

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM OBECNÍHO ÚŘADU

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

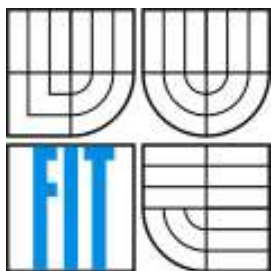
AUTOR PRÁCE  
AUTHOR

JAROSLAV SURÝ

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# INFORMAČNÍ SYSTÉM OBECNÍHO ÚŘADU

INFORMATION SYSTEM OF MUNICIPAL OFFICE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAROSLAV SURÝ

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. MARTIN STRAKA

BRNO 2008

## **Abstrakt**

Bakalářská práce se zabývá problematikou návrhu a implementací informačního systému obecního úřadu. Pro práci byly použity tyto programovací jazyky: PHP, JavaScript, HTML, CSS a databáze MySQL.

## **Klíčová slova**

PHP, JavaScript, HTML, CSS, MySQL, Informační systém

## **Abstract**

The bachelor's thesis is concerned with the dilemma of suggestion and implementation of information system for municipal office. The programming languages – PHP, JavaScript, HTML, CSS and database MySQL were used for this bachelor's thesis.

## **Keywords**

PHP, JavaScript, HTML, CSS, MySQL, Information system

## **Citace**

Surý Jaroslav: Informační systém obecního úřadu. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Informační systém obecního úřadu

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Martina Straky. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jméno Příjmení  
Datum

## Poděkování

Děkuji vedoucímu bakalářské práce Ing. Martinovi Strakovi za vstřícnost, konstruktivní kritiku a odborné vedení při zpracování této práce.

© Jaroslav Surý, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

Úvod .....	4
1 Informační systémy .....	5
1.1 Hranice a okolí .....	5
1.2 Rozdělení systémů .....	5
1.3 Řízení a zpětná vazba .....	5
2 Databáze .....	6
2.1 Historie .....	6
2.2 Databázové modely .....	7
2.3 Relační databáze .....	7
2.3.1 Terminologie .....	7
2.3.2 Klíč .....	7
2.3.3 Integrita databáze .....	7
2.3.4 Druhy integritních omezení .....	8
2.3.5 Vztahy mezi tabulkami .....	8
2.3.6 Normální formy .....	8
3 Použité technologie .....	10
3.1 Popis technologií .....	10
3.2.1 HTML .....	10
3.2.1.1 HTML 2.0 .....	10
3.2.1.2 HTML 3.2 .....	11
3.2.1.3 HTML 4.0 .....	11
3.2.1.4 HTML 4.01 .....	11
3.2.1.5 HTML 5 .....	11
3.2.2 PHP .....	11
3.2.2.1 PHP/FI .....	12
3.2.2.2 PHP 3 .....	12
3.2.2.3 PHP 4 .....	13
3.2.2.4 PHP 5 .....	13
3.2.3 CSS .....	13
3.2.3.1 CSS1 .....	14
3.2.3.2 CSS2 .....	14
3.2.4 JavaScript .....	14
3.2.5 MySQL .....	15
4 Specifikace požadavků .....	16

4.1	Popis vlastního zadání.....	16
4.2	Neformální specifikace.....	16
4.3	Technické požadavky.....	16
5	Analýza a návrh řešení.....	17
5.1	Návrh databáze.....	17
5.1.1	Popis tabulek.....	17
5.1.1.1	Tabulka firmy.....	17
5.1.1.2	Tabulka obec.....	17
5.1.1.3	Tabulka ubytování.....	17
5.1.1.4	Tabulka ubytování_img.....	17
5.1.1.5	Tabulka ubytování_ikona.....	18
5.1.1.6	Tabulka lec_ubytování.....	18
5.1.1.7	Tabulka lec_ubytování_img.....	18
5.1.1.8	Tabulka lec_ubytování_ikona.....	18
5.1.1.9	Tabulka sport_info.....	18
5.1.1.10	Tabulka sport_img.....	18
5.1.1.11	Tabulka sport_ikona.....	19
5.1.1.12	Tabulka guestbook.....	19
5.1.1.13	Tabulka fotogalerie.....	19
5.1.1.14	Tabulka users.....	19
5.1.1.15	Tabulka články.....	19
5.1.1.16	Tabulka úřední deska.....	19
6	Informační systém obecního úřadu.....	20
6.1	Vnitřní systém.....	20
6.1.1	Přihlášení do systému.....	20
6.1.2	Odhlášení ze systému.....	20
6.1.3	Správa firem.....	20
6.1.4	Správa zaměstnanců obecního úřadu.....	20
6.1.5	Správa úřední desky.....	21
6.1.6	Správa diskuzního fóra.....	21
6.1.7	Správa kalendáře akcí.....	21
6.1.8	Správa sportovních aktualit.....	21
6.1.8.1	Přidávání sportovních aktualit.....	21
6.1.8.2	Vkládání obrázku ke sportovní aktualitě.....	22
6.1.9	Správa ubytování v obci.....	22
6.1.9.1	Přidávání nového ubytování.....	22
6.1.9.2	Vkládání obrázku k ubytování.....	22
6.1.10	Správa léčebných pobytů v obci.....	23

6.1.11	Správa článků o obci.....	23
6.1.12	Správa fotogalerie.....	23
6.1.12.1	Vkládání obrázku do fotogalerie.....	23
6.2	Prezentační systém.....	24
6.2.1	Sekce Kde Nás najdete.....	24
6.2.2	Sekce Městské symboly.....	24
6.2.3	Sekce Osobnosti.....	24
6.2.4	Sekce Památky.....	24
6.2.5	Sekce Ubytování.....	24
6.2.6	Sekce Kam na výlet.....	24
6.2.7	Sekce Firmy a služby.....	24
6.2.8	Sekce Fotogalerie.....	25
6.2.9	Sekce Zpravodaj.....	25
6.2.10	Sekce Diskuzní fórum.....	25
6.2.11	Sekce Léčivé prameny.....	25
6.2.12	Sekce Léčebné pobyty.....	25
6.2.13	Sekce Úřední deska.....	25
6.2.14	Sekce E-podatelna.....	26
6.2.15	Sekce Kontakty.....	26
6.2.16	Sekce Homepage.....	26
6.2.17	Sekce Kultura.....	26
6.2.18	Sekce Sport.....	26
6.2.19	Sekce Turistika.....	26
6.2.20	Sekce Kalendář akcí.....	26
6.2.21	Sekce Mapa webu.....	26
7	Závěr.....	27
	Literatura .....	28

# Úvod

**Informační systémy** (IS) jsou systémy pro sběr, udržování, zpracování a poskytování informací a dat. V celé historii měly informační systémy vždy všechny své nezbytné části:

- data modelující stav a množinu transformačních funkcí nad nimi a
- prostředky pro komunikaci s uživatelem.

Historický vývoj neodráží jen přidávání nových systémů, ale o jejich

- klasifikaci,
- separaci a sdružování do vhodných tříd a celků z důvodu snadné definice realizace a údržby a zejména
- způsoby formální definice.

V 50. letech principy modulární výstavby programů zdaleka nebyly propracovány.

K dispozici byly programovací jazyky s primitivním rozhraním k sekundárním persistentním pamětím. Abstrakce typu soubor nebo databáze neexistovala. Síť či podobná komunikační prostředí pro komunikaci neexistovaly. Počítače nebyly ovládány přes klávesnice s monitorem, ale nepřímo zadávaným dávkovým zpracováním. Data nebyla užívána odděleně od procesů a byla jejich součástí. Sdílené použití dat nebylo možné.

V 60. letech nastalo oddělení dat od procesorů. Toto oddělování probíhalo postupně. Nejprve bylo umožněno procesorům pracovat nad více množinami dat. Data byla oddělena od programů, jejich popis však zůstal fyzicky v programech. Výsledek byl takový, že definice programu umožnila pracovat s několika databázemi a jedním systémem spravovat více sad údajů, např. vézt podobné informační systémy pro různé firmy nebo banky [1].

