

# Informace ve Zlínském kraji

Zlin region information

Svatopluk Hastík

---

Bakalářská práce  
2007



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2006/2007

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Svatopluk HASTÍK**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
  
Téma práce: **Informace ve Zlínském kraji**

Zásady pro vypracování:

1. Analýza současného stavu v oblasti informačních serverů.
2. Výběr vhodného programového prostředí a jeho nastudování.
3. Návrh struktury nového informačního serveru a podpůrné databáze.
4. Vytvoření funkční aplikace.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

ULLMAN, L. PHP pokročilé programování pro World Wide Web. SoftPress, 2002, ISBN: 80-86497-36-4.

ULLMAN, L. PHP a MySQL. Computer Press, Brno, 2004, ISBN: 80-251-0063-4.

HOLČÍK, T. a kol. 1001 tipů a triků pro WWW stránky. Computer Press, 2006, ISBN: 80-7169-610-2.

Kolektiv autorů. PHP5, MySQL, Apache- vytváříme webové aplikace. Computer Press, 2006, ISBN: 80-251-1073-7.

STEJSKAL, J. Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu. Computer Press, 2006, ISBN: 80-251-0167-3.

Vedoucí bakalářské práce: **Ing. Zdenka Prokopová, CSc.**  
Ústav aplikované informatiky

Datum zadání bakalářské práce: **13. února 2007**

Termín odevzdání bakalářské práce: **24. května 2007**

Ve Zlíně dne 13. února 2007



prof. Ing. Vladimír Vašek, CSc.  
*děkan*



doc. Ing. Ivan Zelinka, Ph.D.  
*ředitel ústavu*

## **ABSTRAKT**

Cílem této práce je vytvořit informační portál, který by shromažďoval, veškeré akce, které se dějí ve Zlínském kraji. Hlavní myšlenkou bylo vytvořit tento systém otevřený všem uživatelům, kteří by jej obohacovali vlastními novinkami. Uživatelé se zde dozvědí nejen o akcích, které se připravují, ale i o akcích, které už proběhly a mají možnost s ostatními návštěvníky webu tyto akce hodnotit a komentovat.

Klíčová slova: informační systém, databáze akcí, SQL, uživatelé, PHP

## **ABSTRACT**

The main point of this work is to create informative portal which would gather all actions that are taking place in Zlín region. The main idea was to create open system for all users, who would fill it by their own news. Users can be notified about new actions which are being prepared and also about past actions. Users can discuss about actions with other web's visitors and also they can rate them.

Keywords: informative system, database of actions, SQL, users, PHP

Na tomto místě bych chtěl poděkovat Ing. Zdence Prokopové, CSc. za ochotu, připomínky, cenné rady a vedení při tvorbě práce.

Prohlašuji, že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně 9.5.2007

.....

Svatopluk Hastík

**OBSAH**

<b>ÚVOD</b> .....	<b>8</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>9</b>
<b>1 DATABÁZOVÝ SYSTÉM MYSQL</b> .....	<b>10</b>
1.1 JAZYK SQL .....	10
1.1.1 SQL příkazy .....	11
1.1.1.1 Příkazy pro manipulaci s daty .....	11
1.1.1.2 Příkazy pro definici dat .....	11
1.1.1.3 Příkazy pro řízení dat .....	11
1.1.1.4 Ostatní příkazy .....	11
1.2 RELAČNÍ DATABÁZE .....	11
1.2.1 Vztahy mezi tabulkami.....	12
1.3 DATABÁZOVÉ NORMY .....	12
1.3.1 První normální forma (1.NF) .....	13
1.3.2 Druhá normální forma (2.NF) .....	13
1.3.3 Třetí normální forma (3.NF) .....	13
<b>2 PHP</b> .....	<b>14</b>
2.1 OBJEKTOVÉ PROGRAMOVÁNÍ V PHP .....	14
2.2 TERMINOLOGIE.....	15
2.2.1 Třída .....	15
2.2.2 Konstruktor .....	15
2.2.3 Objekt .....	16
2.2.4 Metody .....	16
2.3 PROPOJENÍ PHP + MYSQL .....	16
2.3.1 Připojení k databázi .....	16
2.3.2 Výběr dat .....	17
2.3.3 Ostatní užitečné funkce .....	18
<b>II PRAKTICKÁ ČÁST</b> .....	<b>19</b>
<b>3 ANALÝZA SOUČASNÉHO STAVU</b> .....	<b>20</b>
3.1 INFORMAČNÍ SERVERY V ČESKÉ REPUBLICE .....	20
3.2 POHLED UŽIVATELE.....	20
3.3 STANOVENÍ PRIORITY .....	21
3.3.1 Přehlednost .....	21
3.3.2 Obsahová kvalita .....	21
<b>4 VYTVOŘENÍ INFORMAČNÍHO SERVERU</b> .....	<b>22</b>
4.1 NÁVRH DATABÁZE .....	22
4.1.1 Tabulka <i>akce</i> .....	23
4.1.2 Tabulka <i>uzivatel</i> .....	24
4.1.3 Tabulky <i>diskuze</i> .....	24
4.1.4 Tabulka <i>kategorie</i> .....	25
4.1.5 Tabulka <i>mista_konani</i> .....	26

4.1.6	Tabulka <i>statusy_akci</i> .....	26
4.1.7	Tabulka <i>hodnosti</i> .....	26
4.1.8	Tabulky <i>faktor_natesenosti, hodnoceni_akce</i> .....	27
4.1.9	Tabulka <i>oblibene</i> .....	27
4.1.10	Ostatní tabulky .....	28
4.2	REGISTRACE, PŘIHLÁŠENÍ DO SYSTÉMU .....	28
4.3	HODNOSTI UŽIVATELŮ .....	29
4.4	STATUSY AKCÍ .....	30
<b>5</b>	<b>VLASTNOSTI WEBOVÉHO PORTÁLU .....</b>	<b>31</b>
5.1	VLOŽENÍ, EDITACE NOVÉ AKCE .....	31
5.2	STRUKTURA WEBU .....	33
5.3	VÝBĚR AKCÍ Z DATABÁZE .....	34
5.4	UPLYNULÉ AKCE, ARCHIV AKCÍ .....	34
5.5	STATISTIKY .....	35
5.6	BEZPEČNOSTNÍ OPATŘENÍ .....	36
5.7	MOŽNOSTI NASTAVENÍ .....	36
	<b>ZÁVĚR .....</b>	<b>37</b>
	<b>ZÁVĚR V ANGLIČTINĚ .....</b>	<b>38</b>
	<b>SEZNAM POUŽITÉ LITERATURY .....</b>	<b>39</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>40</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>41</b>
	<b>SEZNAM TABULEK .....</b>	<b>42</b>
	<b>SEZNAM PŘÍLOH .....</b>	<b>43</b>

## ÚVOD

Internet je hlavním elektronickým médiem, které nabízí prakticky neomezené množství informací. Ale i přesto se dají nalézt obory, ve kterých je informací nedostatek. Při hledání akcí ve Zlínském kraji je problém najít informační portál, který by je shromažďoval všechny na jednom místě. Cílem této práce bylo tento informační server vytvořit. Informační portál Zlínsko Žije je dostupný na adrese <http://www.zlinsko.zije.info>.

Nesporným faktem je, že uživatelé se na web vrací za informacemi, proto je nutno udržovat zvláště informační portál často aktualizován. Informační portál Zlínsko Žije je otevřený všem uživatelům internetu a kdokoliv může do databáze přispět svou akcí.

Nejprve byla provedena analýza serverů ve Zlínském kraji, která jen utvrdila skutečný fakt, že nalezené portály se zaměřovaly pouze na jednu kategorickou nebo regionální oblast. Portálům, které mají sloužit jako univerzální weby, chybí zase častá aktualizace. Jednoduchý systém vkládání nových akcí byl hlavní podmínkou vzniku serveru Zlínsko Žije. Uživatelé si tvoří tento portál, uživatelé informují sebe i ostatní.

Celý projekt je naprogramován v populární webovém jazyku PHP a využívá schopností databázového systému MySQL. Tato kombinace byla zvolena, protože se hodí na použití do středně velkých projektů a navíc je velmi často používána u hostingových firem.

Ve výsledku byl vytvořen informační portál, do něhož může kdokoliv vložit novou akci, které lze při vyhledávání třídit podle okresu konání, podle kategorie nebo podle datumu. Registrovaní uživatelé mají možnost přidávat si akce do oblíbených, hodnotit a diskutovat o akcích. Všichni mají možnost sledovat podrobné statistiky akcí. Administrátor má plnou kontrolu nad celým portálem, může akce upravovat nebo mazat.



## **I. TEORETICKÁ ČÁST**

## 1 DATABÁZOVÝ SYSTÉM MYSQL

Databázový systém MySQL byl vyvinut švédskou softwarovou firmou MYSQL AB. Díky tomu, že se jedná o volně šiřitelný software se stal spolu s kombinací programovacího jazyka PHP a serveru Apache nejpoužívanějším webovým databázovým systémem a uvedená kombinace je nejčastěji nabízenou formou pro hostování webových aplikací.

