
JOIN "rocnik" USING ("id_rocniku")
WHERE SYSDATE BETWEEN "zacatek_kola" AND "konec_kola"
AND SYSDATE BETWEEN "datum_zacatku" and "datum_konce"
AND "nazev_kola" <> 'nominace'
```

Je-li výsledkem dotazu číslo 0, tak žádné souboje v tomto kole nejsou a musejí být vygenerovány. V podmínce switch se rozhodne podle názvu kola jaké souboje a jaké statistiky generovat a spustí se procedury, které zajistí toto generování.

8 Vzhled a možnosti uživatele

Tato kapitola je zaměřena na ovládání aplikace a možnosti uživatele.

8.1 Vzhled

Jak je vidět na obrázku (Obr. 14) vzhled aplikace je minimalistický a jednoduchý. Díky tomu se stránky rychle načítají a jsou přehledné. Při návrhu stránek byla prioritní intuitivnost, aby uživatel nemusel nic hledat a se stránkami pracoval co nejpřirozeněji.



Obr. 14 – Vzhled aplikace

8.2 Nominace

První kolo, které v ročníku probíhá, je nominace. V této části uživatel nominuje postavy, které chce, aby se dostaly do hlavní části. Nominovat může libovolnou postavu, která hrála alespoň v jednom seriálu. Každou postavu může nominovat jednou denně. Postavy s největším počtem nominací od uživatelů postoupí, většinou postupuje 32 postav.

Uživatel má tři možnosti jak postavu nominovat. Všechny možnosti jsou na obrázku (Obr. 15). První možností je najít postavu pomocí seriálu, ve kterém hraje. K tomu slouží pole se šipkou č. 1. Druhou možností je najít postavu přesně podle jména v poli se šipkou č. 2. Poslední možností je nominovat postavu, která již byla nominována. Nezáleží, jestli jiným uživatelem nebo jím samotným. Nominované postavy se nacházejí v poli se šipkou č. 3.

The interface is titled "Zde můžete nominovat svoje oblíbené postavy" (Here you can nominate your favorite characters). It is divided into two main sections:

- Nominace postavy, podle seriálu nebo přímo postavy** (Nominate characters by series or directly by character):
 - Top section: "Zadejte jméno seriálu:" (Enter the name of the series) with a text input field and a "Hledej seriál" (Search series) button.
 - Bottom section: "Zadejte jméno postavy:" (Enter the name of the character) with a text input field and a "Hledej postavu" (Search character) button.
- Seznam nominovaných postav, které můžete také nominovat** (List of nominated characters, which you can also nominate):
 - Contains a "Nominuj" (Nominate) button.
 - Three red arrows labeled 1, 2, and 3 point from the search input fields to this list section.

Obr. 15 – Způsoby nominace

Pokud uživatel hledá postavu podle názvu seriálu, ve kterém hrála, může tento název odpovídat více seriálům. Uživatel je vyzván k výběru seriálu, který má na mysli. Tento výběr je vidět na obrázku (Obr. 16). Po této akci se zobrazí 20 postav, které se v seriálu vyskytovaly nejčastěji. Z těchto postav může uživatel nominovat libovolný počet.

The interface is titled "Vyberte seriál, ze kterého chcete nominovat postavy" (Select the series from which you want to nominate characters). It features a list of three radio button options:

- ☐ Garfield and Friends (1988 - 1995) - Garfield a přátelé
- ☐ Friends (1994 - 2004) - Přátelé
- ☐ Přátelé Zeleného údolí (1980 - 1980) - Přátelé Zeleného údolí

At the bottom right, there is a "Hledej postavy" (Search characters) button.

Obr. 16 – Nominace pomocí seriálu





8.3 Skupiny a vyřazování

Ve druhém kole jsou nominované postavy rozděleny do skupin. Ve skupinách se postavy utkají mezi sebou stylem „každý s každým“. Podle počtu postav ve skupině jsou vygenerovány souboje. Všechny souboje nejsou aktivní, ale konají se postupně, aby jedna postava nebyla současně ve dvou soubojích. Za vítězství jsou 3 body, za remízu 1 bod a za prohru 0 bodů. Postup ze skupiny je vyhodnocen podle počtu nasbíraných bodů.



Na obrázku (Obr. 17) je vidět podmenu se skupinami. V horní části jsou postavy, které jsou v dané skupině. V dolní části se nacházejí souboje mezi postavami s možností hlasovat.

