

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Informační systém pro mateřskou školu

Tomáš Bříza

Bakalářská práce

2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš BŘÍZA**  
Osobní číslo: **I08003**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Informační systém pro mateřskou školu**  
Zadávací katedra: **Katedra informačních technologií**

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

V teoretické části bude popsána problematika ochrany osobních dat na základě platné legislativy v souvislosti s řešeným projektem.

Praktická část se bude zabývat tvorbou internetové aplikace určené pro komunikaci mezi rodiči dětí a učitelkami v mateřské škole.

Aplikace bude mít dvě části - veřejnou (informativní) a interní (určenou pro učitelky a rodiče dětí), data budou uložena v databázi.

Veřejná část bude obsahovat aktuality, kalendář plánovaných akcí, jídelníček na aktuální týden, fotogalerii proběhlých akcí, informace pro rodiče (adaptace dětí, atd.) a informace o školce (kontakt, foto, atd.).

Neveřejná část bude určena pracovníkům MŠ pro evidenci seznamu dětí (jméno, příjmení, datum narození, pohlaví, adresa, kontakty na rodiče a osoby, které mohou dítě vyzvednout) a jejich příslušnost k jednotlivým třídám, docházku a typ docházky (dopoledne, celý den, 4hod) a individuální hodnocení dětí (v pěti okruzích). Na základě těchto dat bude možné generovat statistiky docházky, hodnocení a vývoje dětí.

Do neveřejné části bude umožněn po přihlášení přístup i rodičům dětí pro sledování docházky, omlouvání dětí či sledování hodnocení vlastních dětí. Důležitým aspektem je návrh vhodného fyzického datového modelu.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**\*Castagnetto, J. a kol. Programujeme PHP profesionálně. Computer Press, 2004.**

**\*Kout, P. Praktický JavaScript. Zoner Press, 2004.**

**\*Riordan, R.M. Vytváříme relační databázové aplikace. Computer Press, 2001.**

**\*Ullman, L. PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. Computer Press, 2004.**

Vedoucí bakalářské práce:

**RNDr. David Žák, Ph.D.**

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2010**

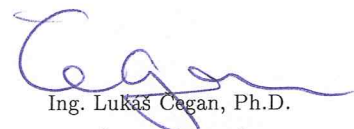
Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2010

## **Prohlášení autora**

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 28. 04. 2010

Tomáš Bříza

## **Poděkování**

Rád bych touto cestou poděkoval vedoucímu mé práce RNDr. Davidu Žákovi, PH.D za jeho věcné připomínky, rady a konzultace při tvorbě této práce.

## **Anotace**

V teoretické části se zabývám problematikou ochrany osobních údajů v informačních systémech. V praktické části popisuji vývoj aplikace pro usnadnění komunikace mezi pracovníci mateřské školy a rodiči dětí.

## **Klíčová slova**

WWW, informační systém mateřské školy, databáze, PHP, ochrana osobních údajů

## **Title**

Kindergarten information system.

## **Annotation**

In this theoretical part i will be occupied with problem about protection of personal information in information systems. In practical part, there will be written about development of application for easy communication between staff of the kindergarten and children parents.

## **Keywords**

WWW, Kindergarten information system, database, PHP, protection of personal information

<b>Seznam zkratků.....</b>	<b>6</b>
<b>Seznam obrázků.....</b>	<b>7</b>
<b>1 Úvod.....</b>	<b>11</b>
<b>2 Ochrana osobních údajů v informačních systémech.....</b>	<b>12</b>
2.1 Problém ochrany osobních údajů .....	12
2.2 Vymezení základních pojmů .....	13
2.3 Osobní údaj.....	14
2.4 Zpracování osobních údajů.....	15
2.5 Rodné číslo .....	16
2.6 Povinnosti správce.....	17
2.6.1 Oprávnění ke zpracování osobních údajů.....	18
2.6.2 Informování subjektů.....	19
2.7 Jak se dovolat nápravy? .....	20
2.8 Shrnutí a závěr .....	20
2.9 Aplikace v praktické části .....	21
<b>3 Požadavky .....</b>	<b>22</b>
3.1 Uživatelé systému.....	22
3.2 USE-Case diagram .....	24
<b>4 Použité technologie .....</b>	<b>27</b>
4.1 HTML.....	27
4.1.1 Trocha historie.....	27
4.1.2 Koncepce .....	27
4.1.3 Struktura stránky .....	28
4.2 CSS.....	28
4.2.1 Syntaxe .....	28
4.2.2 Selektory.....	29
4.2.3 Připojení kaskádových stylů do HTML stránky.....	29
4.3 PHP.....	29
4.4 MySQL .....	30
<b>5 Vývoj aplikace.....</b>	<b>31</b>
5.1 Struktura aplikace.....	31
5.2 Vzhled aplikace .....	31