Komunikace s MySQL probíhá pomocí dotazovacího jazyka SQL. Aktuální verze MySQL je 5.0.27. Vzhledem k možnostem hostingu jsem však pro práci využíval MySQL verzi 4.

Nesporným kladem MySQL je její rozšířenost a to nejen díky tomu, že se jedná o volně šiřitelný software. Mezi další její vyzdvihované klady jsou její rychlost a pro mnoho začátečníků jednoduchost. Naopak kritické hlasy se ozývají z hlediska funkčního vybavení, avšak s novými verzemi tento zápor ztrácí na významu. Z hlediska zjištěných faktů lze usoudit, že se MySQL hodí do středních a středně velkých projektů.

I když má MySQL mezi volně šiřitelnými webovými databázemi výsadní postavení, je vhodné zmínit, že existují i konkurenční projekty jako PostgreSQL, mSQL nebo FireBird. V oblasti komerčního placeného softwaru se pak používají databáze jako např. Oracle nebo MS SQL Server.

### 1.1 Jazyk SQL

SQL je standardizovaný dotazovaný jazyk, jehož počátky spadají do 70. let 20. století. Hlavní myšlenkou k vytvoření, bylo vytvořit sadu příkazů pro ovládání databáze, které by se syntaxí co nejvíce podobaly přirozenému jazyku (angličtině). Z počátku se jednalo o nestandardizovaný jazyk, avšak s dalším vývojem nových databázových systémů jej bylo nutné standardizovat. První standard vyšel v roce 1986 a je označován jako SQL-86. Od té doby vyšlo několik verzí jazyka SQL. Za nejnovější lze považovat verzi SQL2003, avšak většina databázových systémů používá stále verzi SQL-99.

### 1.1.1 SQL příkazy

Příkazy jazyka SQL lze rozdělit do několika základních skupin:

#### 1.1.1.1 Příkazy pro manipulaci s daty

Do této skupiny lze zařadit příkazy, jenž slouží pouze k manipulaci položek, které se nacházejí v databázi.

SELECT (výběr dat)  
INSERT (vkládání dat)  
UPDATE (aktualizace dat)  
DELETE (mazání dat)

#### 1.1.1.2 Příkazy pro definici dat

V této skupině se nacházejí příkazy, jimiž lze ovlivňovat strukturu databáze.

CREATE (vytvoření objektu)  
ALTER (úprava objektu)  
DROP (odstranění objektu)

#### 1.1.1.3 Příkazy pro řízení dat

Do této skupiny patří příkazy, pro nastavování uživatelských práv a řízení transakcí.

GRANT (přidělení práv uživateli)  
REVOKE (odnětí práv uživateli)  
BEGIN (zahájení transakce)  
COMMIT (potvrzení transakce)  
ROLLBACK (zrušení transakce)

#### 1.1.1.4 Ostatní příkazy

V této skupině se nachází příkazy pro správu databáze. Tato skupina příkazů není standardizována a konkrétní syntaxe příkazů závisí na použitém databázovém systému. Například se zde nacházejí příkazy pro změnu kódování, způsob řazení, formát času apod.

## 1.2 Relační databáze

MySQL patří do skupiny relačních databází, nejpoužívanějších databázových systémů současnosti. Tím se myslí, že všechna data v databázi jsou uložena v tabulkách o uspořádaném počtu sloupců a řádků. Jelikož relace je množina, není možné definovat, který řádek v databázi je první, či poslední. Řádky v relaci nejsou uspořádány ve stálém

pořadí, proto je nutno v každé relaci (tabulce) vytvořit speciální sloupec, který bude zastávat úlohu primárního klíče, jímž bude každý řádek tabulky jasně identifikovatelný. Každá tabulka musí mít i vlastnosti (atributy), jenž budou jednotlivé prvky tabulky charakterizovat. Těchto vlastností může mít každá tabulka libovolně mnoho. Při samotném zpracování dat v relačních databázích není důležité znát fyzický způsob uložení dat, ale stačí znát nástroje, jimiž můžeme databázi spravovat.

### 1.2.1 Vztahy mezi tabulkami

V každé databázi se obvykle nachází více tabulek a ve většině případů nastávají situace, kdy je nutné tyto tabulky spojit (utvořit mezi nimi vztahy). Ať už kvůli dodržení normálních forem je potřeba vytvořit další tabulky, které se propojí pomocí vztahu 1:N nebo M:N.

K vytvoření vazby 1:N postačí dvě tabulky. Pro příklad lze uvést tabulku *akce*, která bude obsahovat sloupec *misto\_konani* a druhou tabulkou bude tabulka *mista*, ve které budou uloženy jednotlivé názvy obcí. V tomto případě lze tento vztah definovat tak, že jedna akce může mít pouze jedno místo konání z tabulky *mista*, zatímco jakékoliv místo z tabulky *mista* je možno přiřadit libovolným akcím (proto vztah 1:N). Poté primární klíče z tabulky *mista* se budou do tabulky *akce* ukládat jako klíče cizí. Právě cizí klíč spolu s primárním klíčem zprostředkovává vazbu mezi tabulkami.

K vytvoření vazby M:N jsou potřeba minimálně 3 tabulky. Pro představu budeme mít tabulku *akce* s primárním klíčem *Akce\_ID* a tabulku *uzivatele* s primárním klíčem *Uziv\_ID*. Nyní lze pro příklad vytvořit funkci, která by zaznamenávala hodnocení akcí jednotlivými uživateli. Abychom tyto hodnoty mohli nějak účinně spravovat je nutné vytvořit třetí tabulku *hodnoceni\_akce*, která sama nemusí mít žádný primární klíč, ale má dva atributy (sloupce), jenž jsou cizími klíči tabulek *akce* a *uzivatele*. Podle tohoto příkladu lze definovat vztah M:N tak, že každý uživatel může hodnotit jakoukoliv akci a každá akce může být hodnocena kterýmkoliv uživatelem.

## 1.3 Databázové normy

Při stavbě databáze je nutné dodržovat určité základní pravidla, které potom usnadňují výběr, vkládání a samotnou aktualizaci databáze. Tato pravidla se nazývají normalizace

databáze a shlukují se do několika normálních forem (1.NF, 2.NF, 3.NF, BCNF, 4.NF, 5.NF).

### 1.3.1 První normální forma (1.NF)

První normální forma říká, že každý atribut (sloupec) relace musí obsahovat pouze jednu hodnotu. Jako příklad lze uvést tabulku, která by obsahovala sloupec s názvem *jmeno*, ve kterém by bylo ale uloženo celé jméno uživatele. (např. Jan Novák). Aby se tato tabulka mohla nacházet v 1.NF, musela by být rozdělena na sloupce *jmeno* a *prijmeni*, ve kterých by byla data uložena jednotlivě.

### 1.3.2 Druhá normální forma (2.NF)

Relace se nachází ve druhé normální formě, pokud se nachází v první normální formě a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině.

Vysvětlení na příkladu. Pokud by existovala např. tabulka *akce*, do které by se ukládaly nejen informace o dané akci (jako např. její název, datum konání), ale také údaje o uživateli, který akci uložil (jméno uživatele, e-mail, telefon..), tak by nastal problém při smazání nebo aktualizaci údajů. Protože v okamžiku, kdy by se nějaká akce z tabulky smazala, ztratily by se z databáze i údaje o uživateli, který akci do databáze vložil. Tato relace (tabulka) by se nenacházela v 2.NF. Aby tomu tak bylo, je nutno vytvořit další tabulku *uzivatel*, která by seskupovala údaje o uživateli a s tabulkou *akce* by byla propojena pomocí cizího klíče, jenž by byl primárním klíčem v tabulce *uzivatel*.

### 1.3.3 Třetí normální forma (3.NF)

Aby se relace nacházela ve třetí normální formě, tak se musí nacházet v 1.NF a 2.NF, všechny neklíčové atributy jsou navzájem nezávislé a žádný z jejich atributů není tranzitivně závislý na klíči.

*Tranzitivní závislost* je taková závislost, mezi minimálně dvěma atributy a klíčem, kde jeden atribut je funkčně závislý na klíči a druhý atribut je funkčně závislý na prvním atributu.

Aby relace splňovala 3.NF musí se takové funkční závislosti, které jsou tranzitivní, oddělit do samostatných tabulek.

## 2 PHP

PHP je skriptovací jazyk určený pro tvorbu webových aplikací. PHP skripty jsou prováděny na straně serveru (obvykle to bývá webový server Apache) a teprve poté jsou odeslány do uživatelova prohlížeče. PHP se stal oblíbeným jazykem především pro jeho snadné pochopení a díky syntaxi, která kombinuje jiné programovací jazyky (C, Pascal, Java).