Souboj seriálových postav

Hlavní stránka	Harmonogram	Souboje	Skončené souboje	Nominace	Soutěže	Postavy
Upravit hlavní stránku	Upravit harmonogram	Upravit soutěže	Upravit postavy	Upravit seriály	Upravit herce	
Skupina A	Skupina B	Skupina C	Skupina D	Skupina E	Skupina F	Skupina G

Foto	Postava	Serial	Vyhry	Remizy	Prohry	Body
	Dr. Eric Gablehauser	Big Bang Theory, The	0	3	0	3
	Leonard Hofstadter	Big Bang Theory, The	0	3	0	3
	"kapitán" George Van Smoot	How I Met Your Mother	0	3	0	3
	Gael	How I Met Your Mother	0	3	0	3

Zápasy ve skupině

	<div> <div></div> <div>50%</div> <div></div> </div>	
Leonard Hofstadter Big Bang Theory, The	Hlasování od 28.04.2012 do 28.04.2012 0 hlasů - 0 hlasů	Gael How I Met Your Mother

Obr. 17 - Ukázka skupin a soubojů

Souboje vyřazovací části jsou vytvořeny z postav, kterým se podařilo postoupit ze skupiny. Ve vyřazovací části probíhají všechny souboje současně a postava, která souboj vyhraje, postupuje do dalšího kola, kde se utká s ostatními vítězi minulého kola.

Do posledního kola postoupí i poražení, kteří se spolu utkají o 3. a 4. místo. Vítězové se utkají společně o post absolutního vítěze.

8.4 Soutěže a tipy na vítěze

Přihlášení uživatelé se mohou zúčastnit soutěží. Soutěže jsou na stránce „Soutěže“ a vypadají jako na obrázku (Obr. 18). Správná odpověď je vždy jen jedna. Každý přihlášený uživatel může hlasovat pouze jednou, po odhlasování se soutěž skryje. Výsledky a výherce budou zobrazeny po skončení soutěže na hlavní stránce.

Soutěž o tričko

Soutěž probíhá od 28.4.2012 do 3.5.2012




V jakém seriálu hraje postava Ted Mosby?

☐ HIMYM
 ☐ TBBT
 ☐ TWD
 ☐ HNTLYL

Obr. 18 – Ukázka soutěže

Registrovaní uživatelé mají také možnost tipovat vítěze. Mohou tipovat až tři různé postavy. Tipy nelze měnit, takže si uživatel musí pečlivě rozmyslet, koho bude tipovat. Uživatelé mohou tipovat kdykoliv, ale při vyhodnocování bude brán ohled na čas tipu.

Tipovat je možné na stránce postav, jak je vidět na obrázku (Obr. 19). Pokud na postavu lze tipovat, je zobrazeno tlačítko „Tip na Vítěze“. Pokud tipovat nelze, je zobrazen text s důvodem proč.

Pořadí	Foto	Jméno	Serial	Počet nominací	Výher	Remiz	Proher	Výher skupina	Remiz skupina	Proher skupina	Celkem hlasů	Tipovat
1.		pan Koothrappali	Big Bang Theory, The	2	0	0	0	0	0	0	0	<input type="button" value="Tip na Vítěze"/>
2.		Leonard Hofstadter	Big Bang Theory, The	2	0	0	0	0	0	0	0	Tuto postavu jste již tipoval
3.		Penny	Big Bang Theory, The	2	0	0	0	0	0	0	0	<input type="button" value="Tip na Vítěze"/>
4.		Gael	How I Met Your Mother	2	0	0	0	0	0	0	0	Tuto postavu jste již tipoval

Obr. 19 – Ukázka tipování

Závěr

Cílem práce byl návrh a realizace webového portálu o seriálových postavách a jejich obsazení herci s více kolovým hodnocením seriálů, epizod či seriálových postav.

Teoretická část popisovala internetové hlasování. V rámci této problematiky byly popsány způsoby zabezpečení hlasování, včetně identifikace způsobů a realizace, jak lze hlasování ovlivnit. Jak vyplývá z teoretické části, není možné vytvořit 100% bezpečné hlasování. Vždy se najde možnost, jak anketu ovlivnit. Cílem autora ankety je vytvořit takové hlasování, které by útočníka odradilo od pokusu o vícenásobné hlasování. Proto se hlasování vytváří takové, aby jeho ovlivnění více hlasy od jednoho uživatele bylo co nejobtížnější a co nejvíce časově náročné.

V teoretické části jsou dále popsány možnostmi vyhodnocování návštěvnosti internetových stránek a s termíny a technologiemi, které se pro to používají. Doporučeny jsou nepoužívanější nástroje na internetu a způsob, jak vybrat nejvhodnější nástroj.