Slovo, které vymezuje informační systém je slovo *systém*. Systém je specifický typ znalostního (kognitivního) modelu pro který platí: Je to soubor relevantních znalostí o vytčené části reálného světa zapsaných ve vhodném objektovém jazyce, který je součástí formálního jazykového systému se syntaktickou inferencí. Hodnota intenzionální vágnosti (z toho tedy hodnota sémantického diferenciálu) významu všech jazykových konstrukcí systému je nulová. Systém je soubor jmen atributů (tato jména mají konsenzuální exaktní interpretaci čili nulovou hodnotu sémantického diferenciálu) a vztahů popisujících formálním jazykem znalosti - hypotézy rozpoznané mezi atributy. Systém je účelově definovaný soubor komponent, mezi kterými existují určité vztahy, a které splňují nějaký cíl. Systémová analýza se zabývá systémy vytvořených lidmi, jež se skládají ze vstupů, procesů a výstupů.



# 1 Informační systémy

Informační systém je obecným systémem, který pracuje s konceptuálními zdroji. Skládá se ze vstupní a výstupní části, kudy se do systému vkládají, resp. získávají informace (data, která interpretuje uživatel jako informace). Mezi vstupem a výstupem probíhá transformace těchto dat.

Transformační část typicky provádí nad daty různé algoritmy. Probíhají zde *procesy* a zabýváme se zde způsobem definice procesů, jejich vzájemnou interakcí, paralelním prováděním apod.

Typickou částí informačního systému je zpětná vazba, která využívá uloženého *stavu* systému. Ne všechny výstupy procesů jsou tedy přímo závislé na okamžitých vstupech, ale typicky závisejí na stavu systému.

## 1.1 Hranice a okolí

Hranice systému vymezuje samotný systém nebo odděluje více systémů. *Logická hranice* je pomyslnou hranicí a vymezuje podsystémy v rámci systému, ovšem *okolí systému* je již „viditelnou“ hranicí. Prvky vně hranice pak ovlivňují chování systému.

## 1.2 Rozdělení systémů

Systémy dělíme na:

- uzavřené x otevřené,
- deterministické x stochastické,
- statické x dynamické,
- spojité x diskrétní.

## 1.3 Řízení a zpětná vazba

V systémech může nastat zpětná vazba, kdy výstupní veličina opětovně ovlivňuje vstupní veličinu, a tedy i samotný systém. Každý systém má tedy tendence být *nestabilní*, a proto může někdy pomoci implementace tzv. regulátorů. Je zde vidět analogie s logickými obvody [2].

## 2 Databáze

Databáze je určitá uspořádaná množina informací (dat) uložená na paměťovém médiu. V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Tento systém se v české odborné literatuře nazývá systém řízení báze dat (SŘBD).

### 2.1 Historie

Předchůdcem databází byly papírové kartotéky. Umožňovaly uspořádávání dat podle různých kritérií a zařídování nových položek. Veškeré operace s nimi prováděl přímo člověk. Správa takových kartoték byla v mnohém podobná správě dnešních databází.

Dalším krokem bylo převedení zpracování dat na stroje. Za první velké strojové zpracování dat lze asi považovat sčítání lidu ve Spojených státech v roce 1890. Paměťovým médiem byl dřevěný štítek a zpracování sebraných informací probíhalo na elektromechanických strojích.

Velkým impulsem pro další rozvoj databází byl překotný vývoj počítačů v padesátých letech 20. století. Ukázalo se, že původně univerzální používání strojového kódu procesorů je (nejen) pro databázové úlohy neefektivní, a proto se objevil požadavek na vyšší jazyk pro zpracování dat.

V roce 1959 se konala konference zástupců firem, uživatelů a amerického ministerstva obrany, jejímž závěrem byl požadavek na univerzální databázový jazyk. Výsledkem byla o rok později na konferenci CODASYL publikovaná první verze jazyka COBOL, který byl po mnoho dalších let nejrozšířenějším jazykem pro hromadné zpracování dat.

V roce 1965 na konferenci CODASYL byl vytvořen výbor Database Task Group (DBTG), který měl za úkol vytvořit koncepci databázových systémů. Začaly vznikat první síťové SŘBD na sálových počítačích. Jedním z prvních průkopníků databází byl Charles Bachman.

V roce 1971 vydal výbor zprávu The DBTG April 1971 Report, kde se objevily pojmy jako schéma databáze, jazyk pro definici schématu, subschéma a podobně. Byla zde popsána celá architektura síťového databázového systému.

Ve stejné době byly vyvíjeny i hierarchické databáze. Jedním z prvních SŘBD byl IMS, který byl vyvinut firmou IBM pro program letu na Měsíc Program Apollo. Systém IMS patří stále k nejrozšířenějším na sálových počítačích.

V roce 1970 začínají zveřejněním článku E. F. Codd první relační databáze, které pohlížejí na data jako na tabulky. Kolem roku 1974 se vyvíjí první verze dotazovacího jazyka SQL. Vývoj této technologie po 10 letech přinesl výkonově použitelné systémy, strovnatelné se síťovými a hierarchickými databázemi.

V 90. letech 20. století se začínaly objevovat první objektově orientované databáze, jejichž filozofie byla přebírána z objektově orientovaných jazyků. Tyto databáze měly podle předpokladů

vytlačit relační systémy. Původní předpoklady se však nenaplnily a vznikla kompromisní objektově-relační technologie.

## 2.2 Databázové modely

Z hlediska způsobu ukládání dat a vazeb mezi nimi můžeme rozdělit databáze do základních typů:

- hierarchická databáze,
- síťová databáze,
- relační databáze,
- objektová databáze,
- objektově relační databáze [3].

## 2.3 Relační databáze

Relační databáze je databáze, založená na relačním modelu. Často se tímto pojmem označuje nejen databáze samotná, ale i její konkrétní softwarové řešení.

Je založena na tabulkách, které uchovávají informace o relacích v matematickém slova smyslu.

### 2.3.1 Terminologie

Základem relačních databází jsou databázové tabulky. Jejich sloupce se nazývají atributy, řádky, jsou pak záznamy. Atributy mají určen datový typ. Řádek je průmětem sloupců tabulky a slouží k vlastnímu uložení dat. Konkrétní tabulka pak realizuje podmnožinu kartézského součinu možných dat všech sloupců – relaci.

### 2.3.2 Klíč

Klíč je jednoznačný identifikátor záznamu, řádku tabulky. Klíčem může být jediný sloupec či kombinace více sloupců tak, aby byla zaručena jednoznačnost. Pole klíče musí obsahovat hodnotu, tzn. nesmí se zde vyskytovat nedefinovaná prázdná hodnota NULL. V praxi se často používají umělé klíče, což jsou číselné či písmenné identifikátory. Každý nový záznam dostává identifikátor větší než záznam předchozí. Dalším důležitým pojmem jsou nevlastní klíče. Slouží pro vyjádření vztahů, relací, mezi databázovými tabulkami. Jedná se o pole či skupinu polí, která nám umožní identifikovat, které záznamy z různých tabulek spolu souvisí.

### 2.3.3 Integrita databáze

Integrita databáze znamená, že data v ní uložená jsou konzistentní. Lze zadávat pouze data, která vyhovují předem definovaným kritériím (např. musí respektovat datový typ nastavený pro daný

sloupec tabulky). K zajištění integrity slouží integritní omezení. Jedná se o nástroje, které zabrání ztrátě či poškození stávajících záznamů v průběhu práce s databází. Tímto způsobem je možné zajistit mazání dat, která již ztratila svůj význam. Například smažeme-li uživatele, odstraní se i zbytek jeho záznamů v ostatních databázových tabulkách.

### 2.3.4 Druhy integritních omezení

- entitní integritní omezení,
- doménová integritní omezení,
- referenční integritní omezení,
- aktivní referenční integrita.