V současné době nabízejí poskytovatelé hostingových služeb na svých serverech PHP verzi 4 a novější verzi 5. Většina skriptů naprogramovaných pro PHP 4 funguje i v PHP 5. Zásadní rozdíl je ale u objektového programování, které ve verzi 5 dosáhlo značných změn.

### 2.1 Objektové programování v PHP

Objektové programování v PHP je nejvyspělejší od verze 5, jelikož jsem ale vyvíjel svůj projekt v PHP 4, zmíním se zde o objektovém programování právě pro tuto verzi. Objektové programování v PHP není to samé objektové programování, které používá např. C jazyk. Zásadním problémem může být to, že PHP neprovádí kontrolu syntaxe objektu a na chybu upozorní až na ni při zpracování narazí.

Základním prvkem objektového programování je stejně jako u ostatních jazyků nadefinování třídy. Třída se musí nadefinovat před vytvořením samotného objektu. Nepsaným pravidlem je, že třída má být nadefinována v samostatném souboru, mimo kořenový adresář serveru a do samotné stránky je potom vložena pomocí direktiv *include* nebo *require*.

Definici třídy provádí klíčové slovo *class*. Ve vytvořené třídě lze potom nadefinovat jednotlivé proměnné a metody, jimiž je třída charakterizována. Na následujícím příkladě je ukázka definice třídy *Akce*.

```
class Akce() {
    var $nazev;
    var $obsah;

    function UlozAkci() {
        $vlozit_akci = mysql_query ("
            INSERT INTO akce
            VALUES('".$this->nazev."','".$this->obsah."'
            ") or die (mysql_error());
    }
}
```

Třída *Akce* je charakterizována proměnnými *nazev* a *obsah*, jenž vyjadřují vlastnosti třídy a jednou metodou *UlozAkci*, která vyvolá určitou událost (V tomto případě uloží vlastnosti třídy do databáze).

Pokud je třída nadefinována, tak můžeme vytvořit instanci objektu, který bude využívat vlastnosti třídy. To se provádí klíčovým slovem *new*.

```
Např. $akce = new Akce();
```

Nyní může být proměnná *\$akce* použita pro volání funkcí nebo výpis proměnných, které se vyskytují uvnitř třídy *Akce*.

```
$akce->nazev = "Nová akce";  
$akce->obsah = "Informace o nové akci";  
$akce->UlozAkci();
```

Pro přístup k proměnným nebo k metodám uvnitř třídy se používá operátoru '*->*'. Na příkladu definice třídy *Akce* je také vidět použití operátoru *\$this*. Tento operátor se používá při volání funkce nebo proměnné uvnitř jedné třídy. Nelze ho tedy použít mimo definici dané třídy.

## 2.2 Terminologie

### 2.2.1 Třída

Třída je šablonou objektu. Tím se myslí, že při vytvoření objektu třída shlukuje veškeré data, se kterými objekt pracuje. Daty se míní vlastnosti vytvořeného objektu. Třída se definuje pomocí klíčového slova *class*. Jednotlivé vlastnosti (atributy) třídy se musí nadefinovat pomocí klíčového slova *var*. Pro přístup k atributům objektu uvnitř třídy se používá operátoru '*->*' a klíčového slova *\$this*.

### 2.2.2 Konstruktor

Konstruktor je speciální metoda, kterou by měla obsahovat každá třída. V PHP se definuje stejně jako funkce, rozdíl oproti ostatním funkcím je ten, že se jmenuje stejně jako třída. Konstruktor se vyvolává pokaždé při vytvoření nového objektu, tudíž se dá použít např. k nastavení počátečních proměnných.

### 2.2.3 Objekt

Aby se dalo pracovat s jednotlivými metodami uvnitř nadefinované třídy je nutno vytvořit objekt. V PHP se k tomu používá speciální operátor *new*. Po nadefinování objektu lze potom pomocí operátoru *'->'* přistupovat k jednotlivým metodám třídy.

### 2.2.4 Metody

Aby třída měla nějaký smysl a nějakou funkci je nutné aby obsahovala nějaké metody (funkce). Tyto metody a funkce se definují stejně jako obyčejné funkce v php pomocí klíčového slova *function*. Pro volání metod uvnitř třídy se používá klíčové slovo *\$this*.

## 2.3 Propojení PHP + MySQL

Když se řekne PHP + MySQL, lze s přesvědčením tvrdit, že se jedná o jednu z nejčastěji používaných kombinací mezi programovacím jazykem a databázovým serverem na poli internetu. V PHP4 jsou pro připojení a práci s databází vytvořeny speciální funkce. V nové verzi PHP5 bylo vyvinuto objektové rozhraní *mysqli*, jenž dokáže pracovat s databází objektově (To lze samozřejmě i v PHP4, ale jen za cenu vytvoření vlastních obslužných tříd). Funkcí, které slouží k práci s databází v PHP je velké množství. V následujícím souhrnu uvedu jen ty nejdůležitější.

### 2.3.1 Připojení k databázi

Nejprve ukáží příklad souboru, jenž slouží k připojení k databázi. Je vhodné připojovací funkce oddělit do vlastního souboru, ať už z hlediska bezpečnosti, či z hlediska možné pozdější úpravy při změně přihlašovacích údajů. Z hlediska bezpečnosti je doporučeno neukládat připojovací soubor k databázi do kořenového adresáře.

```
<?php
$dbhost = 'localhost';
$dbusername = 'root';
$dbpasswd = '123456';
$database_name = 'zlinskozije';
$connection = mysql_connect("$dbhost","$dbusername","$dbpasswd")
               or die ("Není možné připojit databázový server.");
$db = mysql_select_db("$database_name", $connection)
     or die ("Není možné vybrat databázi.");
?>
```



Z předchozího příkladu se skript nejprve pomocí funkce *mysql\_connect* připojí k databázi. Funkce *mysql\_connect* má více parametrů, ale pro připojení jsou nejdůležitější *\$dbserver* (název serveru), *\$dbusername* (uživatelské jméno), *\$dbpasswd* (uživatelské heslo). Do proměnné *\$dbhost* stačí na většině serverů nechat vyplněnu položku *localhost*. Pokud je ovšem databázový server spuštěn na jiném počítači než webový server, je nutno vyplnit jiné jméno. Do položek *\$dbusername* a *\$dbpasswd* je nutno vyplnit přihlašovací údaje k databázi. Pokud jsou všechny údaje zadány správně funkce vrátí hodnotu *TRUE*. V tomto případě ji uloží do proměnné *\$connection*. V případě neúspěšného připojení vrátí *FALSE*, ukončuje skript a zobrazuje chybovou hlášku.

Další důležitou funkcí je *mysql\_select\_db*. Tuto funkci je možné použít až po připojení k databázi. Funkce má dva parametry, název databáze a odkaz na spojení, jímž je v tomto případě proměnná *\$connection*. V případě úspěšného výběru databáze vrátí *TRUE*, v opačném případě vrátí *FALSE*.

Připojení k databázi končí automaticky s ukončením skriptu. Pokud je potřeba ukončit připojení dříve, stačí použít funkci *mysql\_close()*.

### 2.3.2 Výběr dat

Pokud byl připojovací skript k databázi úspěšně proveden je možné spolupracovat s databází formou příkazů jazyka SQL. Aby bylo tyto příkazy možné použít, musí se nejprve odeslat přes webový server, v tomto případě přes PHP, pomocí speciální funkce.

Tato funkce se jmenuje *mysql\_query()* a provede SQL dotaz na aktuálním spojení v databázi a vrátí identifikátor výsledku.

```
$vysledek = mysql_query("SELECT * FROM akce") or die (mysql_error());
```

Na předchozím příkladu je volán SQL dotaz, jenž vybere všechny položky s tabulky akce. V případě, úspěchu vrátí *TRUE* a ukládá identifikátor výsledku do proměnné *\$vysledek*. V případě vzniku chyby vrátí *FALSE* a funkce *mysql\_error()* vrátí text chybové zprávy.

Nyní je v proměnné *\$vysledek* identifikátor výsledku skriptu, ale stále se s nimi nedá přímo pracovat. Proto je nutné uložit všechna nalezená data z databáze do pomocného pole. Tuto operaci vykonává funkce *mysql\_fetch\_array()*.

```
$obsah = mysql_fetch_array($vysledek);
```

Proměnná *\$obsah* nyní reprezentuje pole hodnot načteného výsledku. Jednoduše řečeno, v proměnné *\$obsah* jsou nyní všechna data z databáze, která vrátil SQL skript, jehož identifikátor byl uložen v proměnné *\$vysledek*. Nyní lze s proměnnou *\$obsah* pracovat stejně jako s poli v PHP.