Cílem praktické části bakalářské práce bylo vytvoření internetových stránek pro souboje seriálových postav. Webové stránky jsou určeny fanouškům seriálů, kteří zde mohou hlasovat pro své oblíbené seriálové postavy.

Aplikace byla vyvíjena tak, aby ji byl schopný používat i uživatel s méně zkušenostmi s počítačem. Stránky jsou tedy jednoduché a intuitivní. Webové stránky obsahují všechny základní funkce, které jsou potřeba pro běh v ostrém provozu. Před spuštěním do ostrého provozu je nutné stránky nejdříve spustit v testovacím provozu pro omezený počet uživatelů. Po skončení testovacího provozu se zjistí zpětná vazba od uživatelů, opraví se chyby a doplní se chybějící funkcionality. Poté může být aplikace spuštěna.

Možný budoucí vývoj je ve využití více interaktivních prvků pomocí javascriptu nebo využití technologie AJAX. Nová funkcionality může být například historie hlasování pro jednotlivé uživatele, přidání postavy mezi oblíbené nebo přímá spolupráce se serverem <serialzone.cz>.

Přechod do ostrého provozu je naplánován na začátek září roku 2012, kdy začíná seriálová sezóna.

Literatura

Adaptic, s.r.o. Proklik. *adaptic.cz*. [Online] [Citace: 19. 4 2012.] <http://www.adaptic.cz/znalosti/slovnicek/proklik/>.

Adobe Systems Incorporated. 2009. Omniture acquisition. *omniture.com*. [Online] 26. 8 2009. [Citace: 19. 4 2012.] http://www.omniture.com/en/company/adobe_faq.

Blueboard.cz. Nejčastější dotazy. *blueboard.cz*. [Online] [Citace: 20. 4 2012.] <http://miniaplikace.blueboard.cz/caste-dotazy#anketa>.

Bright, Doug. 2010. E-Commerce Web Analytics Market Share. *istobe.com*. [Online] 25. 1 2010. [Citace: 19. 4 2012.] <http://istobe.com/blog/2010/01/25/e-commerce-web-analytics-market-share-january-2010/>.

Čech, Martin. 2010. Nástroje webové analytiky. *inflow.cz*. [Online] 6. 6 2010. [Citace: 19. 4 2012.] <http://www.inflow.cz/nastroje-webove-analytiky>.

Čížek, Jakub. 2009. TOR: Skutečně anonymní internet. *zive.cz*. [Online] 2. 10 2009. [Citace: 18. 4 2012.] <http://www.zive.cz/clanky/tor-skutecne-anonymni-internet/sc-3-a-149055/default.aspx>.

Dobiáš, Richard. 2011. Co upoutá pozornost čtenáře. *o-psani.cz*. [Online] 30. 6 2011. [Citace: 28. 4 2012.] <http://www.o-psani.cz/2011/06/co-upouta-pozornost-ctenare.html>.

Gerds, Eric. 2012. Browser Plugin Detection with PluginDetect. *pinlady.net*. [Online] 11. 4 2012. [Citace: 26. 4 2012.] <http://www.pinlady.net/PluginDetect/>.

Google Inc. . 2012. Návod Analytics. *google.com*. [Online] 2012. [Citace: 19. 4 2012.] <http://support.google.com/googleanalytics/bin/answer.py?hl=cs&answer=57164>.

GUTMAS, A., RETHANS, D. a BAKKEN, S. 2007. *Mistrovství v PHP 5*. místo neznámé : Computer Pres, 2007. ISBN 978-80-251-1519-0.

HARRINGTON, J., L. 2010. *SQL clearly explained*. 3.vyd. místo neznámé : Elsevier, 2010. ISBN 978-0-12-375697-8.

HERNANDEZ, J., M., Viescas, L., J. 2004. *Myslíme v jazyku SQL: Tvorba dotazů*. místo neznámé : Grada, 2004. ISBN 978-80-247-0899-7.

Hulán, Radek. 2007. Jak ovlivnit libovolné hlasování na iDNES.cz. *MyEgo.cz*. [Online] 16. 8 2007. [Citace: 18. 4 2012.] <http://myego.cz/item/jak-ovlivnit-libovolne-hlasovani-na-idnes-cz>.

iPerceptions inc. 2010. Who Runs Analytics? Top 500 Retail Websites Report. *WASP Blog*. [Online] 24. 1 2010. [Citace: 19. 4 2012.] <http://blog.webanalyticssolutionprofiler.com/2010/02/who-run-analytics-top-500-retail-web-site-report/>.

Janovský, Dušan. 2012. Úvod do JavaScriptu. *jakpsatweb.cz*. [Online] 3. 4 2012. [Citace: 24. 4 2012.] <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>.