### 2.3.5 Vztahy mezi tabulkami

Vztahy, neboli relace, slouží ke svázání dat, která spolu souvisejí a jsou umístěny v různých databázových tabulkách. V zásadě rozlišujeme čtyři typy vztahů.

- Mezi daty v tabulkách není žádná spojitost, proto nedefinujeme žádný vztah.
- 1:1 používáme, pokud záznamu odpovídá právě jeden záznam v jiné databázové tabulce a naopak.
- 1:N přiřazuje jednomu záznamu více záznamů z jiné tabulky. Jedná se o nejpoužívanější typ relace, jelikož odpovídá mnoha situacím v reálném životě.
- M:N je méně častým. Umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé. V databázové praxi bývá tento vztah z praktických důvodů nejčastěji realizován kombinací dvou vztahů 1:N a 1:M, které ukazují do pomocné tabulky složené z kombinace obou použitých klíčů (třetí resp. tzv. vazební tabulka).

### 2.3.6 Normální formy

Pod pojmem normalizace rozumíme proces zjednodušování a optimalizace navržených struktur databázových tabulek. Hlavním cílem je navrhnout databázové tabulky tak, aby obsahovaly minimální počet redundantních dat. Správnost navržení struktur lze ohodnotit některou z následujících normálních forem.

- Nultá normální forma (0NF) - tabulka v nulté normální formě obsahuje alespoň jeden sloupec (atribut), který může obsahovat více druhů hodnot.
- První normální forma (1NF) - tabulka je v první normální formě, pokud všechny sloupce (atributy) nelze dále dělit. Jeden sloupec neobsahuje více druhů hodnot.
- Druhá normální forma (2NF) - tabulka je v druhé normální formě, pokud obsahuje

pouze atributy (sloupce), které jsou závislé na klíči.

- Třetí normální forma (3NF) - tabulka je ve třetí normální formě, pokud neexistují žádné vztahy mezi neklíčovými atributy (sloupci).
- Čtvrtá normální forma (4NF) - tabulka je ve čtvrté normální formě, pokud sloupce (atributy) v ní obsažené popisují pouze jeden fakt nebo jednu souvislost.
- Pátá normální forma (5NF) - tabulka je v páté normální formě, pokud by se přidáním libovolného nového sloupce (atributu) rozpadla na více tabulek [4].

## 3 Použité technologie

### 3.1 Volba technologií

Pro tvorbu mé bakalářské práce, informačního systému, jsem si vybral skriptovací jazyk PHP a z menší části jazyk JavaScript.

Pro volbu PHP jsem se rozhodl proto, že je velmi oblíbený především díky své jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s databázovým serverem (především s MySQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací.

Databázi jsem zvolil MySQL, pro svou snadnou implementovatelnost, výkon a především díky tomu, že se jedná o volně dostupný software.

Pro úpravu vzhledu informačního systému jsem se rozhodl použít značkovací jazyk HTML, přesněji řečeno jeho rozšířenou verzi XHTML spolu s využitím technologií tzv. kaskádových stylů CSS.

### 3.2 Popis technologií

#### 3.2.1 HTML

HyperText Markup Language, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language).

V roce 1989, ve výzkumném centru fyziky CERNu, spolupracovali Tim Berners-Lee a Robert Caillau na propojeném informačním systému. V té době se pro tvorbu dokumentů obvykle používaly jazyky TeX, PostScript a také SGML. Berners-Lee si uvědomoval, že potřebují něco jednoduššího a v roce 1990 byl tedy navržen jazyk HTML a protokol pro jeho přenos v počítačové síti – HTTP (HyperText Transfer Protocol – přenosový protokol hypertextu). Zároveň také Tim Berners-Lee napsal první webový prohlížeč, který nazval WorldWideWeb.

##### 3.2.1.1 HTML 2.0

Zachycuje stav jazyka v polovině roku 1994. Standard vydala komunita IETF (*Internet Engineering Task Force*). Je to první verze, která odpovídá syntaxi SGML.

### **3.2.1.2 HTML 3.2**

Byla vydána 14.1. 1997 a zachycuje stav jazyka v roce 1996. Přípravovaná verze HTML 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá a žádná firma nebyla schopna naprogramovat její podporu. Přidává k jazyku tabulky, zarovnávání textu a stylové elementy pro ovlivňování vzhledu.

### **3.2.1.3 HTML 4.0**

Byla vydána 18.12. 1997. Do specifikace jazyka přibyly nové prvky pro tvorbu tabulek, formulářů a nově byly standardizovány rámy.

### **3.2.1.4 HTML 4.01**

Byla vydána 24.12. 1999. Tato verze opravuje některé chyby verze předchozí. Podle původního předpokladu se mělo jednat o poslední verzi, po které by se přešlo na XHTML.

### **3.2.1.5 HTML 5**

7.3. 2007 byla založena nová pracovní skupina HTML, jejíž cílem je vývoj nové verze HTML. V květnu 2007 bylo odhlasováno, že základem nové specifikace se stanou Web Applications 1.0 a Web Forms 2.0 ze specifikace WHATWG. Jako název nové specifikace bylo odhlasováno HTML 5. Specifikace by měla být hotova v letech 2010 – 2012 [5].

## **3.2.2 PHP**

Označení PHP bylo původně zkratkou anglické fráze "Personal Home Page". Tuto technologii vytvořil v roce 1994 Rasmus Lerdorf kvůli sledování návštěvníků svých stránek. S postupným nárůstem užitečnosti a možností této technologie (postupně se začala prosazovat ve stále profesionálnějších řešeních) se ujal název "PHP: Hypertext Preprocessor".

Podle oficiální webové prezentace PHP, je "PHP vloženým skriptovacím jazykem". Je to komplexní, ale přesto výstižná definice, jejíž význam si nyní vysvětlíme.

Když někdo řekne, že je PHP *vložen do HTML*, znamená to, že jej lze interpretovat přímo v kódu HTML, díky čemuž je vývoj dynamických webových prezentací snáze dostupný. PHP je navíc *skriptovací*, nikoli programovací jazyk. Jazyk PHP je navržen, aby vykonal určitou činnost jako reakci na výskyt určité události.

PHP technologie je nezávislá na platformě, určená pro servery. Oba tyto faktory jsou velmi důležité. Skutečnost, že jde o *serverovou* technologii, znamená, že vše, co se v kódu PHP odehrává, se odehrává na serveru (nikoli na klientovi, jímž je počítač návštěvníka webového serveru). Fakt, že jde o technologii *nezávislou na platformě*, znamená, že jazyk PHP lze používat na většině operačních systémů Microsoft Windows, Unix (v mnoha jeho variantách včetně Linuxu) a Macintosh. Ještě

důležitější je však skutečnost, že skript napsaný na jednom serveru bude fungovat prakticky bez úprav (nebo s minimálními změnami) také na jakémkoli jiném serveru.

Mezi přednosti jazyka PHP patří jeho výkon, těsná integrace s většinou dostupných databázových systémů, stabilita, přenositelnost a téměř neomezené možnosti rozšiřování [6].

### **3.2.2.1 PHP/FI**

PHP je nástupcem staršího produktu, nazvaného PHP/FI. PHP/FI vytvořil Rasmus Lerdorf v roce 1995, na počátku jako jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech k jeho webu. Tuto sadu nazval Personal Home Page Tools. Protože byla potřeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v jazyku C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro web. Rasmus se rozhodl uvolnit zdrojový kód PHP/FI pro všechny, takže kdokoli ho může používat, stejně jako opravovat chyby a vylepšovat kód.

PHP/FI, což znamená Personal Home Page/Forms Interpreter, obsahovalo něco ze základní funkcionality PHP, jak ho známe dnes. Mělo proměnné perlovského typu, automatickou interpretaci formulářových proměnných a syntaxi vloženou do HTML. Syntaxe samotná byla podobná jazyku Perl, přestože mnohem omezenější, jednodušší a v něčem nekonzistentní.