### 2.3.3 Ostatní užitečné funkce

PHP má v sobě pro práci s databází MySQL integrováno 52 funkcí. Většina z nich funguje na principu *mysql\_fetch\_array()*, tím myslím, že slouží ke zpracování identifikátoru, jenž vygeneruje funkce *mysql\_query()*. Např. mohu zmínit funkci *mysql\_num\_rows()*, jenž vrací počet nalezených řádků (položek) po vyvolání SQL skriptu.

## **II. PRAKTICKÁ ČÁST**

### 3 ANALÝZA SOUČASNÉHO STAVU

#### 3.1 Informační servery v České Republice

Nejprve jsem hledal informační servery, které se zaměřují na celou Českou Republiku. Nalezl jsem několik takových serverů, které slouží jako veřejné portály akcí. Lze do nich akce vkládat, lze v nich vyhledávat, vše je členěno do kategorií.

Ve Zlínském kraji existuje také několik podobných systémů. Pokud se však zaměřím na informační servery do nichž může uživatel sám jednoduše vložit novou akci, tak jich naleznu velmi málo. Hlavním problémem bývá většinou obsahová kvalita a mnohdy také nepřehlednost při vkládání nové akce, či samotném hledání požadované akce.

U větších informačních portálů jsem si také všiml faktu, že se spíše zaměřují na akce ve velkých městech (což je z jedné strany pochopitelný fakt), ale na druhé straně tím odsunují na druhou kolej menší obce. Právě toto byl také jeden z důvodů, proč jsem se rozhodl vytvořit takový portál, který by byl volně otevřený pro každého uživatele, který by byl přehledný, a ve kterém by šlo jednoduše vyhledávat.

#### 3.2 Pohled uživatele

Ze statistik vyplívá, že uživatel, při prvním pohledu na otevřenou webovou stránku, obepisuje očima písmeno F. Díky tomuto poznatku se dá odvodit důležitost uživatelského pohledu na informační portál při první návštěvě. Je proto vhodné do těchto míst dát nejdůležitější odkazy a informace v takové míře, aby uživatele zaujaly a navíc mu pomohly k rychlé orientaci.

Aby byl informační server pro uživatele nějak zajímavý, musí být kvalitní nejen po obsahové stránce, ale i po stránce orientační. Aby se uživatel na stránky vrátil, tak je potřeba, aby ho zaujaly hlavně po obsahové a také po designové stránce. U některých nalezených informačních portálů ve Zlínském kraji je právě designová stránka webu nevýrazná a jejich nesprávné zobrazení v nejpoužívanějších prohlížečích spíše mnoho uživatelů od další návštěvy odradí.

### **3.3 Stanovení priorit**

#### **3.3.1 Přehlednost**

Pro každý informační portál je jedním z nejdůležitějších kritérií pro přilákání uživatelů jeho přehlednost. Proto je velmi důležité zvážit volbu designu na úkor přehlednosti. Web by měl už při první návštěvě pro uživatele jasně ukazovat jeho rozložení. Kde kliknout, aby se zobrazily kategorie, kde kliknout, aby se vyhledávalo. Jasná struktura webu dává více možností pro vyhledání informace, která je požadována.

#### **3.3.2 Obsahová kvalita**

Nejdůležitějším kritériem informačního portálu je jeho obsahová kvalita. Jelikož naplňování portálu akcemi není schopen zajistit jeden člověk, je vhodným řešením udělat informační portál otevřený, čímž do něj může přispívat kdokoliv. Při tomto způsobu sice hrozí riziko zahlcení databáze nesmyslnými akcemi, ale vhodnými prostředky lze tento negativní faktor eliminovat na minimum. Pro tento způsob naplňování databáze hovoří naopak mnohá pozitiva. Informační portál je spojovacím členem mezi uživatelem a pořadatelem akce. Pro většinu pořadatelů akcí je to vlastně reklama zdarma a tím, že o dané akci budou informovat na internetu, si zajistí více návštěvníků.

## 4 VYTVOŘENÍ INFORMAČNÍHO SERVERU

### 4.1 Návrh databáze

Celá databáze obsahuje 18 tabulek. Při návrhu bylo přihlédnuto k faktu, aby se neukládaly všechny diskusní příspěvky do jedné společné tabulky, a proto byla tabulka shromažďující diskusní příspěvky u všech akcí rozdělena na 5 tabulek, které shromažďují diskusní příspěvky akcí pro jednotlivé hlavní kategorie. Seznam všech tabulek:

- akce
- uzivatele
- diskuze\_kultura, diskuze\_sport, diskuze\_spolecenske, diskuze\_zajmy, diskuze\_ostatni
- kategorie
- mista\_konani
- statusy\_akci
- hodnosti
- faktor\_natesenosti, hodnoceni\_akce
- oblibene
- akce\_ve\_dnech
- dny
- hodnoty
- potvrzeni

Páteřními tabulkami jsou tabulky *akce* a *uzivatele*. Ostatní tabulky reprezentují vztahy 1:N nebo M:N mezi těmito dvěma tabulkami a jejich funkce bude rozebrána v následujících kapitolách.

#### 4.1.1 Tabulka *akce*

Tabulka *akce* shromažďuje veškeré akce, které kdy uživatelé vložili do systému. Primárním klíčem, který jednoznačně identifikuje každý záznam v tabulce je pole *Akce\_ID*. Část řetězce ID i v ostatních tabulkách znamená, že se jedná o pole reprezentující buď primární nebo cizí klíč tabulky.

Další atributy popisují vlastnosti vložených akcí. Pole *nazev* udává název akce. Pole *Uziv\_ID* je cizí klíč, jenž se odkazuje do tabulky *uzivatele* a reprezentuje tedy registrovaného uživatele, který akci do databáze vložil. Atribut *text* obsahuje popis akce. Ve sloupci *obrazek* je uložena relativní cesta k obrázku symbolizující kategorii. Sloupec *datum\_konani* je typu DATE a je v něm uloženo datum, kdy se akce koná. Naproti tomu ve sloupci *datum\_konec* je uloženo datum, kdy daná akce končí. V případě, že je akce jednodenního charakteru jsou oba datумы stejné. Sloupec *datum\_vlozeni* je neměnný a je v něm uloženo datum, kdy byla akce vložena do databáze. Atribut *Kat\_ID* je cizím klíčem, jenž propojuje tabulku vazbou 1:N s tabulkou *kategorie* a přiřazuje akci do jedné z kategorií. Stejně tak sloupec *Mista\_ID* je cizím klíčem, jenž je vazbou 1:N s tabulkou *mista\_konani* a přiřazuje akci místo konání z míst, jenž jsou uložena v tabulce *mista\_konani*. Ve sloupci *pocet\_ctenaru* se nachází číslo, jenž určuje počet lidí, jenž na danou akci na webu „klikli“. Sloupec *Status\_ID* je cizím klíčem reprezentujícím vazbu 1:N s tabulkou *statusy\_akci* a vyjadřuje stav, ve kterém se zrovna akce nachází (neaktivní, aktivní, archivní, smazaná). Pole *autor\_akce* bývá naplněno pouze v případě, kdy akci do databáze vloží neregistrovaný uživatel. V tomto případě do něj patří jméno uživatele, který akci do databáze vložil. Ve sloupci *ipadress* se nachází IP adresa počítače, z něhož byla akce do databáze vložena.

Tab 1: Tabulka *akce*

Název pole	Datový typ
Akce_ID	INTEGER
nazev	VARCHAR (250)
Uziv_ID	INTEGER
text	TEXT
obrazek	VARCHAR (250)
datum_konani	DATE
datum_vlozeni	DATETIME
datum_konec	DATE
Kat_ID	INTEGER
Mista_ID	INTEGER

pocet_ctenaru	INTEGER
Status_ID	INTEGER
autor_akce	VARCHAR (50)
ipadress	VARCHAR (60)

#### 4.1.2 Tabulka *uzivatel*

Tabulka *uzivatel* spravuje informace o všech registrovaných uživateliích v systému. Primárním klíčem, jenž identifikuje každého uživatele, je pole *Uziv\_ID*. Další položky vyjadřují vlastnosti jednotlivých uživatelů. Pole *nick* charakterizuje přezdívku uživatele s kterou vystupuje na fórech. Atribut *heslo* je MD5 zašifrovaný řetězec, který slouží spolu s nickem uživatele jako přihlašovací údaj do systému. Obsah polí *jmeno*, *prijmeni* a *email* jasně vystihuje jejich název. Ve sloupci *datum\_reg* je uloženo datum, kdy se uživatel registroval do systému. Sloupec *datum\_prihlaseni* se pravidelně aktualizuje při každém přihlášení uživatele do systému, tudíž obsahuje datum jeho posledního přihlášení. Atribut *ipadress* obsahuje IP adresu počítače, z něhož byla provedena registrace. Poslední sloupec *Hodnost\_ID* je cizím klíčem z tabulky *hodnosti*, v níž je klíčem primárním a určuje hodnosti uživatelů, o nichž bude podrobněji pojednáno v dalších kapitolách.