5.3 Adresářová struktura .....	32
5.4 Databázové schéma .....	33
5.4.1 Popis tabulek .....	34
5.5 Přidání dítěte do systému.....	38
5.6 Seznamy dětí .....	39
5.7 Statistiky docházky.....	42
5.8 Zadávání hodnocení dítěte.....	44
<b>6 Zabezpečení aplikace.....</b>	<b>46</b>
6.1 Role uživatelů .....	46
6.2 Hesla v databázi.....	47
6.3 Formuláře .....	47
6.3.1 Regulární výrazy .....	47
<b>7 Závěr.....</b>	<b>48</b>
<b>8 Použité zdroje: .....</b>	<b>49</b>
<b>Příloha A celý E-R diagram.....</b>	<b>50</b>



## Seznam zkratek

MŠ	mateřská škola
Pdf	Portable Document Format
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SGML	Standard Generalized Markup Language
CERN	Evropská organizace pro jaderný výzkum
xHTML	extensible hypertext markup language
WYSIWYG	What you see is what you get
MD5	Message-Digest algorithm

## Seznam obrázků

Obrázek 1 - Use case Host.....	24
Obrázek 2 - Use case rodič .....	25
Obrázek 3 - Use case učitelka.....	26
Obrázek 4 - Vzhled aplikace .....	31
Obrázek 5 - E-R diagram.....	33
Obrázek 6 - formulář pro vložení dítěte do systému .....	38
Obrázek 7 - Seznam dětí.....	39
Obrázek 8 - Statistiky docházky .....	42
Obrázek 9 - Zadávání hodnocení.....	44

# 1 Úvod

V této bakalářské práci se zabývám ochranou osobních údajů, se zaměřením na ochranu osobních údajů na internetu a v informačních systémech. Rozebírám co je osobní údaj, který zákon upravuje ochranu osobních údajů, jaké povinnosti jsou kladeny na správce osobních údajů. Dotknu se problematiky rodných čísel. Naznačím možnosti dovolání se nápravy při zjištění porušení práv na ochranu osobních údajů.

V praktické části rozebírám vývoj informačního systému určeného pro mateřské školy. Popisuji důvody výběru jednotlivých technologií. Je zde kapitola o struktuře aplikace, její vzhled a databázové schéma s popisem tabulek. Vybral jsem některé zajímavé funkcionality a zevrubně je popsal. Zaměřuji se také na ochranu dat v aplikaci, jak jsou uložena hesla v databázi, jakým způsobem se přihlašují uživatelé nebo jak se kontrolují vstupní data z formulářů.

## **2 Ochrana osobních údajů v informačních systémech**

### **2.1 Problém ochrany osobních údajů**

V dnešní době internetových obchodů, různých sociálních sítích, všemožných internetových stránek s různou tematikou, kdy lidé prostřednictvím těchto stránek o sobě píšou leckdy i velmi osobní informace, narůstá potřeba tyto informace chránit. V médiích všech druhů, tištěných, v televizi a samozřejmě i na internetu se poměrně často objevují zprávy o zneužívání těchto informací.

Proto z důvodu chránit tyto osobní informace byl 4. dubna 2000 parlament České republiky schválen zákon č. 101/2000 Sb. [1], o ochraně osobních údajů. Který chrání osoby, před zneužitím osobních i citlivých údajů. Tento zákon nebyl prvním zákonem, který se zabýval ochranou osobních údajů, ale nahrazoval zákon č. 256/1992 Sb., o ochraně osobních údajů v informačních systémech platící od roku 1992, který sice byl ve své době svým způsobem průlomovým, ale určitě nebyl dokonalým. Potýkal se s řadou nepřesných definicí, problémy také byly při domáhání práv poškozených osob, nebo počítal se zřízením orgánu, který by se staral o ochranu osobních údajů jednotlivců, který ale nebyl nikdy zřízen.

Zákon [1], opravil chyby svého předchůdce 256/1992 Sb., zavedl několik dosud neznámých novinek do zpracování osobních údajů v ČR a navíc je kompatibilní se směrnicemi Evropské unie. Také [3] jako jeden z mála zákonů neukládá povinnosti fyzickým osobám, ale všem osobám, které na základě tohoto zákona mohou přijít do styku s osobními údaji, jsou dány obecné povinnosti, aby subjekty údajů neutrpěly žádnou újmu na svých právech, zejména na právu zachování lidské důstojnosti a také aby neoprávněně nezasahovali do soukromí subjektu údajů.

V rámci ČR je zákon 101/2000 Sb. [1] brán jako obecná právní norma určující práva subjektu údajů (viz základní pojmy) a povinnosti správce, případně zpracovatele osobních údajů. Také reguluje přenos dat do zahraničí a vztahy, které v souvislosti s nimi vznikají. Současně však vedle něj existují zvláštní zákony pro jednotlivé speciální případy pro zacházení s osobními údaji.