V roce 1997 se PHP/FI 2.0, druhá implementace psaná v C, stala kultovní záležitostí pro (odhadem) tisíce uživatelů po celém světě a přibližně 50 000 domén oznamujících nainstalované PHP/FI, což čítalo zhruba 1 % všech domén na Internetu. I když do projektu začalo svými kusy kódu přispívat více lidí, stále to byl velký projekt jednoho muže.

### **3.2.2.2 PHP 3**

PHP 3.0 byla první verze, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski v roce 1997 jako kompletně přepsaný celek poté, co shledali PHP/FI 2.0 výrazně "poddimezovaným" pro vývoj svých aplikací pro e-komerci. Ve snaze spolupracovat a zahájit budování nad existující uživatelskou základnou PHP/FI, rozhodli se Andi, Rasmus a Zeev pracovat společně a prohlásit PHP 3.0 za oficiálního nástupce PHP/FI 2.0 a vývoj PHP/FI 2.0 byl v podstatě zastaven.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. Kromě pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly především možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka.



### **3.2.2.3 PHP 4**

V zimě 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přepisu jádra PHP. Cílem návrhu bylo zlepšit modularitu kódové báze PHP a zvýšit výkon pro složité aplikace. Tyto aplikace byly schopny pracovat s PHP 3.0 (díky novým možnostem a podpoře široké škály databází a API od jiných tvůrců), ale PHP 3.0 nebylo navrženo pro efektivní práci tak náročných aplikací.

Nový engine, nazvaný Zend Engine (název byl sestaven z jejich křestních jmen - Zeev a Andi), úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho různých WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

### **3.2.2.4 PHP 5**

Na vývoji PHP 5 se začalo pracovat již v roce 2002. Základem této verze je zcela přepracovaný Zend Engine 2, který jednak přinesl vyšší výkon kritických PHP aplikací, jednak umožnil zakomponovat do PHP řadu pokročilejších programovacích struktur, čímž především otevřel cestu kvalitnější podpoře objektově orientovaného programování v PHP.

PHP 5 je do značné míry zpětně kompatibilní s PHP 4. V nové verzi jazyka byly především posíleny bezpečnostní mechanismy (což může vést k nefunkčnosti některých špatně postavených aplikací pro PHP 4) a uveden nový, podstatně kvalitnější objektový model, umožňující používat PHP jako skutečný objektově orientovaný jazyk [7].

## **3.2.3 CSS**

CSS je zkratka pro anglický název Cascading Style Sheets, česky tabulky kaskádových stylů. Je to jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.

Byly vydány zatím dvě úrovně specifikace CSS1 a CSS2, dokončuje se revize CSS 2.1 a pracuje se na verzi CSS3.

Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentů a vyhledávání informací není takový vývoj žádoucí [8].

### 3.2.3.1 CSS1

První velmi rozšířený prohlížeč, který trochu podporoval CSS, byl Internet Explorer verze 3 z roku asi 1996. Podpora CSS se omezovala na písma a barvy. Dost velká mizérie, ale zároveň vlašťovka. Slušné podpory se kaskádové styly dočkaly až ve čtvrtých verzích prohlížečů Internet Explorer a Netscape Navigator (asi 1998). Tyto prohlížeče se snaží podporovat celé CSS1 včetně pozicování.

Přesto nebyla podpora různých CSS vlastností nijak slavná, zejména u Netscapu. Tento prohlížeč se chová chaoticky a spoustu vlastností nepodporuje. To je i hlavní důvod, proč se kaskádové styly dlouho nedostaly do oblíbenosti širokých vrstev webmasterů.

### 3.2.3.2 CSS2

CSS2 je druhá verze kaskádových stylů. Obsahuje další vlastnosti pro formátování písma, ale i spoustu dalších praktických věcí. Konečná verze se ustanovila někdy kolem roku 2000.

Pomalou měrou se prosazuje do nových prohlížečů. Má ale širší ambice - má se stát formátovacím jazykem i pro jiné než webové platformy a i pro jiné jazyky než HTML. Už nyní se CSS2 běžně používá pro formátování XML dokumentů.

Nástup podpory CSS2 ve webových prohlížečích probíhá jakousi difúzí. Žádný nový prohlížeč nepodporuje z CSS2 všechno, ale jenom něco. Nejlépe je na tom prohlížeč Mozilla, ta se snaží pečlivě dodržovat standardy. Prohlížeče Microsoftu v každé nové podverzi vylepšují podporu CSS2, ale plné podpory zatím nedosáhly.

Hlavní finesou CSS2 jsou různá média, na které se zaměřuje. Kromě obrazovkového výstupu se totiž styly dají použít na odlišný tisk, na zobrazení na mobilech, projekcích, či dokonce na hlasový výstup nebo zařízení pro slepecké písmo. Už v nynějších prohlížečích je podpora odlišného tisku.

Vývoj jednoznačně ukázal, že stylové jazyky jsou pro publikační technologie nezbytné. Obzvláště silný význam budou mít v kombinaci s XML. V budoucnu se dá očekávat doladění podpory CSS2 nejen ve webových prohlížečích, ale i lepší podporu v editorech. Například i Microsoft Word se snaží zapisovat formát svých souborů pro web v pojmech CSS a kde může, používá CSS2 [9].

## 3.2.4 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape.

Nyní se zpravidla používá jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer

Manufacturers Association) a v srpnu 1998 ISO (International Organization for Standardisation). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript.

JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem Mocha, později LiveScript, ohlášen byl společně se společností Sun Microsystems v prosinci 1995 jako doplněk k jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript.

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.

JavaScript je možné použít i na straně serveru. První implementací JavaScriptu na straně serveru byl LiveWire firmy Netscape vypuštěný roku 1996, dnes existuje několik možností včetně opensource implementace Rhinola založená na Rhino, gcj a Apache [10].

### **3.2.5 MySQL**

Aplikace MySQL byla vyvinuta a je udržována švédskou společností MySQL AB. Jde o systém správy databází (DBMS, database management system) určený pro relační databáze. Relační databáze je kolekcí vzájemně provázaných dat uložených v podobě textu, čísel nebo binárních souborů řízenou právě systémem správy DBMS.

MySQL je nejoblíbenější a podle mnohých také nejlepší databázový systém s veřejným zdrojovým kódem. Ve skutečnosti, zejména když byly ve verzi 4 do systému přidány nové funkce, je systém MySQL životaschopným konkurentem drahých Goliášů, jakými jsou Oracle nebo Microsoft SQL Server. Stejně jako jazyk PHP nabízí také systém MySQL výtečný výkon, přenositelnost a spolehlivost s přiměřenou dobou zaučení a minimální náklady na provoz [11].

## **4 Specifikace požadavků**

### **4.1 Popis vlastního zadání**

Zadání této bakalářské práce je navrhnout a implementovat informační systém obecního úřadu. Proto jsem se nejprve musel seznámit s problematikou informačních systémů. Dále pak prohloubit své znalosti tvorby dynamických www stránek a skriptovacích jazyků, jakou jsou PHP a JavaScript.

### **4.2 Neformální specifikace**

Cílem této práce je navrhnout a implementovat informační systém obce, který bude možno ovládat pomocí standardních webových prohlížečů.

Tento informační systém by měl mít chráněný přístup pro oprávněné uživatele, kteří jsou vedeni v databázi.

Po přihlášení uživatele do systému by měl mít každý uživatel přístup jen k datům a funkcím dle jeho nastavených práv.

Uživatel se svým příslušným oprávněním má možnost:

- spravovat uživatele systému,
- spravovat databázi firem,
- spravovat databázi zaměstnanců obecního úřadu,
- spravovat sekci fotogalerie,
- spravovat sekci ubytování,
- spravovat sekci sport,
- spravovat databázi článků,
- spravovat sekci kalendář kulturních akcí.