Tab 2: Tabulka *uzivatele*

Název pole	Datový typ
Uziv_ID	INTEGER
nick	VARCHAR (20)
heslo	VARCHAR (30)
jmeno	VARCHAR (30)
prijmeni	VARCHAR (50)
email	VARCHAR (50)
datum_reg	DATETIME
datum_prihlaseni	DATETIME
ipadress	VARCHAR(60)
Hodnost_ID	INTEGER

#### 4.1.3 Tabulky *diskuze*

Pro ukládání příspěvků do diskuzí jsem použil pět tabulek, které se liší pouze svým názvem a názvem primárního klíče. Příspěvky se ukládají do tabulky, která přísluší jedné z hlavních kategorií. Tím je myšleno, že příspěvek na sportovní akci se uloží do tabulky *diskuze\_sport* atd. Jelikož jsou všechny tabulky strukturou prakticky stejné, uvedu zde pouze strukturu tabulky *diskuze\_kultura*.



Tabulka obsahuje primární klíč *D\_Kultura\_ID*. Sloupec *Kat\_ID* je cizím klíčem z tabulky *kategorie* a jednoznačně identifikuje, do jaké konkrétní kategorie příspěvek patří. Sloupec *Akce\_ID* je cizím klíčem tabulky *akce* a přiřazuje příspěvek k požadované akci. Stejně tak sloupec *Uziv\_ID* ukazuje na registrovaného uživatele, jenž daný příspěvek vložil. Atribut *obsah* obsahuje text příspěvku. Atribut *jmeno* obsahuje jméno nebo nick neregistrovaného uživatele. Sloupec *datum\_vlozeni* obsahuje datum a čas vložení zprávy do databáze. Sloupec *ipadress* značí IP adresu počítače, ze které byl příspěvek odeslán.

Tab 3: Tabulka *diskuze\_kultura*

Název pole	Datový typ
D_Kultura_ID	INTEGER
Kat_ID	INTEGER
Akce_ID	INTEGER
Uziv_ID	INTEGER
obsah	TEXT
jmeno	VARCHAR (30)
datum_vlozeni	DATETIME
ipadress	VARCHAR (60)

#### 4.1.4 Tabulka *kategorie*

Tabulka *kategorie* obsahuje seznam všech kategorií. Celkem je uživatelům nabízeno 5 hlavních kategorií (Kultura, Sport, Společenské akce, Zájmy, Ostatní akce), které jsou rozděleny do 24 podkategorií. Uložení kategorií do samostatné tabulky je nejen splněna jedna z podmínek 1. NF, ale navíc je umožněno do budoucna jednoduše rozšířit počet podkategorií.

Každá kategorie má svůj primární klíč *Kat\_ID*. Primární klíč je tvořen 3 číslicemi, z nichž první číslice identifikuje, do jaké hlavní kategorie jednotlivé podkategorie spadají.

1xx – podkategorie spadá do kategorie Kultura

Další dvě číslice se už nikdy neopakují a jsou jednoznačné pro každou podkategorii. Pole *hlavni* udává název hlavní kategorie, do které daná podkategorie patří. Sloupec *podkategorie* obsahuje název podkategorie.

Tab 4: Tabulka *kategorie*

Název pole	Datový typ
Kat_ID	INTEGER
hlavni	VARCHAR (250)
podkategorie	VARCHAR (250)

#### 4.1.5 Tabulka *mista\_konani*

V tabulce *mista\_konani* se nachází seznam všech obcí ve Zlínském kraji. Celkem se jedná o 304 položek jejich primárním klíčem je sloupec *Mista\_ID*. Primární klíč se skládá ze 4 číslic, kde první číslice identifikuje, do kterého okresu (Kroměříž, Uherské Hradiště, Vsetín, Zlín) daná obec patří.

1xxxx – obec z okresu Kroměříž

Další tři číslice jsou pro každou obec v daném okrese jiné. Pole *nazev* poté obsahuje název obce.

Tab 5: Tabulka *mista\_konani*

Název pole	Datový typ
Mista_ID	INTEGER
nazev	VARCHAR (250)

#### 4.1.6 Tabulka *statusy\_akci*

Tabulka *statusy\_akci* obsahuje popis statusů, ve kterých se jednotlivé záznamy v tabulce akce mohou nacházet. Pole *Status\_ID* svou číselnou hodnotou reprezentuje hodnotu statusu akce a zároveň je primárním klíčem tabulky. Sloupec *hodnota* je pak písemný název statusu. Podrobnější popis statusů je v dalších kapitolách.

Tab 6: Tabulka *statusy\_akci*

Název pole	Datový typ
Status_ID	INTEGER
hodnota	VARCHAR (250)

#### 4.1.7 Tabulka *hodnosti*

Tabulka *hodnosti* shromažďuje hodnosti uživatelů. V tabulce *uzivatele* je tedy její primární klíč *Hodnost\_ID* cizím klíčem. Pole *nazev* je pak textový popis jednotlivých hodností.

V tabulce se nachází 5 druhů hodností a jejich funkce je spojena s možností vkládání příspěvků (akcí) do databáze.

Tab 7: Tabulka *hodnosti*

Název pole	Datový typ
Hodnost_ID	INTEGER
nazev	VARCHAR (30)

#### 4.1.8 Tabulky *faktor\_natesenosti*, *hodnoceni\_akce*

Tabulky *faktor\_natesenosti* a *hodnoceni\_akce* jsou naprosto shodné, avšak každá byla vytvořena pro jiný účel. Už z názvu tabulek lze usoudit, že tabulka *faktor\_natesenosti* bude shromažďovat hodnocení uživatelů ještě před konáním akce, zatímco tabulka *hodnoceni\_akce* shromažďuje hodnocení uživatelů až po zahájení akce.

Tabulky jsou M:N spojením tabulek *akce* a *uzivatele*. Tím se myslí, že kterýkoliv uživatel může hodnotit jakoukoliv akci a jakákoliv akce může být hodnocena kterýmkoliv uživatelem. Tudíž pole *Akce\_ID* a *Uziv\_ID* jsou v nich cizí klíče. Pole *Hodnota\_ID* je cizí klíč z tabulky *hodnoty* a vyjadřuje procentuální hodnocení (10 – 100%). Sloupec *datum\_hodnoceni* zaznamenává datum a čas hodnocení akce.

Tab 8: Tabulka pro hodnocení

Název pole	Datový typ
Akce_ID	INTEGER
Uziv_ID	INTEGER
Hodnota_ID	INTEGER
datum_hodnoceni	DATETIME

#### 4.1.9 Tabulka *oblíbene*

Tabulka *oblíbene* je M:N vazbou mezi tabulkami *akce* a *uzivatele*. Její funkcí je zaznamenávat akce, jenž si uživatelé přidali do svých oblíbených a navíc ukládat informaci, zda-li si uživatelé přejí zaslat upozornění na e-mail den před konáním akce.

I přesto, že lze za primární klíče v této tabulce považovat sloupce *Akce\_ID* a *Uziv\_ID*, přidal jsem do této tabulky sloupec *Oblíbene\_ID*, jenž považuji za primární klíč. Pole *Potvrzeni\_ID* je cizím klíčem z tabulky *potvrzeni* a jeho hodnota vyjadřuje, jestli chce uživatel na danou akci upozornit e-mailem. V poli *datum\_pridani* je uložen datum a čas, kdy byla akce do oblíbených přidána.

Tab 9: Tabulka *oblíbene*

Název pole	Datový typ
Oblíbene_ID	INTEGER
Akce_ID	INTEGER
Uziv_ID	INTEGER
Potvrzeni_ID	SMALLINT
datum_pridani	DATETIME

#### 4.1.10 Ostatní tabulky

Další tabulky, které se v databázi nacházejí splňují především funkci seskupení stále se opakujících hodnot, aby databáze splňovala jednu z podmínek pro 1.NF (tabulky *dny*, *hodnoty*, *potvrzeni*).

Za zmínku ještě stojí tabulka *akce\_ve\_dnech*, která ukládá dny, ve kterých se akce koná, pokud se jedná o dlouhodobou akci.

## 4.2 Registrace, přihlášení do systému

Aby mohli uživatelé plně využívat funkce webu, je nutné se nejprve do systému registrovat a následně se přihlásit. Registrovaní uživatelé mají oproti obyčejným uživatelům o mnoho více výhod. Jejich vložené akce se zobrazí na webu ihned po jejich vložení, mohou si přidávat akce do oblíbených a tudíž je na ně může systém upozorňovat nebo mají možnost libovolně upravovat pole popisu akce v editoru.

Především z hlediska větší přehlednosti v kódu bylo při programování využito objektových prvků. Proto pro samotný registrační proces byla vytvořena třída *Uzivatel*. V ní jsou veškeré metody, které předcházejí samotnému zaregistrování uživatele do databáze. Hlavně se jedná o ošetřující metody z hlediska volby uživatelova nicku nebo jeho e-mailu. Také je ošetřeno zadávání html entint, ošetření uvozovek, aby se zabránilo pokusům o narušení databáze.