## 2.2 Vymezení základních pojmů

Pro plné pochopení problematiky ochrany osobních údajů je třeba se seznámit se základními pojmy, jak je definuje zákon [1].

Zákon vymezuje tyto základní pojmy v odstavci §4:

- A. *osobním údajem je jakákoliv informace týkající se určeného nebo určitelného subjektu údajů. Subjekt údajů se považuje za určený nebo určitelný, jestliže lze subjekt údajů přímo či nepřímo identifikovat zejména na základě čísla, kódu nebo jednoho či více prvků, specifických pro jeho fyzickou, fyziologickou, psychickou, ekonomickou, kulturní nebo sociální identitu,*
- B. *citlivý údaj je osobní údaj vypovídající o národnostním, rasovém nebo etnickém původu, politických postojích, členství v odborových organizacích, náboženství a filozofickém přesvědčení, odsouzení za trestný čin, zdravotním stavu a sexuální životě subjektu údajů a genetický údaj subjektu údajů; citlivým údajem je také biometrický údaj, který umožňuje přímou identifikaci nebo autentizaci subjektu údajů,*
- C. *anonymní údaj je takový údaj, který buď v původním tvaru, nebo po provedeném zpracování nelze vztáhnout k určenému nebo určitelnému subjektu údajů,*
- D. *subjekt údajů je fyzická osoba, k níž se osobní údaje vztahují,*
- E. *zpracováním osobních údajů je jakákoliv operace nebo soustava operací, které správce nebo zpracovatel systematicky provádějí s osobními údaji a to automatizovaně nebo jinými prostředky. Zpracováním osobních údajů se rozumí zejména shromažďování, ukládání na nosiče informací, zpřístupňování, úprava nebo pozměňování, vyhledávání, používání, předávání, šíření, zveřejňování, uchovávání, výměna, třídění nebo kombinování, blokování a likvidace,*
- F. *shromažďováním osobních údajů je systematický postup nebo soubor postupů, jehož cílem je získání osobních údajů za účelem jejich dalšího uložení na nosič informací pro jejich okamžité nebo pozdější zpracování,*
- G. *uchováváním osobních údajů je udržování údajů v takové podobě, které je umožňuje dále zpracovávat,*
- H. *blokováním osobních údajů je vytvoření takového stavu, při kterém je osobní údaj určitou dobu nepřístupný a nelze jej jinak zpracovávat,*
- I. *likvidací osobních údajů se rozumí fyzické zničení jejich nosiče, jejich fyzické vymazání nebo jejich trvalé vyloučení z dalších zpracování,*

- J. *správce je každý subjekt, který určuje účel a prostředky zpracování osobních údajů, provádí zpracování a odpovídá za něj. Zpracováním osobních údajů může správce zmocnit nebo pověřit zpracovatele, pokud zvláštní zákon nestanoví jinak,*
- K. *zpracovatelem je každý subjekt, který na základě zvláštního zákona nebo pověření správcem zpracovává osobní údaje podle tohoto zákona,*
- L. *zveřejněným osobním údajem je osobní údaj zpřístupněný zejména hromadnými sdělovacími prostředky, jiným veřejným sdělením nebo jako součást veřejného seznamu,*
- M. *evidencí nebo datovým souborem osobních údajů (dále jen „datový soubor“) je jakýkoliv soubor osobních údajů uspořádaný nebo zpřístupnitelný podle společných nebo zvláštních kritérií,*
- N. *souhlasem subjektu údajů se rozumí svobodný a vědomý projev vůle subjektu údajů, jehož obsahem je svolení subjektu údajů se zpracováním osobních údajů,*
- O. *příjemcem je každý subjekt, kterému jsou osobní údaje zpřístupněny; za příjemce se nepovažuje subjekt, který zpracovává osobní údaje podle § 3 odst. 6 písm. g).*

## 2.3 Osobní údaj

Osobním údajem se rozumí podle zákona [1]:

*„osobním údajem se rozumí jakákoliv informace týkající se určeného nebo určitelného subjektu údajů. Subjekt údajů se považuje za určený nebo určitelný, jestliže lze subjekt údajů přímo či nepřímo identifikovat zejména na základě čísla, kódu nebo jednoho či více prvků, specifických pro jeho fyzickou, fyziologickou, psychickou, ekonomickou, kulturní nebo sociální identitu.“ [1]*

Z [2] toho vyplývá, že za osobní údaj se považuje informace nebo skupina informací, dle kterých lze přímo či nepřímo zjistit identitu subjektu údajů. O osobní údaj se nejedná, pokud je třeba ke zjištění identity subjektu údajů nepřiměřené množství času, úsilí či materiálních prostředků. Za osobní údaje jsou považovány ty, které fyzickou osobu určují nebo mohou určit, a to i nepřímo. Zpravidla se jedná o soubor údajů, může se jednat ale jen o jeden (např. rodné číslo). Proto lze za osobní údaj považovat jméno, příjmení a adresu. Ve většině případů lze subjekt údajů jednoznačně určit pomocí těchto 3 údajů, nebo v případě že na stejné adrese bydlí více osob se stejným jménem a příjmením, můžeme připojit datum narození. Ve zvláštních případech lze subjekt údajů určit i pomocí jiných údajů, jako například „starosta obce Radim k datu 1. 1. 2010“.