### **4.3 Technické požadavky**

Informační systém by měl být implementován za použití:

- skriptovacího jazyka PHP a JavaScript,
- relační databáze MySQL,
- technologie CSS a XHTML.

# 5 Analýza a návrh řešení

Podle rozboru požadavků jsem se snažil vhodně navrhnout daný informační systém.

Informační systém jsem rozdělil na dvě základní části:

- **prezentační systém**, který je volně přístupný každému uživateli.
- **vnitřní systém**, do kterého mají přístup pouze oprávnění uživatelé na základě jejich přihlašovacího jména a hesla.

## 5.1 Návrh databáze

Návrh databáze byla jednou z prvních věcí, kterou jsem začal. Její návrh je velmi důležitý, protože špatně navrhnutá databáze může danou situaci značně zkomplikovat.

### 5.1.1 Popis tabulek

#### 5.1.1.1 Tabulka firmy

Tabulka *firma* obsahuje informace jako je *název*, *adresa* a *kontakt* o jednotlivých firmách, které jsou uloženy v databázi. Jednoznačným identifikátorem tabulky *firma* je její *ID*.

#### 5.1.1.2 Tabulka obec

Tato tabulka nese informace o osobách - zaměstnancích obecního úřadu. Jednoznačným identifikátorem tabulky *obec* je její *ID*. Atribut *funkce* udává pozici, funkci jednotlivých zaměstnanců. Další atributy nesou kontaktní informace zaměstnance.

#### 5.1.1.3 Tabulka ubytování

Tato tabulka nese informace o ubytování v obci. Jednoznačným identifikátorem je *ID*. Atribut *úvod* udává informace o krátkém uvedení do článku a atribut *celytext* nese informace o celém textu. Další atributy udávají kontaktní informace.

#### 5.1.1.4 Tabulka ubytování\_img

Tato tabulka nese informace o obrázcích, které jsou umístěny na konci každého článku. Jednoznačným identifikátorem obrázku je *ID*. Atribut *id\_ubytovani* nese informaci, ke kterému ubytování obrázek přísluší. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.5 Tabulka ubytování\_ikona**

Tato tabulka nese informace o obrázku, který slouží jako ikona k danému článku. Jednoznačným identifikátorem obrázku je *ID*. Atribut *id\_ubytovani* nese informaci, ke kterému ubytování obrázek přísluší. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.6 Tabulka lec\_ubytování**

Tato tabulka obsahuje informace o léčebných pobytech v obci. Jednoznačným identifikátorem je *ID*. Atribut *úvod* udává informace o krátkém uvedení do článku a atribut *celytext* nese informace o celém textu. Další atributy udávají kontaktní informace.

#### **5.1.1.7 Tabulka lec\_ubytování\_img**

Tato tabulka obsahuje informace o obrázcích, které jsou umístěny na konci každého článku. Jednoznačným identifikátorem obrázku je *ID*. Atribut *id\_ubytovani* nese informaci, ke kterému léčebnému ubytování obrázek přísluší. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.8 Tabulka lec\_ubytování\_ikona**

Tato tabulka nese informace o obrázku, který slouží jako ikona k danému článku. Jednoznačným identifikátorem obrázku je *ID*. Atribut *id\_ubytovani* nese informaci, ke kterému léčebného ubytování obrázek přísluší. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.9 Tabulka sport\_info**

Tato tabulka nese informace o sportovních aktualitách v obci. Jednoznačným identifikátorem je *ID*. Atribut *úvod* udává informace o krátkém uvedení do aktuality a atribut *celytext* nese informace o celém textu. Atributy *nazev* udává informace o názvu aktuality. Dále atribut *datum* nese informaci o datumu, kdy byl článek napsán a atribut *login* uchovává jméno uživatele, který článek napsal.

#### **5.1.1.10 Tabulka sport\_img**

Tato tabulka obsahuje informace o obrázcích, které jsou umístěny na konci každé aktuality. Jednoznačným identifikátorem obrázku je *ID*. Atribut *id\_sport* nese informaci, ke které aktualitě obrázek přísluší. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.11 Tabulka sport\_ikona**

Tato tabulka nese informace o obrázku, který slouží jako ikona k dané aktualitě. Jednoznačným identifikátorem obrázku je *ID*. Atribut *id\_sport* nese informaci, ke které aktualitě obrázek přísluší. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.12 Tabulka guestbook**

Tato tabulka nese informace o uživatelích, kteří navštívili diskuzní fórum a zanechali nějaký příspěvek. Jednoznačným identifikátorem této tabulky je *ID* napsaného příspěvku. Atributy *name*, *email* a *url* nesou kontaktní informace o uživateli, který napsal příspěvek. Atribut *message* nese informace o příspěvku zanechaném uživatelem a atribut *entry\_date* nese informace o datumu, kdy byl příspěvek napsán.

#### **5.1.1.13 Tabulka fotogalerie**

Tato tabulka nese informace o obrázcích, které se nacházejí v sekci fotogalerie. Jednoznačným identifikátorem obrázku je *ID*. Atribut *typ* udává, ke které kategorii bude obrázek přiřazen. Dalšími atributy jsou *name* - informaci o názvu obrázku, *type* - informace o formátu obrázku, *size* - informace o velikosti obrázku, *content* - typ.

#### **5.1.1.14 Tabulka users**

Tabulka nese informace o uživateli, který má přístup do vnitřní části systému. Jednoznačným identifikátorem je *login*. Další povinný atribut je *heslo*. Zbývající atributy nesou kontaktní informace uživatele.

#### **5.1.1.15 Tabulka články**

Tato tabulka nese informace o člancích, které se týkají obce. Jednoznačným identifikátorem je *ID*. Atribut *kategorie* udává do jaké sekce článek patří. Atribut *úvod* udává informace o krátkém uvedení do článku a atribut *celytext* nese informace o celém článku.

#### **5.1.1.16 Tabulka úřední\_deska**

Tato tabulka nese informace o zprávách, které se týkají obecního úřadu. Jednoznačným identifikátorem je *ID*. Atribut *úvod* udává informace o krátkém uvedení zprávy a atribut *celytext* nese informace o celém článku. Dále atribut *datum* nese informaci o datumu, kdy byla zpráva vyvěšena a atribut *login* uchovává jméno uživatele, který zprávu napsal.

## **6 Informační systém obecního úřadu**

### **6.1 Vnitřní systém**

#### **6.1.1 Přihlášení do systému**

Do toho systému mají přístup, jak už jsem zmínil výše, pouze uživatelé s příslušným oprávněním. Aby se uživatel mohl dostat do této části systému, musí zadat své přihlašovací jméno a heslo.

Všechna důležitá data jsou ukládána do proměnné typu *session*. Ukládám zde login, práva, jméno a příjmení uživatele. Tím je zajištěna identifikace uživatele po celou dobu jeho aktivity v systému.

Zde bych ještě zdůraznil, že proměnná typu *session* neboli relace umožňuje přesnou identifikaci uživatele a pohyb na serveru. Za pomoci *session* jsou naprogramovány všechny web-based mailové servery. Má tedy použití všude tam, kde je zapotřebí monitorovat nebo ověřovat uživatele, například pomocí jeho IP adresy.

#### **6.1.2 Odhlášení ze systému**

Odhlášení ze systému probíhá tak, že jsou všechny tyto proměnné *session* na serveru odstraněny.

#### **6.1.3 Správa firem**

Uživatel má možnost spravovat databázi firem. Má přehled o informacích (název, adresa, kontakt) všech firem, které se nacházejí v databázi informačního systému. Zde jsou jeho možnosti následující:

- může přidávat novou firmu do databáze,
- jednotlivé firmy může editovat a vzápětí uložit změny,
- a nebo může firmu smazat z databáze.