Po úspěšném registračním procesu je uživateli nabídnuta možnost přihlášení do systému. V okamžiku, kdy se uživatel úspěšně přihlásí jsou do superglobálního pole `$_SESSION` uloženy proměnné `„prihlaseni“` a `„uzivatel“`, jimiž se každý přihlášený uživatel identifikuje. Session je při nečinnosti aktivní 1300 sekund. Po uplynutí této doby dojde k automatickému ukončení session a tedy i k odhlášení uživatele.

`$_SESSION[„prihlaseni“]` slouží k identifikaci přihlášených uživatelů. Jejím hlavním úkolem je rozlišovat přihlášené a nepřihlášené uživatele a podle toho jim zpřístupňovat některé části webu. V `$_SESSION[„uzivatel“]` se nachází nick přihlášeného uživatele. Jelikož v databázi se nemůžou nacházet uživatelé s dvěma stejnými přezdívkami, tak je to dostatečný symbol přihlášeného uživatele. Jeho využití je hlavně v identifikaci hodnoti uživatele, kterou obstarává příslušná funkce. Tímto způsobem se uživatelům poskytují větší či menší pravomoci v oblasti vkládání a upravování akcí.

### 4.3 Hodnosti uživatelů

Na každém webu, do jehož databáze může přispívat, kdokoliv, kdo má přístup k internetu, je třeba řešit otázku bezpečnosti při vkládání údajů do databáze. Zvláště při vkládání informací, které se ihned zobrazují široké veřejnosti.

Prvním problémem, který jsem řešil, bylo rozhodnutí, zda-li umožnit vkládání příspěvků neregistrovaným uživatelům. Nakonec však po prozkoumání podobných portálů v České Republice bylo rozhodnuto, že i neregistrovaní uživatelé budou mít možnost vložit do databáze nové informace. Avšak je pro ně připraveno omezení v podobě maximálního vložení akcí za den a to v počtu 2. Navíc je potřeba, aby takovou akci nejprve schválil administrátor.

Pro registrované uživatele byl zvolen systém hodností, který je jednoznačně rozděluje na 5 druhů. Seznam hodností spravuje tabulka *hodnosti*. Když se uživatel zaregistruje, je mu automaticky přidělena základní hodnost 1 (viz. tabulka seznamu hodností). Tento uživatel má právo vložit za den maximálně 10 akcí a z toho první dvě se na webu zobrazí ihned po vložení do databáze. Se zvyšující se hodností se zvyšuje počet akcí, které se zobrazí na webu bez nutnosti potvrzení administrátorem. Byl zvolen větší počet hodností z důvodu, případného postupného ověření uživatele, než-li by mu byla dána absolutní volnost. Navíc do budoucna lze systém obohatit tak, že by se akce, jejichž uživatel má vyšší hodnost, zobrazovaly v databázi na předních místech. Z tohoto důvodu bylo také zvoleno číselné značení hodností.

Hodnost 4 je nejvyšší hodností po administrátorovi. Uživatel této hodnosti má kromě práva vložení největšího počtu akcí (40 denně s okamžitým zobrazením) i právo upravovat ostatní akce. Tuto hodnost lze proto udělit jen ověřené osobě.

Nejvyšší hodnost (číselné označení 5) má administrátor. Administrátor má neomezená práva, kromě úplného smazání akce. Tím se myslí, že administrátor sice jako jediný může akci smazat, ale nesmaže ji fyzicky z databáze, ale pouze změní její status na status *deleted*. Tím bude akce vyřazena ze zobrazení na webu a nikdo, kromě uživatele, jenž ji uložil a administrátora, ji neuvidí. Toto opatření vzniklo z důvodu rychlého “smazání” akce z webu, pokud by jakýmkoliv způsobem porušovala pravidla. Jelikož však akce fyzicky z databáze nezmizí, je možné ji zpětně obnovit, anebo v případě závažného prohřešku zakázat přístup na web uživateli s IP adresou, ze které byla akce vložena.

Tab 10: Seznam hodnotí uživatelů

číslo	písemný název	okamžitě zobrazených akcí / den	max akcí / den	ostatní možnosti
1	beginner	2	10	upravování vlastních akcí
2	advanced	5	15	upravování vlastních akcí
3	professional	10	20	upravování vlastních akcí
4	certified	40	40	upravování všech akcí
5	admin	neomezeně	neomezeně	upravování a mazání všech akcí

#### 4.4 Statusy akcí

Stejně jako uživatelé mají své hodnoty, tak každá akce má svůj status. Status akce vyjadřuje její stav. Statusy akcí spravuje tabulka *statusy*, jenž je vztahem 1:N propojena s tabulkou *akce*. Statusy jsou 4 druhů:

- a) Status\_ID = 1, hodnota = waiting, akce čekající na potvrzení administrátorem
- b) Status\_ID = 2, hodnota = active, aktivní akce
- c) Status\_ID = 3, hodnota = archiv, archivní akce
- d) Status\_ID = 4, hodnota = deleted, smazaná akce

Akce, jenž mají status 2 a 3 jsou volně k zobrazení na webu. Zatímco akce se statusy 1 a 4 mají omezené zobrazení. Tím je myšleno, že se zobrazují pouze uživateli, který akci vložil do databáze a administrátorovi. Funkce statusů je úzce spjata s hodnotami uživatelů. Tím je myšleno, že to, jaký status nová akce dostane, záleží na hodnotě uživatele. Nová akce od neregistrovaného uživatele dostává automaticky status 1 a její zobrazení na webu musí potvrdit administrátor tím, že změní její status na status 2. U registrovaných uživatelů se statusy mění podle toho, zda-li má daný uživatel právo vložit akci přímo na web bez nutnosti potvrzení administrátora. Tedy akce, jenž se zobrazí přímo na web dostanou status 2, zatímco akce, jenž musí čekat na potvrzení získají status 1 (počet akcí, které získají jednotlivý status viz tabulka seznamu hodnotí uživatelů).

Status 3 (archivní status), získává akce v okamžiku, jakmile ode dne, kdy skončila uplynulo více jak 14 dnů. Změnu tohoto statusu hlídá skript, který se spouští na serveru 1x denně.

## 5 VLASTNOSTI WEBOVÉHO PORTÁLU

### 5.1 Vložení, editace nové akce

Jednou z nejdůležitějších vlastností informačního serveru je vkládání nových aktualit (akcí). Pro tento účel se na každé webové stránce zobrazuje odkaz „Vložit Novou Akcí“. Po odkliknutí se uživateli zobrazí kalendář z něhož si musí zvolit datum konání akce. Akci lze vložit pouze do aktuálního dne nebo do dne, jenž ještě nenastal. Jakmile dojde k výběru datumu konání akce zobrazí se několik formulářových polí, které slouží k záznamu popisu akce.

Je nutno vyplnit název akce. Zvolit místo konání a to nejprve podle jednoho z bývalých okresních měst a poté vybrat jednotlivou obec. Dále je nutno zvolit kategorii, do níž akce spadá. Popis akce není nutno vyplňovat, ale pro větší úspěšnost při vyhledávání je dobré popis akce vyplnit. Neregistrovaní uživatelé musí ještě vyplnit pole *jméno*, které slouží k identifikaci autora akce.

Textové pole, kde se vepisuje popis akce se liší v zobrazení pro neregistrované a registrované uživatele. Zatímco neregistrovaným uživatelům se zobrazuje obyčejné textové pole, tak registrovaní uživatelé mají možnost využít úpravy výsledného popisu akce pomocí integrovaného TinyMCE editoru, který je upraven pro základní formátování textu. Je tedy možné vkládání přímých odkazů, pozicování textu, nastavení stylu a vkládání externích obrázků. Editor i textové pole mají šířku nastavenou tak, jakou šířku lze využít v prostoru na webu.

Po odeslání formuláře dojde nejprve ke kontrole vyplnění údajů přímo v prohlížeči pomocí javascriptu. Pokud některé povinné údaje nejsou vyplněny, zobrazí se chybová hláška, která uživatele na chybu upozorní. Když je vše v pořádku odešlou se informace na server, kde je provedena druhá kontrola údajů pomocí PHP skriptů. Stejně jako registrování nových uživatelů, tak i vkládání a editaci akcí obsluhuje samostatná *třída Akce*. V této třídě se nacházejí metody, které ošetřují všechny vstupní položky při vkládání a editaci nových akcí na úrovni serveru a připravují je pro zápis do databáze. Pokud jsou všechny údaje v pořádku, tak je akce uložena do databáze a uživateli se zobrazí hláška, která úspěšné vložení potvrdí.