Zákon [1] kromě osobního údaje také definuje údaje citlivý:

*„citlivým údajem se rozumí osobní údaj vypovídající o národnostním, rasovém nebo etnickém původu, politických postojích, členství v odborových organizacích, náboženství a filozofickém přesvědčení, odsouzení za trestný čin, zdravotním stavu a sexuálním životě subjektu údajů a genetický údaj subjektu údajů; citlivým údajem je také biometrický údaj, který umožňuje přímou identifikaci nebo autentizaci subjektu údajů.“ [1]*

Což v praxi znamená, že zákon chrání údaje, které vedou k identifikaci subjektu údajů, ale také chrání údaje citlivé. Za citlivé se považují ty údaje, které nějak vypovídají o osobním životu, rasovém nebo etnickém původu, o zdravotním stavu subjektu údajů. Na zpracování citlivých údajů klade vyšší požadavky, než na zpracování údajů osobních.

## **2.4 Zpracování osobních údajů**

Zpracování osobních údajů zákon [1] definuje v několika bodech, zejména §4 e):

*„zpracováním osobních údajů se rozumí jakákoliv operace nebo soustava operací, které správce nebo zpracovatel systematicky provádějí s osobními údaji, a to automatizovaně nebo jinými prostředky. Zpracováním osobních údajů se rozumí zejména shromažďování, ukládání na nosiče informací, zpřístupňování, úprava nebo pozměňování, vyhledávání, používání, předávání, šíření, zveřejňování, uchovávání, výměna, třídění nebo kombinování, blokování a likvidace.“ [1]*

Velice [3] důležité ve znění zákona je *systematičnost činnosti* s osobními údaji, to se odráží v postupech, jejichž cílem je získání osobních údajů za účelem jejich uložení na datové úložiště k jejich okamžitému či pozdějšímu zpracování. Pojem *shromažďování* by mohl vést k dojmům, že o pojmu zpracování osobních údajů lze hovořit, až když zpracováváme větší množství dat, ale zákon nijak neurčuje, od jakého množství dat lze hovořit o zpracovávání osobních údajů.

Dále je pro zpracování dat velice důležitý §4 f):

*„shromažďováním osobních údajů je systematický postup nebo soubor postupů, jehož cílem je získání osobních údajů za účelem jejich dalšího uložení na nosič informací pro jejich okamžitě nebo pozdější zpracování.“ [1]*

O shromažďování lze hovořit již od chvíle, jakmile byl získán první údaj a dále jsou cíleně a systematicky získávány další údaje. Přičemž je nezbytné, že osobní údaje musí být pro správce trvale dosažitelné za účelem jejich zpracování. [3] V opačném případě, by se ve smyslu zákona [1] spíše jednalo o blokování, ne-li již přímo likvidované údaje. Takže údaje uložené do osobních spisů studentů nebo zaměstnanců, skladované někde v kartotéce a nadále nepoužité, svým charakterem zjevně nesplňují podmínky kladené na zpracování osobních údajů dle zákona [1].

## 2.5 Rodné číslo

Nejdříve je potřeba si uvědomit co vlastně je rodné číslo.

Rodné číslo je jednoznačný číselný identifikátor přidělovaný občanům České republiky státem a také státem spravovaným. Lze z něj vyčíst datum narození a pohlaví příslušné osoby. Prvních šest číslic vyjadřuje datum narození (ve formátu yyddmm), přičemž ženy mají k měsíci přičteno 50. Zbytek rodného čísla odlišuje osoby narozené během jednoho dne, v naší republice se odděluje lomítkem. Do 1. ledna 1954 za lomítkem následovali pouze tři číslice, poté byla přidána číslice čtvrtá, která slouží ke kontrole správnosti rodného čísla. Toto číslo se doplní tak, aby celé rodné číslo bylo dělitelné beze zbytku 11 (zda číslo je dělitelné jedenácti lze jednoduše zjistit z paměti, tak když sečteme čísla na lichých pozicích a odečteme od nich součet číslic na sudých pozicích, když nám vyjde jedenáct je číslo dělitelné 11 a tudíž formálně správně. Například rodné číslo 736028/5163 součet číslic na lichých pozicích  $7+6+2+5+6=26$  součet číslic na pravé straně  $3+0+8+1+3=15$ , rozdíl  $26-15=11$  což znamená, že rodné číslo je formálně správně)