#### **6.1.4 Správa zaměstnanců obecního úřadu**

Zde je to téměř stejné jako u správy firem. Uživatel má opět přehled o informacích (funkce, jméno, kontakt) všech zaměstnanců, kteří pracují na obecním úřadě dané obce. Uživatelovy volby jsou následující:

- může přidávat nového zaměstnance do databáze,
- jednotlivé zaměstnance může editovat a vzápětí uložit změny,



- a nebo může zaměstnance smazat z databáze.

### **6.1.5 Správa úřední desky**

Uživatel má možnost aktualizovat stav úřední desky a to tak, že:

- může přidávat nové zprávy, které se udály na schůzi Rady města,
- má možnost jednotlivé zprávy editovat,
- může staré zprávy smazat z databáze.

### **6.1.6 Správa diskuzního fóra**

Zde má uživatel přehled o všech příspěvcích, které zanechali návštěvníci v diskuzním fóru. Každý příspěvek obsahuje informace o návštěvníkovi, jako je jeho jméno, email, url adresa, text zprávy a datum, kdy byl příspěvek vložen. Uživatel zde má jen jednu možnost a tou je:

- smazání kteréhokoliv příspěvku.

### **6.1.7 Správa kalendáře akcí**

Zde má uživatel možnost volby zadávat kulturní akce, které se budou v obci konat během aktuálního měsíce. O všech akcích, které jsou v databázi uloženy, má přehled a zároveň je může:

- editovat
- a akce, které nejsou aktuální, může smazat z databáze.

### **6.1.8 Správa sportovních aktualit**

Zde může uživatel přidávat nové sportovní aktuality, které se v obci dějí. Vedle toho má ještě možnost je editovat a starší informace, které nejsou aktuální může smazat z databáze.

#### **6.1.8.1 Přidávání sportovních aktualit**

Pokud chce uživatel přidat novou sportovní aktualitu, musí nejprve kliknout na tlačítko *přidat sport-info*. Vzápětí se mu zobrazí vstupní textové pole, kam může napsat název sportovní činnosti. Dále má k dispozici dva textové editory, z nichž první slouží pro krátké uvedení do děje a druhý pro zobrazení celého textu. V těchto textových editorech může uživatel text libovolně upravovat, tzn. zvýraznit text, použít kurzívu, vkládat text, používat seznamy atd.

### **6.1.8.2 Vkládání obrázků ke sportovní aktualitě**

Uživatel má možnost vložit obrázky ke každé aktualitě. Po kliknutí na ikonu *přidat obrázky* se mu nabízí dvě volby:

- Jedna z možností je ta, že může vložit obrázek jako ikonu k dané aktualitě.
- A druhou možností je vložení obrázků do fotogalerie, která je součástí každé sportovní aktuality.

Pokud uživatel není spokojený s výběrem vloženého obrázku, může ho kdykoliv smazat a vložit jiný.

## **6.1.9 Správa ubytování v obci**

Zde může uživatel přidávat nové informace o ubytování v obci v podobě krátkých článků. Vedle toho se mu nabízí možnost každý článek editovat a smazat z databáze.

### **6.1.9.1 Přidávání nového ubytování**

Pokud chce uživatel přidat informace o novém ubytování, musí nejprve kliknout na tlačítko *přidat ubytování*. Vzápětí se mu zobrazí vstupní textová pole, kam může napsat název, adresu a kontakt ubytování. Dále má k dispozici dva textové editory, z nichž první slouží pro krátké uvedení do děje a druhý pro zobrazení celého textu. V těchto textových editorech může uživatel opět text libovolně upravovat, tzn. zvýraznit text, použít kurzívu, vkládat text, používat seznamy atd.

### **6.1.9.2 Vkládání obrázků k ubytování**

Součástí přidávání nového ubytování je i možnost vložit obrázky ke každému ubytování. Po kliknutí na ikonu *přidat obrázky* má uživatel dvě volby:

- Jedna z možností je ta, že může vložit obrázek jako ikonu k danému ubytování.
- A druhou možností je vložení obrázků do fotogalerie, která je součástí každého ubytování.

Pokud uživatel není spokojený s výběrem vloženého obrázku, může ho kdykoliv smazat a vložit jiný.

## 6.1.10 Správa léčebných pobytů v obci

Zde má uživatel ty samé volby jako u *správy ubytování*. Rozdíl je jen ten, že se nejedná o běžné ubytování, ale o léčebný pobyt v lázních.

## 6.1.11 Správa článků o obci

Zde má uživatel přehled o všech článcích, které se týkají informací o obci. Jsou zde zahrnuty články sekcí:

- *Kde Nás najdete,*
- *Městské symboly,*
- *Osobnosti,*
- *Památky,*
- *Kam na výlet,*
- *Léčivé prameny,*
- *Kultura,*
- *Turistika.*

Jednotlivé články mají přiřazenou kategorii dle toho, k jaké sekci se přísluší. Uživatel má přehled o všech těchto článcích. Jeho možnosti jsou zde takové, že jednotlivé články může upravovat.

Zde bych chtěl zdůraznit jednu věc a to tu, že články lze pouze upravovat. Udělal jsem to zde z toho důvodu, protože si myslím, že jsou to nejdůležitější a nejvýznamnější informace, kterými se obec vyznačuje. Pokud se uživatel o některé z nich dozví aktuality, může si daný článek vyhledat a upravit jej.

## 6.1.12 Správa fotogalerie

Zde může uživatel aktualizovat fotogalerii dané obce. Fotogalerie obsahuje tři sekce a uživatel má možnost v každé z nich obrázky editovat. Pokud uživatel není spokojený s výběrem vloženého obrázku, může ho kdykoliv smazat a vložit jiný.

### 6.1.12.1 Vkládání obrázků do fotogalerie

Pokud chce uživatel vložit nový obrázek do fotogalerie, musí si nejprve vybrat jednu z uvedených sekcí, do které chce obrázek vložit.

## **6.2 Prezentační systém**

K prezentační vrstvě systému mají přístup všichni uživatelé.

### **6.2.1 Sekce Kde Nás najdete**

Pokud se uživatel podívá do této sekce, nalezne zde základní informace o obci, jakými jsou např. název obce, poloha, počet obyvatel.

### **6.2.2 Sekce Městské symboly**

V této sekci může uživatel nalézt něco málo o historii obce, což je znak obce a prapor obce.

### **6.2.3 Sekce Osobnosti**

V této sekci se může uživatel dozvědět o slavných osobnostech, které obec proslavily.

### **6.2.4 Sekce Památky**

Zde se nacházejí nejvýznamnější památky, které jsou pro obec typické.

### **6.2.5 Sekce Ubytování**

Pokud by se chtěl někdo rád podívat do obce a strávit tam několik dní i s noclehem, může mu posloužit tato sekce. Uživatel zde může nalézt zajímavé informace o ubytování. Tato sekce je tvořena formou krátkých článků. Uživatel nejprve spatří krátký úvod s malým obrázkem a pokud se bude chtít dozvědět více, může kliknout na odkaz *...dále...*, kde se mu zobrazí celý text se všemi důležitými informacemi a fotogalerií.

### **6.2.6 Sekce Kam na výlet**

Jestliže se někdo rozhodne navštívit obec a chtěl by zde trávit delší čas, tato sekce mu může posloužit k tomu, jak naložit s volným časem. Nacházejí se zde okolní poznávací místa.

### **6.2.7 Sekce Firmy a služby**

Tato sekce slouží uživateli pro informaci o firmách a službách v obci.

## **6.2.8 Sekce Fotogalerie**

Pokud uživatele zajímá, jak obec vypadá, může navštívit fotogalerii. Má zde na výběr ze tří sekcí:

- Pohled z letadla,
- Historie,
- Roční období.

## **6.2.9 Sekce Zpravodaj**

V této sekci se nachází zpravodaj obce, přesněji řečeno noviny o obci, které jsou vydávány každý měsíc.