Každý registrovaný uživatel má právo svou akci následně i upravit. V upravovacím formuláři se také nacházejí pokročilé položky, umožňující nastavit délku trvání akce, pokud je akce dlouhodobého charakteru. Upravovací tlačítko se zobrazuje pouze u výpisu akce. Upravovací formulář vypadá stejně jako formulář pro vkládání nových akcí, ale jsou zde přidány právě nové položky pro nastavení délky konání akce. Uživatel musí nejprve zaškrtnout dny v týdnu, v nichž se akce koná a poté zadat datum konce akce, který má jinak primární hodnotu datumu, kdy akce začíná. Ještě je nutno zaškrtnout tlačítko, kterým se tyto úpravy potvrzují a po uložení akce do databáze se vše zobrazí podle zadaných podmínek.

Vložte název akce:

Zvolte místo konání akce:

Přiřadte akci kategorii:

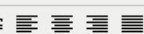
Datum konání akce (YYYY-MM-DD):




**Pokročilé nastavení**  
opakování akce (pro stálé akce typu výstavy apod.):  
Zaškrtněte dny, kdy se akce bude opakovat:  
 Pondělí  Úterý  Středa  Čtvrtek  Pátek  Sobota  Neděle

Zadejte datum ukončení akce ve formátu YYYY-MM-DD (nutno zadat v každém případě, kdy se akce opakuje):

Zaškrtněte v případě, že chcete uložit pokročilé nastavení

Vložte text k akci:  

**B I U** ABC 

**Odeslat**

Obr 1: Formulář pro vložení akce, pro přihlášené uživatele



## 5.2 Struktura webu

V analýze jsem se zmínil, že důležitou vlastností informačního portálu by měla být přehlednost při vyhledávání konkrétních akcí. Proto byl při rozmíst'ování důležitých odkazů brán ohled na rychlou orientaci ve struktuře webu.

Při vyhledávání konkrétních akcí může uživatel volit rozdělení akcí na jednotlivé okresy. Dále má na výběr vypsání akcí pro jednotlivé kategorie a nakonec může s pomocí kalendáře vyhledat akce podle jednotlivého měsíce nebo dne. Postupným vyhledáváním na principu stromového větvení lze dosáhnout požadovaných výsledků:

Místo konání - > Kategorie - > Datum konání

**Zlínsko Žije** 30. Duben, 2007 [Chcete se Přihlásit](#) nebo [Registrovat?](#)  
Naleznete zde 4 připravovaných akcí [Uplynulé akce](#) [Archiv akcí](#) [Statistiky](#)  
Dnes vloženo 0 akcí

Zvolte lokalitu: **Kroměříž** Uherské Hradiště Vsetín Zlín Celý kraj

**VLOŽIT NOVOU AKCI** Vyhledat akci:  **Hledat** [Podrobnější vyhledávání](#)

**Kalendář všech akcí Zlínského kraje**

<< Duben 2007 >>

Po	Út	St	Čt	Pá	So	Ne
26	27	28	29	30	31	01
02	03	04	05	06	07	08
09	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	01	02	03	04	05	06

**Výpis akcí Zlínského kraje:**

**Hrad Buchlov (1. 4. 2007- 30. 10. 2007) DNES**  
DUBEN - APRIL - APRIL ŘÍJEN - OCTOBER - OKTOBER prohlídkyvisit ? Bes ...  
[Zájem](#) -> [Výlety](#)

**Ječmínkův běh (1. 5. 2007)**  
Atleti poběží Ječmínkův běh CHROPYNĚ Dvaadvacátý ročník přespolního ...  
[Sport](#) -> [Ostatní sporty](#)

**Trhy drobného zvířectva a okra ... (6. 5. 2007)**  
ZO ČSCH KROMĚŘÍŽ, HORNÍ ZAHRADY Každou první neděli v měsíci od 7 h - ...  
[Společenské akce](#) -> [Trhy](#)

<< 1 >>

**Výběr podle kategorií**

- Kultura**  
Koncerty, Divadla, Kina, Výstavy, Festivaly, Hudba,
- Společenské Akce**  
Plesy, Diskotéky, Akce pro děti, Dny otevřených dveří, Slavnosti, Trhy,
- Sport**  
Fotbal, Hokej, Tenis, Ostatní sporty,
- Zájem**  
Výlety, Turistika, Kurzy, kroužky,
- Ostatní akce**  
Přednášky, Prezentační akce, Humanitární akce, Nákupní centra, Ostatní,

**Nejnavštěvovanější akce Zlínského kraje:**

**Hrad Buchlov**  
DUBEN - APRIL - APRIL ŘÍJEN - OCTOBER - OKTOBER prohlídkyvisit ? Bes ...  
[Zájem](#) -> [Výlety](#) Počet zobrazení: 3x

**Ostatní Sporty**  
**Ječmínkův běh**  
Atleti poběží Ječmínkův běh CHROPYNĚ Dvaadvacátý ročník přespolního ...  
[Sport](#) -> [Ostatní sporty](#) Počet zobrazení: 3x

**Trhy drobného zvířectva a okra ...**  
ZO ČSCH KROMĚŘÍŽ, HORNÍ ZAHRADY Každou první neděli v měsíci od 7 h - ...  
[Společenské akce](#) -> [Trhy](#) Počet zobrazení: 2x

**Nejnovější akce ve Zlínském kraji:**

**Ostatní Sporty**  
**Ječmínkův běh**  
Atleti poběží Ječmínkův běh CHROPYNĚ Dvaadvacátý ročník přespolního ...  
[Sport](#) -> [Ostatní sporty](#) Vloženo: 27. 4. 2007

**Hrad Buchlov**  
DUBEN - APRIL - APRIL ŘÍJEN - OCTOBER - OKTOBER prohlídkyvisit ? Bes ...  
[Zájem](#) -> [Výlety](#) Vloženo: 27. 4. 2007

Obr 2: Struktura webu, rozdělení na okresy, kategorie, kalendář

S výhodou lze použít i standardního vyhledávání pomocí klíčových slov, ke kterému slouží formulářové pole, které se zobrazuje na každé stránce. Pokud je potřeba nastavit pokročilejší možnosti vyhledávání (volba konkrétní lokality – i jednotlivých obcí, volba kategorie, volba co se má prohledat – aktuální akce, uplynulé akce, archiv) je k tomu určena volba „pokročilé vyhledávání“.

### 5.3 Výběr akcí z databáze

Nejčastější činností informačního serveru bude výběr akcí z databáze. Veškeré skripty, které slouží pro tento účel jsou součástí třídy *AkceVyber*. Třída obsahuje 41 metod, které zajišťují výběr z databáze a je klíčová pro funkčnost celého webu.

Vzhledem k možnostem hostingu, na které běží databázový systém MySQL 4.0 muselo být upuštěno od používání vnořených dotazů při výběru dat z databáze. Při potřebě použít složitější výběrový skript, je použito řešení pomocí vytvoření dočasné tabulky (TEMPORARY TABLE), ve které je vnořený dotaz obsažen:

```
CREATE TEMPORARY TABLE docasna
SELECT akce.Akce_ID FROM akce WHERE akce.Status_ID='2';
```

```
SELECT * FROM oblíbene
INNER JOIN akce ON oblíbene.Akce_ID=akce.Akce_ID
INNER JOIN docasna ON docasna.Akce_ID=akce.Akce_ID
WHERE oblíbene.Akce_ID=docasna.Akce_ID
```

Na výše uvedeném příkladě je uvedeno nahrazení vnořeného dotazu, vytvořením dočasné tabulky. Do tabulky *docasna* jsou nejprve uloženy všechny akce, které mají status 2. Poté je proveden výběr dat z tabulky oblíbených akcí a jsou vybrány právě ty akce, které jsou v tabulce oblíbených a zároveň v tabulce *docasna*.

### 5.4 Uplynulé akce, archiv akcí

Aby se na webu nenacházely akce, které už nejsou aktuální byl vytvořen postup, v němž jsou akce, které skončily před více jak 14 dny přesunuty do archivu akcí (je jim nastaven status 3). V hlavní vyhledávací části webu se však nacházejí pouze akce, které probíhají, ty které proběhly a od jejich konce ještě neuplynulo 14 dnů se nacházejí v odkazu „uplynulé akce“. V archivu akcí ani v uplynulých akcí nelze akce vyhledávat podle kalendáře, ale lze

je třídít podle kategorií obdobným způsobem, kterým se třídí akce aktuální. V uplynulých akcích i v archivu akcí lze i samostatně vyhledávat podle klíčového slova.

## 5.5 Statistiky

Každému, kdo web navštíví, jsou k dispozici statistiky akcí. Pod stejnojmenným odkazem si uživatel bude moci vypsát různé druhy statistik (viz Tab. 11) pro jednotlivé okresy nebo pro jednotlivé kategorie. Statistiky obsluhuje skript, který je metodou ve třídě *AkceVyber*.