Problém rodného čísla je v tom, že je jednoznačné, tudíž stačí k jednoznačné identifikaci osoby pouze toto jedno číslo. Proto rodné číslo je velmi citlivý údaj a obecně se doporučuje nepoužívat je jako jednoznačný identifikátor v ne-státních systémech, a to z celkem prozaického důvodu. Protože, když by došlo k úniku těchto identifikátorů, jakýmkoliv způsobem ať ukradením nebo pouhou ztrátou, mohlo by to vést k jejich snadnému zneužití. Kdyby třeba bankám byly zcizeny tabulky s informacemi o účtech svých klientů a identifikace klienta by byla vedena právě pomocí rodného čísla, případný zneuživatel těchto informací by si mohl snadno dohledat, o koho se přesně jedná, a vytipovat si třeba nejbohatší klienty banky. Ale kdyby zavedli svůj vlastní identifikátor, třeba také nějaký číselný řetězec, byť generovaný pomocí interních pravidel, případným zneuživatelům těchto informací bychom minimálně značně ztížili jednoznačnou identifikaci osoby, protože by tento identifikátor byl jednoznačný pouze v našem systému. A tudíž vůči ostatním systémům zcela nepoužitelný.



## 2.6 Povinnosti správce

Zákon [1] stanovil mnoho povinností pro správce, které musí při zpracování osobních informací brát na vědomí, já se budu zabývat několika nejdůležitějšími a nejkomplicovanějšími.

Hned [3] první bod: *stanovit účel, k němuž mají být osobní údaje zpracovány*. Je v celku dosti komplikovaný, a to proto, že správce vždy nemusí přesně odhadnout, k čemu všemu osobní údaje bude zpracovávat. Takže při špatně stanovenému účelu, který správce musí stanovit zásadně předem, může posléze zjistit, že jím nasbíraná data, nemá právo zpracovat pro některé své potřeby. Proto musí stanovit nový účel, který je spojen se znovu odsouhlasením všech subjektů údajů vedených v systému a je s tím v podstatě spojena stejná práce jako při shromažďování údajů nových. Proto je potřeba stanovit spíše širší účel zpracování a tím předejít těmto komplikacím.

Další [3] komplikovanou věcí: *zpracovávat pouze pravdivé a přesné osobní údaje a současně ověřovat*. Tento problém nevzniká, když data vznikají na straně správce, který si sám ohlíká jejich správnost, nebo případně provede aktualizace. Jediným problémem v tomto případě by mohla být rychlost zadání údajů nebo jejich případná aktualizace plynoucí ze zaneprázdněnosti správce nebo jeho nedostupnosti. Ale hlavní komplikace v tomto bodě vznikají, když data vznikají na straně subjektu údajů, nebo někde mimo kompetence správce. Zde pouze zbývá najít rovnováhu mezi denní kontrolou, zda se údaje nezměnili a ponechání dat beze změny, dokud se neprojeví problémy spojené se zastaralostí dat. Nejideálnější variantou je zavázat subjekt údajů, aby neprodleně hlásil změnu údajů. Ideální v tomto případě je poskytnout subjektu údajů okamžitý přístup, nejlépe on-line, ke všem svým osobním údajům, aby mohl provést případné změny sám.

Další zásadní věcí zmíněnou v zákoně je: *nesdružovat osobní údaje, které byly získány k rozdílným účelům*. Příkladem zde může společnost, která provádí statistické analýzy pro třetí strany, které jim dodávají data. Tato společnost může zpracovávat pouze data v takovém rozsahu, jaký jim dodal zadavatel a není oprávněna sloučit tyto data s daty od zadavatele jiného.

### 2.6.1 Oprávnění ke zpracování osobních údajů

Obecně [3] platí, že správce může zpracovávat pouze osobní údaje se souhlasem subjektu údajů, případně jeho zástupce (je-li v tomto smyslu omezena způsobilost k právním úkonům nebo je-li jich přímo zbaven). Kdyby toto, ale platilo pro všechny případy, znamenalo by to v některých situacích až neřešitelné komplikace, proto je v zákoně několik výjimek:

- a) jestliže provádí zpracování upravené zvláštním zákonem,*
- b) jestliže je nezbytné, aby subjekt údajů mohl vstoupit do jednání o smluvním vztahu nebo aby plnil ujednání smlouvy uzavřené se správcem,*
- c) pokud je to nezbytně třeba k ochraně důležitých zájmů subjektu údajů,*
- d) jedná-li se o oprávněně zveřejněné osobní údaje v souladu se zvláštním právním předpisem,*
- e) pokud je to nezbytné pro ochranu práv správce.*

Mezi uvedené výjimky doplnila novela [4] i další podmínku:

*„f) pokud je to nezbytné pro výkon oprávněné činnosti politických stran, politických hnutí, občanských sdružení, odborových organizací, církví nebo náboženských společností“, protože zde se hned v první fázi aplikace projeví značné potenciální komplikace života těchto organizací. Vylučuje, aby se údaje shromažďovaly pod záminkou jiného účelu nebo jiné činnosti.*

Bez souhlasu subjektu údajů lze osobní údaje zpracovávat také za účelem statistického nebo vědeckého zpracování. Pro tyto účely zpracování, je nutné tyto informace anonymizovat, hned jakmile je to možné. Nicméně pro tyto účely, jsou stanoveny také limity pro zpracování osobních údajů a to ve smyslu, že nesmějí být v rozporu s právem subjektu údajů na jeho soukromého a osobního života.

V původním znění zákona bylo jasně definováno, že souhlas může být výhradně v písemné formě, a to proto, aby z něj bylo jasné v jakém rozsahu, k jakému účelu a na jak dlouho budou osobní údaje zpracovávány. A aby bylo jasné, že subjekt údajů může svůj souhlas kdykoliv odvolat. Ale toto ustanovení v některých případech bylo těžko splnitelné, například v internetových aplikacích by bylo dosti komplikované vyžadovat po uživateli vytisknutí souhlasu, jeho následné podepsání a poslání poštou správci. Proto byla tato povinnost novelou [4] upravena. A to tak, že se obecně nepožaduje písemná forma souhlasu, což umožňuje využívat některé možnosti elektronického potvrzení. S podmínkou, že musí být prokazatelné, například po přihlášení uživatele (subjektu údajů do systému) do systému a zaznamenání tohoto souhlasu.

Zákon určuje [3] zvláštní pravidla pro zpracování citlivých údajů, kde povoluje pouze výslovný souhlas a to pouze písemný. Zároveň v tomto souhlasu musí být jednoznačně určené, jaké údaje subjekt údajů dává, jakému správci, za jakým účelem zpracování a na jakou dobu. Tento písemný souhlas musí být po celou dobu zpracování citlivých údajů uchován a je odvolatelný. Správce je při podpisu tohoto souhlasu, dovolující zpracování citlivých údajů, povinen poučit subjekt údajů o těchto svých právech. Z tohoto výčtu povinností je zřejmé, že je potřeba pořádně si rozmyslet, zda nutně potřebujeme některý údaj, podléhající definici údaje citlivého. Proto se po zavedení tohoto zákona z databází vytratil údaj o národnosti osob, který pro administrativu většiny systémů je naprosto nadbytečný. Problém ale nastal v některých zaměstnaneckých databázích, ve kterých se evidovala způsobilost pro výkon dané práce, proto se ustálila definice citlivého údaje na tom, že zdravotní stav ve smyslu způsobilosti k vykonávání dané práce se nepovažuje za citlivý údaj, ale přesná diagnóza již ano.

### **2.6.2 Informování subjektů**

Zákon [1] v bodech §11 a 12 stanovil také povinnosti správce informovat subjekty údajů. Tyto [3] informace lze rozdělit na dva druhy. První druh jsou základní informace o datech, které o subjektu údajů shromažďujeme (například že o něm shromažďuje údaje, v jakém rozsahu a pro jaký účel, kdo je bude dále zpracovávat a pro jaký účel a komu mohou být zpřístupněny, či komu jsou údaje určeny, poučení o tom, zda je podle zákona [1] povinen pro zpracování osobní údaje poskytnout, jaké důsledky budou vyvozeny, pokud tak neučiní, a kdy je oprávněn odmítnout poskytnutí osobních údajů, nebo zda poskytnutí osobních údajů je dobrovolné, o jeho právu k přístupu k osobním údajům a, jestliže správce nezískal osobní údaje od subjektu údajů, také informace o tom, kdo mu údaje poskytl, informace o druhu osobních údajů, a také obsah těchto údajů) a pak, jako druhý druh, samotné informace. Poslání samotných informací samo osobě nijak komplikované není, je to pouze o vyřízení žádosti, od subjektu údajů, a poslání správně zpracovaného formuláře, ze strany správce či provozovatele. V původním znění zákona byl ale u prvního druhu informací problém, protože správcům a provozovatelům nebylo vždy jasné, co vlastně mají posílat, nebo když jim to jasné bylo, tak jejich posílání považovali za příliš formální nebo dokonce za zbytečné, protože to všechny zúčastněné subjekty údajů věděli. Proto s novelou [4] přišla změna, která upřesnila, že zasílání tohoto druhu informací je nutné pouze v případě, když subjekt údajů tyto informace nezná. Také bylo změněno to, že první druh informací se zasílá pouze na písemnou žádost subjektu údajů a také se tyto informace nemusejí zasílat, když subjekt údajů poskytl souhlas k jejich zpracování. Samozřejmě zde existují výjimky pro zpracování osobních údajů pro statistické, vědecké a archivnické účely.