## **6.2.10 Sekce Diskuzní fórum**

Jestliže má uživatel nějaké připomínky týkající se stránek, nebo by se rád dozvěděl něco nového o obci, může zde zanechat svůj příspěvek, na který mu bude odpovězeno. Uživatel je povinen zadat své jméno a text příspěvku. Další informace nejsou povinné. Pokud chce, aby mu bylo odpovězeno na email, musí ho zadat.

## **6.2.11 Sekce Léčivé prameny**

Zde může uživatel nalézt informace o léčivých pramenech, které se v obci vyskytují.

## **6.2.12 Sekce Léčebné pobyty**

Pokud by se chtěl někdo rád dozvědět něco o rekreacích či léčebných pobytech v obci, může mu k tomu posloužit tato sekce. Uživatel zde může nalézt zajímavé informace. Tato sekce je tvořena formou krátkých článků. Uživatel nejprve spatří krátký úvod s malým obrázkem a pokud se bude chtít dozvědět více, může kliknout na odkaz *...dále...*, kde se mu zobrazí celý text se všemi důležitými informacemi a fotogalerií.

## **6.2.13 Sekce Úřední deska**

Zde si může uživatel přečíst zprávy z radnice obce.

## **6.2.14 Sekce E-podatelna**

Zde může uživatel zasílat své dotazy na obecní úřad. E-podatelna je naimplementována, na serveru Apache ale bohužel nefunguje.

## **6.2.15 Sekce Kontakty**

Tato sekce slouží uživateli pro informaci o zaměstnancích pracujících na obecním úřadě.

## **6.2.16 Sekce Homepage**

Zde může uživatel spatřit úvodní stranu prezentační vrstvy informačního systému.

## **6.2.17 Sekce Kultura**

V této sekci se uživatel může dozvědět něco o kultuře obce, co se v obci nachází, např. divadlo, muzeum.

## **6.2.18 Sekce Sport**

Zde se uživatel může dočíst o sportovních aktualitách, které se v obci udály. Tato sekce je tvořena formou krátkých článků. Uživatel nejprve spatří krátký úvod s malým obrázkem a pokud se bude chtít dozvědět více, může kliknout na odkaz *...dále...*, kde se mu zobrazí celý text se všemi informacemi a fotogalerií a navíc se mu zde zobrazí, kým byla aktualita napsána a kdy.

## **6.2.19 Sekce Turistika**

V této sekci může uživatel nalézt informace týkající se turistiky, zejména pak pěší turistika a cykloturistika.

## **6.2.20 Sekce Kalendář akcí**

Zde se nacházejí informace o kulturních akcích, které budou probíhat během aktuálního měsíce. Tato sekce je tvořena formou krátkých článků. Uživatel nejprve spatří krátký úvod s malým obrázkem a pokud se bude chtít dozvědět více, může kliknout na odkaz *...dále...*, kde se mu zobrazí celý text se všemi informacemi a navíc se mu zde zobrazí, kým byla kulturní akce napsána a kdy.

## **6.2.21 Sekce Mapa webu**

Slouží uživateli k tomu, aby se mohl zorientovat, co může v jednotlivých sekcích naléznout.

## 7 Závěr

Na počátku tvorby mé bakalářské práce jsem toho o tvorbě dynamických webových stránek moc nevěděl. Týká se to i programování ve skriptovacích jazycích PHP a JavaScript. Něco málo jsem věděl o používání technologií kaskádových stylů CSS. Proto bylo třeba překonat počáteční nezdary a stále se zlepšovat. Domnívám se, že z uvedeného zadání mé bakalářské práce jsem splnil vše, až na zaslání info-SMS a info pošty registrovaným spoluobčanům. Tuto část jsem po dohodě s vedoucím panem Ing. Martinem Strakou vynechal z časových důvodů. Tato práce je pro mě velkým přínosem, protože jsem si poprvé zkusil tvorbu rozsáhlého projektu s časovým omezením a dozvěděl jsem se spoustu nových věcí co se programování dynamických webových stránek týče.

Informační systém by si určitě zasloužil jeho rozšíření. Dala by se zde naimplementovat správa uživatelů, do které by měl přístup jenom administrátor a uživatelům by tak mohl přidělovat jejich práva. Jako další věc, o kterou by se dal informační systém rozšířit, je již zmíněné zaslání info-SMS a info pošty registrovaným spoluobčanům.

# Literatura

- [1] - Hruška, T. *Informační systémy (IIS, PIS)*. Brno, 2007.  
Dostupný na URL:  
<https://www.fit.vutbr.cz/study/courses/WAP/private/opory/OporaIIS2PISPojemDataProcesyTransakce.pdf>.
- [2] – WWW stránky. Otevřená encyklopedie Wikipedie. URL:  
[http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD\\_syst%C3%A9m](http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD_syst%C3%A9m).  
Vloženo v květnu 2008.
- [3] - WWW stránky. Otevřená encyklopedie Wikipedie. URL:  
<http://cs.wikipedia.org/wiki/Datab%C3%A1ze>.  
Vloženo v květnu 2008.
- [4] - WWW stránky. Otevřená encyklopedie Wikipedie. URL:  
[http://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD\\_datab%C3%A1ze](http://cs.wikipedia.org/wiki/Rela%C4%8Dn%C3%AD_datab%C3%A1ze).  
Vloženo v květnu 2008.
- [5] - WWW stránky. Otevřená encyklopedie Wikipedie. URL:  
<http://cs.wikipedia.org/wiki/HTML>.  
Vloženo v květnu 2008.
- [6] - Ullman, L. *PHP a MySQL*. Brno: Computer Press, 2004. 527 s.
- [7] - WWW stránky. Interval.cz - Knihovna PHP. URL:  
<http://php.interval.cz/clanky/historie-a-soucasnost-php/>.  
Vloženo v květnu 2008.
- [8] - WWW stránky. Otevřená encyklopedie Wikipedie. URL:  
[http://cs.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/wiki/Cascading_Style_Sheets).  
Vloženo v květnu 2008.
- [9] - WWW stránky. Jakpsatweb.cz – CSS kurz. URL:  
<http://www.jakpsatweb.cz/css/css-historie.html>.  
Vloženo v květnu 2008.
- [10] - WWW stránky. Otevřená encyklopedie Wikipedie. URL:  
<http://cs.wikipedia.org/wiki/JavaScript>.  
Vloženo v květnu 2008.
- [11] - Ullman, L. *PHP a MySQL*. Brno: Computer Press, 2004. 527 s.



# Seznam příloh

Příloha 1. Manuál

Příloha 2. ER – diagram

Příloha 3. Digram případů použití – administrátor

Příloha 4. Digram případů použití - návštěvníci

Příloha 5. CD/DVD

# Příloha č. 1. Manuál

## Umístění systému

Systém je nutné nainstalovat z příloženého CD.

Všechny potřebné soubory se nacházejí v adresáři **web**.

Je nutné mít nainstalovaný server Apache.

**Prezentační systém:** `localhost/www/web/index.php`

**Vnitřní systém:** `localhost/www/web/admin`

## Přihlášení do IS

Pro účel testování jsem vytvořil tento uživatelský účet:

- administrátor, login: jarda, heslo: 1234

## Instalace z CD

### Instalace databáze

V systému MySQL je nutné vytvořit novou databázi a požadované tabulky. Pro tento účel přikládám soubor *tabulky.sql*, který obsahuje i data pro prezentaci IS.