Karban, Radek. 2009. Měření a vyhodnocování návštěvnosti webu nástroji Google. *slideshare.net*. [Online] 23. 1 2009. [Citace: 19. 4 2012.] <http://www.slideshare.net/seoexpertcz/men-a-vyhodnocovn-nvtvnosti-webu-nstroji-google-presentation>.

Kebrt, Michal. 2002. Jak vědět více o návštěvnících pomocí PHP. *interval.cz*. [Online] 9. 1 2002. [Citace: 2012. 4 2012.] <http://interval.cz/clanky/jak-vedet-vice-o-navstevnicich-pomoci-php/>.

Lahvička, Jiří. 2000. Hlasovací systémy na Internetu. *Lupa.cz*. [Online] 6. 6 2000. [Citace: 2012. 4 18.] <http://www.lupa.cz/clanky/hlasovaci-systemy-na-internetu/>.

Nette Foundation. 2012. Formuláře. *nette.org*. [Online] 2012. [Citace: 27. 4 2012.] <http://doc.nette.org/cs/forms>.

—. **2012.** Zabezpečení před zranitelnostmi. *nette.org*. [Online] 2012. [Citace: 27. 4 2012.] <http://doc.nette.org/cs/vulnerability-protection#toc-cross-site-scripting-xss>.

—. **2012.** Začínáme s Nette Frameworkem. *nette.org*. [Online] 2012. [Citace: 24. 4 2012.] <http://doc.nette.org/cs/getting-started>.

Optimics s.r.o. Adobe/Omniture SiteCatalyst. *optimics.cz*. [Online] [Citace: 19. 4 2012.] <http://www.optimics.cz/sluzby/webova-analytika/adobe-omniture-sitecatalyst>.

Plotěný, Luboš. 2009. Newsletter: Kam návštěvníci klikají. *dobryweb.cz*. [Online] 25. 5 2009. [Citace: 28. 4 2012.] <http://blog.dobryweb.cz/newsletter-kam-navstevnici-klikaji>.

Rajavu. 2008. How to get factorial in oracle. *club-oracle.com*. [Online] 14. 10 2008. [Citace: 26. 4 2012.] <http://www.club-oracle.com/articles/how-to-get-factorial-in-oracle-159/>.

Schafer, Steven M. 2009. *HTML, XHTML a CSS – Bible pro tvorbu www stránek*. 4. vyd. místo neznámé : Grada, 2009. ISBN 978-80-247-2850-6.

Vrána, Jakub. 2009. Detekce sítě Tor. *php.vrana.cz*. [Online] 5. 10 2009. [Citace: 18. 4 2012.] <http://php.vrana.cz/detekce-site-tor.php>.

—. **2006.** Kontrola data. <http://php.vrana.cz>. [Online] 7. 4 2006. [Citace: 3. 5 2012.] <http://php.vrana.cz/kontrola-data.php>.

—. **2006.** PHP okénko: Unikátnost návštěvníka. *root.cz*. [Online] 19. 5 2006. [Citace: 18. 4 2012.] <http://www.root.cz/clanky/php-unikatnost-navstevnika/>.

Weida, Petr. 2011. Školení webová analytika. *slideshare.net*. [Online] 4. 1 2011. [Citace: 19. 4 2012.] <http://www.slideshare.net/ACOMWARE/kolen-webov-analytika-6810733>.

Wikimedia Foundation Inc. 2011. Konverzní poměr. *wikipedia.org*. [Online] 14. 11 2011. [Citace: 19. 4 2012.]

http://cs.wikipedia.org/wiki/Konverzn%C3%AD_pom%C4%9Br.

—. **2012.** Proxy server. *wikipedia.org*. [Online] 30. 3 2012. [Citace: 18. 4 2012.]

http://cs.wikipedia.org/wiki/Proxy_server.

—. **2012.** Webová analytika. *wikipedia.org*. [Online] 27. 2 2012. [Citace: 19. 4 2012.]

http://cs.wikipedia.org/wiki/Webov%C3%A1_analytika.

Zatomik. 2012. Přihlášení přes Facebook (PHP). *webbusiness.cz*. [Online] 7. 4 2012.

[Citace: 26. 4 2012.] <http://www.webbusiness.cz/2012/04/prihlaseni-pres-facebook/>.

Žák, David. 2011. Přednáška k předmětu KIT/IDAS1. [Online] 23. 3 2011. [Citace: 24. 4 2012.] <https://portal.upce.cz/StagPortletsJSR168/StudijniMaterialyServlet?id=483905>.

Příloha A – ER-diagram