## 2.7 Jak se dovolat nápravy?

Pokud [2] subjekt údajů zjistí, že došlo k porušení povinností správcem nebo zpracovatelem, má právo obrátit se na Úřad pro ochranu osobních údajů (se sídlem v Praze, Havelkova 22, Praha 3) s žádostí o zajištění opatření k nápravě. Došlo-li k porušení povinností správcem nebo zpracovatelem, má subjekt údajů právo požadovat:

- aby se správce či zpracovatel zdržel takového jednání, odstranil takto vzniklý stav či poskytl na svoje náklady omluvu nebo jiné zadostiučinění,
- aby správce či zpracovatel provedl opravu nebo doplnění osobních údajů tak, aby byly pravdivé a přesné,
- aby osobní údaje byly zablokovány nebo zlikvidovány,
- zaplacení peněžité náhrady, jestliže tím bylo porušeno jeho právo na lidskou důstojnost, osobní čest, dobrou pověst či právo na ochranu jména.

Právo na ochranu má subjekt údajů vůči každému správci, ať jím je orgán veřejné správy či subjekt soukromý. Které z výše uvedených opatření subjekt údajů zvolí, je v jeho dispozici. Nelze vyloučit ani uplatnění všech opatření.

## 2.8 Shrnutí a závěr

Jak je patrné z těchto několika málo stránek, problematika ochrany osobních údajů, je velice rozsáhlá a komplikovaná. Zcela klíčové je, před započítím shromažďování osobních či citlivých údajů, stanovit jaké přesně údaje budeme shromažďovat a zpracovávat. Poté je třeba si uvědomit, zda potřebujeme výslovný písemný souhlas, to v případě, že bude zpracovávat údaje citlivé nebo nám postačí souhlas elektronický, ten by nám měl postačit v případě, kdy budeme zpracovávat pouze údaje osobní. Dále je zapotřebí zajistit, aby zpracovávané údaje byli přístupné pouze pověřeným pracovníkům a zamezit zcizení a jejich případnému následnému zneužití. Dále je potřeba připravit se na možnost, že subjekt údajů bude požadovat informace o osobních údajích, které o něm shromažďujeme či zpracováváme.

## 2.9 Aplikace v praktické části

Do praktické části byly zpracovány některé poznatky získané během vytváření teoretické části o ochraně osobních údajů. Zejména byl zaveden specifický identifikátor, jednoznačný pouze pro danou aplikaci, také bylo upuštěno, po domluvě, od evidování rodného čísla, které pro evidenci docházky a hodnocení není nikterak důležité. Z důvodů zpracovávání a evidence pouze pravdivých údajů, budou všechny nové údaje zadávané do systému přidávat pouze učitelky mateřské školy po přihlášení. Samotní rodiče jsou zavázání k neprodlené úpravě údajů zavedených do systému (kontakt, adresa), jakmile se změní. Rodiče mají přístup ke všem informacím vedených o nich a jejich dětech. Samozřejmostí je, že rodiče se musí do systému přihlásit. Veškeré informace vedené o dětech, jak jejich hodnocení tak docházka, jsou přístupné pouze přihlášeným učitelkám a administrátorovi. Neautorizovaní uživatelé nemají přístup k žádným informacím. Vzhledem k tomu že rodiče při zápisu dětí do školky podepisují souhlas se zpracováním osobních i citlivých údajů, není zapotřebí vytvářet další prohlášení určené pro informační systém.

### 3 Požadavky

Aplikace je určena pro usnadnění komunikace mezi učitelkami mateřské školy a rodiči dětí. Ve veřejné části aplikace jsou dostupné aktuality o dění ve školce, kalendář plánovaných akcí, jednoduchá galerie, jídelníček, informace o školce a dokumenty, které mohou rodiče potřebovat. V interní části je vedený seznam dětí se základními informacemi o nich (jméno, příjmení, datum narození, adresa, informace o jejich rodičích atd.). Aplikace zjednodušuje evidenci docházky a poskytuje jednoduché statistiky. Je vypracován systém hodnocení jednotlivých dětí.

Aplikace nemá ambice plně nahradit stávající agendu mateřské školy, která je velmi rozsáhlá, pouze má za účel zjednodušit práci v některých částech této agendy. Aplikace byla primárně vyvíjena pro mateřskou školu Radim. Kde začátku školního roku 2010/2011 by měla být spuštěna pro testování.

#### 3.1 Uživatelé systému

System rozlišuje několik druhů uživatelů:

- náhodný návštěvník (host),
- rodič,
- učitelka MŠ,
- administrátor.