Tab 11: Seznam statistik

Název statistiky
nejnavštěvovanější
nejméně navštěvovanější
nejoblíbenější
největší natěšení
nejmenší natěšení
nejlépe hodnocené
nejhůře hodnocené
nejdiskutovanější
včera vložené
dnes vložené

**Výpis nejnavštěvovanějších akcí všech kategorií Zlínského kraje**

**Výstavy** **Núbie v dobách faraonů**  
 Výstava ze sbírek Národního muzea - Náprstkova muzea a Městského muzea ...  
[Kultura](#) -> [Výstavy](#) Počet zobrazení: 11x

**Záimv** **Buchlovice - zámek**  
 Buchlovice - zámek, Buchlovice, 68708, Vstupné: 50tel: 572 434 240, em ...  
[Záimv](#) -> [Výlety](#) Počet zobrazení: 8x

**Ostatní - Kultura** **Největší logo města - pokus o ...**  
 Pomozte nám od 1. března do 17. dubna sbírat víčka od PET lahví a zůč ...  
[Kultura](#) -> [Ostatní - Kultura](#) Počet zobrazení: 7x

**Ostatní akce** **Pálení čarodějnic a stavění má ...**  
 30.4.2007, 17.00 (Sportovní areál, Babice)  
[Ostatní akce](#) -> [Ostatní](#) Počet zobrazení: 6x

**Zvolte si, jakou statistiku chcete vypsát:**

Zvolte typ statistiky: (seřadí od)

Zvolte lokalitu:

Zvolte kategorii:

Obr 3: Formulář pro výběr statistik

## 5.6 Bezpečnostní opatření

Základními bezpečnostními opatřeními při práci s formulářovými poli je dvojitá kontrola zadaných údajů. Nejprve proběhne kontrola na straně uživatelského prohlížeče a následně kontrola na serveru. Před samotným zpracováním dat nebo uložením dat do databáze jsou texty ošetřeny proti zadávání html znaků nebo pro správné zobrazení uvozovek. Každá aktivita uživatele na webu (vkládání příspěvků, vkládání akcí) je zaznamenávána. Tím se myslí, že je zaznamenána IP adresa a datum aktivity uživatele. Tyto údaje mohou sloužit pro budoucí identifikaci počítače u něhož vzniklo podezření pro porušení pravidel.

## 5.7 Možnosti nastavení

Tato sekce obsahuje osobní údaje uživatele. Je přístupná pouze přihlášeným uživatelům a je možno v ní změnit jméno, příjmení a e-mailovou adresu.. Nick uživatele je volen při registraci a jednoznačně identifikuje uživatele a proto je neměnný.

Dále má uživatel možnost změnit si heslo, kde musí nejprve zadat své původní a poté dvakrát nové heslo.

Uživatel s hodností 5 (administrátor) má navíc k dispozici administrační rozhraní. V tomto odkazu je mu umožněno vyhledávání uživatelů podle jejich primárních klíčů nebo podle jejich nicků. Dále má také možnost vyhledávat akce podle jejich ID. Následná správa mu umožní změnit uživatelskou hodnost a nebo upravovat či mazat akce.

## ZÁVĚR

V rámci bakalářské práce byl vytvořen informační portál, který si vytyčil za cíl přehlednost, jednoduchost, uživatelskou přístupnost a s tím spojenou obsahovou kvalitu. Ta se odvíjí především od příspěvků uživatelů, protože otevřenost portálu má za smysl jeho častou aktualizaci všemi uživateli. Bylo a je nutné portál nejprve obohatit akcemi, kterými by si získal pravidelné čtenáře. Propagace probíhala především formou letáků v prostředí univerzity. Web byl zaregistrován i do nejnavštěvovanějších českých internetových katalogů a v době psaní této práce se po zadání klíčových slov „akce ve Zlínském kraji“ nebo „akce Zlínsko“ vyskytoval na předních místech ve vyhledávači Google.

Díky logickému třídění na kategorie, okresy a kalendářní měsíce mají uživatelé bohaté možnosti k nalezení požadované akce. Navíc je jim nabídnuta pokročilá funkce vyhledávání podle klíčových slov. Dělení akcí podle datumu konání na aktivní, uplynulé a archivní jednoznačně uživatelům zjednodušuje orientaci při vyhledávání. Vzhledem k podrobným informacím, které se ke každé akci ukládají do databáze lze do budoucna web obohatit podrobnějšími statistikami o vkládání nebo návštěvnosti akcí za jednotlivé měsíce. Protože lze očekávat, že se bude archiv akcí plnit, je možno do budoucna doprogramovat modul, který by archiv rozdělil podle akcí na jednotlivé měsíce nebo roky.

Celý portál je tedy nastartován k plnění své základní funkce: „informovat široké okolí“. Nyní záleží na tom, jestli najde uplatnění u uživatelů internetu.

## ZÁVĚR V ANGLIČTINĚ

This informative portal was created in terms of bachelor's work. The aim was lucidity, simplicity, users friendly approach and mainly quality content. High-quality content depends on all user's contributions. By opening this portal for all users, we bring them a chance to fill it only by them selves. It was, and it still is, necessary to fill the portal by new actions to obtain regular visitors. The promotion of this portal proceeded by leaflet posted on bulletin board in university. The web was registered to the most frequented czech internet catalogues. It is quite easy to find it in forward position in Google by ordering keywords „akce ve Zlínském kraji“ or „akce Zlínsko“, by the time of writing this text.

Thanks to separated logical sorting of categories, regions and months, users are given all they need to find what they seek. By using the list of keywords, users can also use the advanced searching functions. Actions are sorted out by date of start from the oldest to the most actual, passed and archived actions. This sorting is great simple orientation for users during their search. In the future, more detailed statistics of editing or visiting every single action can be added - every action saves detailed information to the database about itself. A program unit to archive a part of any web, that would sort archive actions in months or years, can be pre-set.

Whole portal is ready to do his basic functions: „to inform people“.

Now, in only depends on its' popularity among internet users.

## SEZNAM POUŽITÉ LITERATURY

Monografie:

- [1] ULLMAN, LARRY *PHP pokročilé programování pro World Wide Web*. SoftPress, 2002. 512 s. ISBN: 80-86497-36-4.
- [2] WILLIAMS, Hugh E. , LANE, David. *PHP a MySQL*. Computer Press, 2002. 552 s. ISBN: 807226704.

Internetové zdroje:

- [3] Jak psát web, návod na html stránky [online]. [cit. 2007-04-30] Dostupný z WWW: <<http://www.jakpsatweb.cz/>>
- [4] Root.cz, seriál PHP v objetí objektů [online]. [cit. 2007-04-30] Dostupný z WWW: <<http://www.root.cz/serialy/php-v-objeti-objektu/>>
- [5] Manualy.net, tvorba webu, tutoriály, PHP, MySQL [online]. [cit. 2007-04-30] Dostupný z WWW: <<http://www.manualy.net/>>
- [6] LinuxSoft.cz, seriál o MySQL [online]. [cit. 2007-04-30] Dostupný z WWW: <[http://www.linuxsoft.cz/article\\_list.php?id\\_kategory=232](http://www.linuxsoft.cz/article_list.php?id_kategory=232)>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

html	HyperText Markup Language, jazyk využívající html entit pro tvorbu statických WWW stránek.
MD5	Rozšířená hašovací funkce (Message-Digest algorithm 5).
MySQL	Databázový server, využívající jazyk SQL.
NF	Normální forma, soubor doporučení pro návrh datové struktury databáze.
PHP	Skriptovací jazyk, používaný k tvorbě WWW stránek
session	Připojení mezi klientem a serverem. V PHP pro delší udržování informací v proměnných.
SQL	Standardizovaný dotazovací databázový jazyk
TEMPORARY TABLE	Dočasná tabulka, kterou lze vytvořit za pomoci SQL příkazu, může sloužit k uložení dočasných záznamů.
TinyMCE	Open source WYSIWYG editor



**SEZNAM OBRÁZKŮ**

Obr 1: Formulář pro vložení akce, pro přihlášené uživatele.....	32
Obr 2: Struktura webu, rozdělení na okresy, kategorie, kalendář.....	33
Obr 3: Formulář pro výběr statistik.....	35

**SEZNAM TABULEK**

Tab 1: Tabulka <i>akce</i> .....	23
Tab 2: Tabulka <i>uzivatele</i> .....	24
Tab 3: Tabulka <i>diskuze_kultura</i> .....	25
Tab 4: Tabulka <i>kategorie</i> .....	25
Tab 5: Tabulka <i>mista_konani</i> .....	26
Tab 6: Tabulka <i>statusy_akci</i> .....	26
Tab 7: Tabulka <i>hodnosti</i> .....	26
Tab 8: Tabulka pro hodnocení .....	27
Tab 9: Tabulka <i>oblibene</i> .....	27
Tab 10: Seznam hodnotí uživatelů .....	30
Tab 11: Seznam statistik .....	35

## SEZNAM PŘÍLOH

PI: Disk CD s bakalářskou prací.