### Instalace aplikace

Zdrojové soubory je nutné nahrát do prostoru serveru, kde jsou akceptovány soubory webových stránek. Dále je nutné nastavit přístup k databázi v souboru *connect.php*.

```
// Vytvoreni pripojeni a vybrani pracovni databaze

$db = mysql_connect('adresa DB serveru', 'login do DB', 'heslo
do DB');

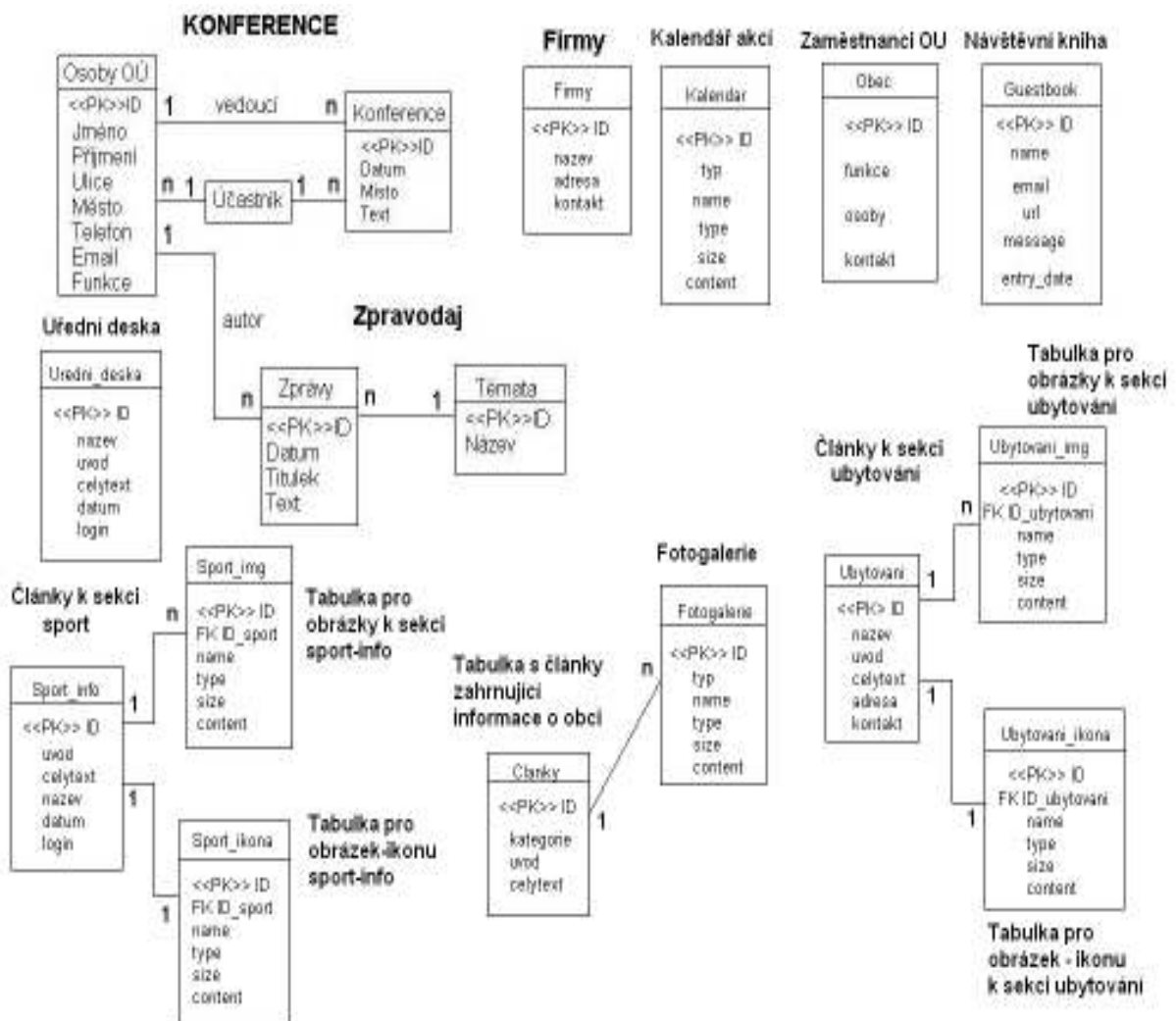
if (!$db) die('nelze se pripojit '.mysql_error());

if (!mysql_select_db('jméno DB', $db)) die('database není
dostupna '.mysql_error());

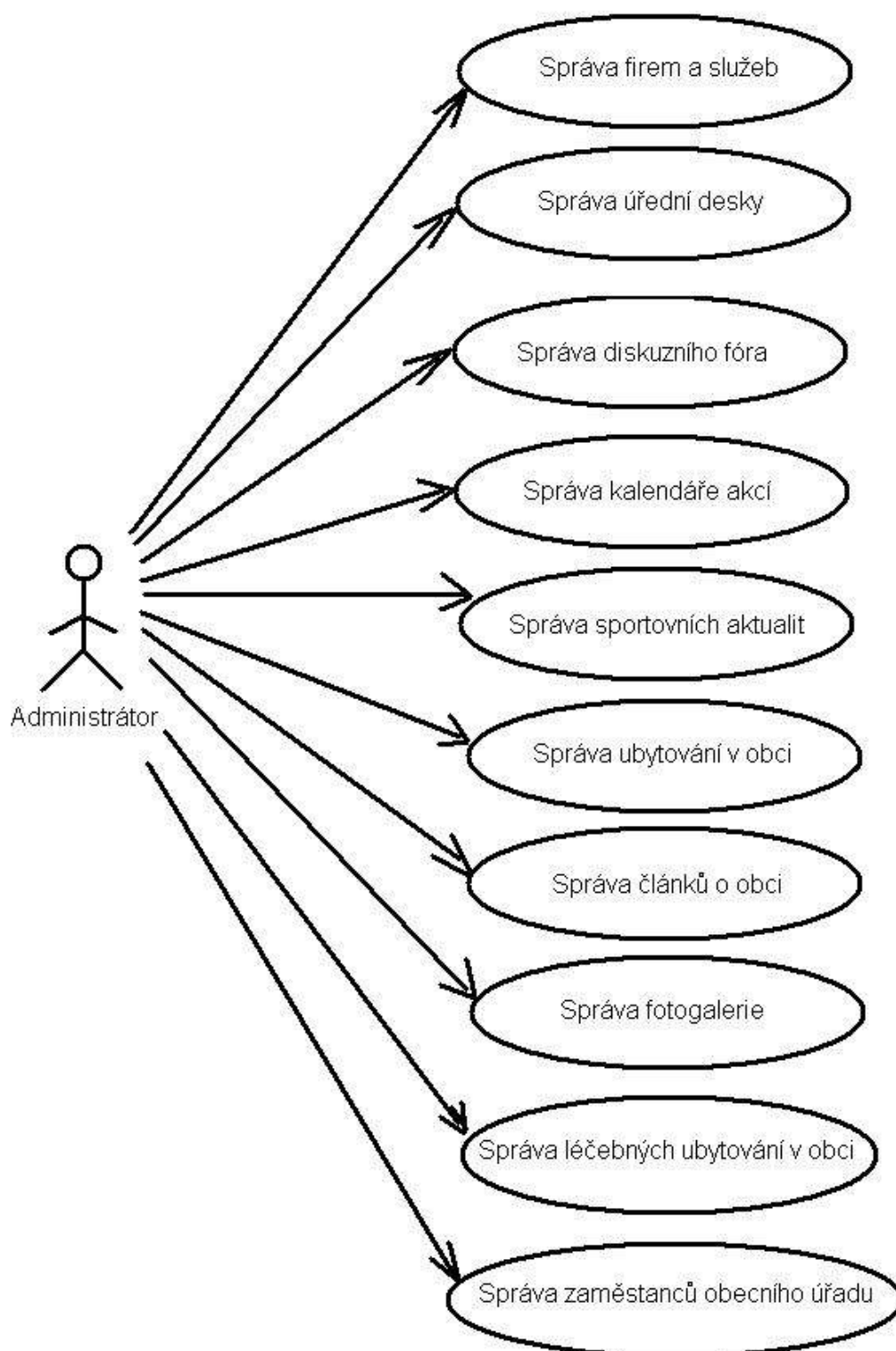
$sql= "SET names 'cp1250'";

mysql_query($sql);
```

## Příloha č. 2. ER – diagram



### Příloha č. 3. Digram případů použití – administrátor



## Příloha č. 4. Digram případů použití - návštěvníci