Možnosti náhodného uživatele:

- prohlédnout základní informace o školce,
- najít si potřebné kontakty na vedení MŠ,
- prohlížet aktuality,
- prohlížet plánované akce,
- galerii,
- jídelníček,
- stáhnout si potřebné dokumenty.

Možnosti rodiče:

- přístup k údajům o svých dětech,
- prohlížení docházky,
- prohlížení hodnocení.

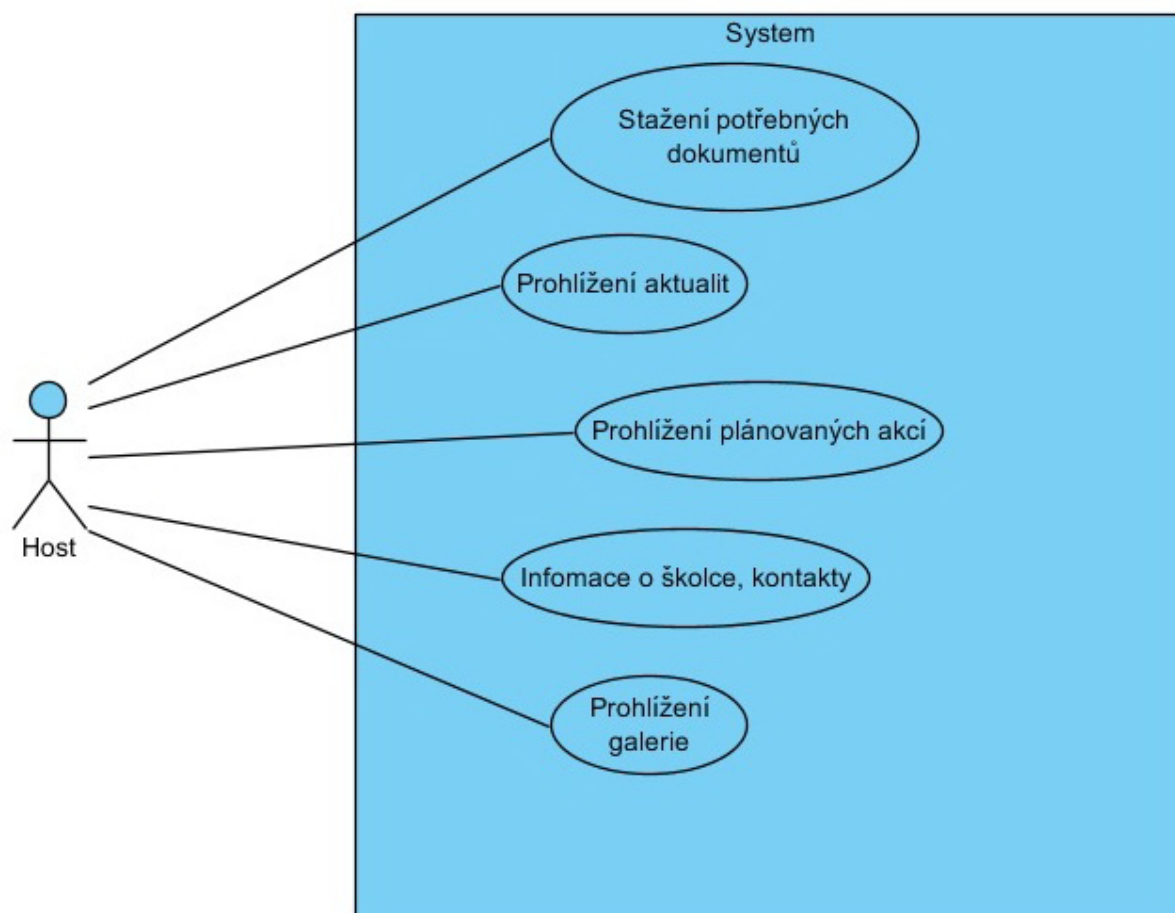
Možnosti učitelky MŠ:

- přidávat děti do evidence,
- vyřazovat děti z evidence,
- prohlížet seznamy dětí ve školce, tříděné pomocí různých parametrů (v současné době lze řadit podle abecedy, data narození a třídit děti lze také podle zařazení do tříd) a generovat z nich soubory pdf,
- měnit přiřazení dítěte do třídy,
- vedení docházky, kdy pracovnice MŠ vidí omluvené děti,
- zobrazení statistik docházky,
- vedení hodnocení jednotlivých dětí. Měnit okruhy hodnocení, varianty hodnocení podle aktuálních potřeb, přičemž starší hodnocení zůstávají beze změny,
- zobrazovat si všechna hodnocení dítěte a tím zjistit jeho pokrok,
- přidávat aktuality o dění ve školce,
- přidávat plánované akce,
- upravovat jídelníček.

Samotný administrátor má stejné možnosti jako učitelky MŠ a navíc má možnost přidávat nové pracovnice mateřské školy.

## 3.2 USE-Case diagram

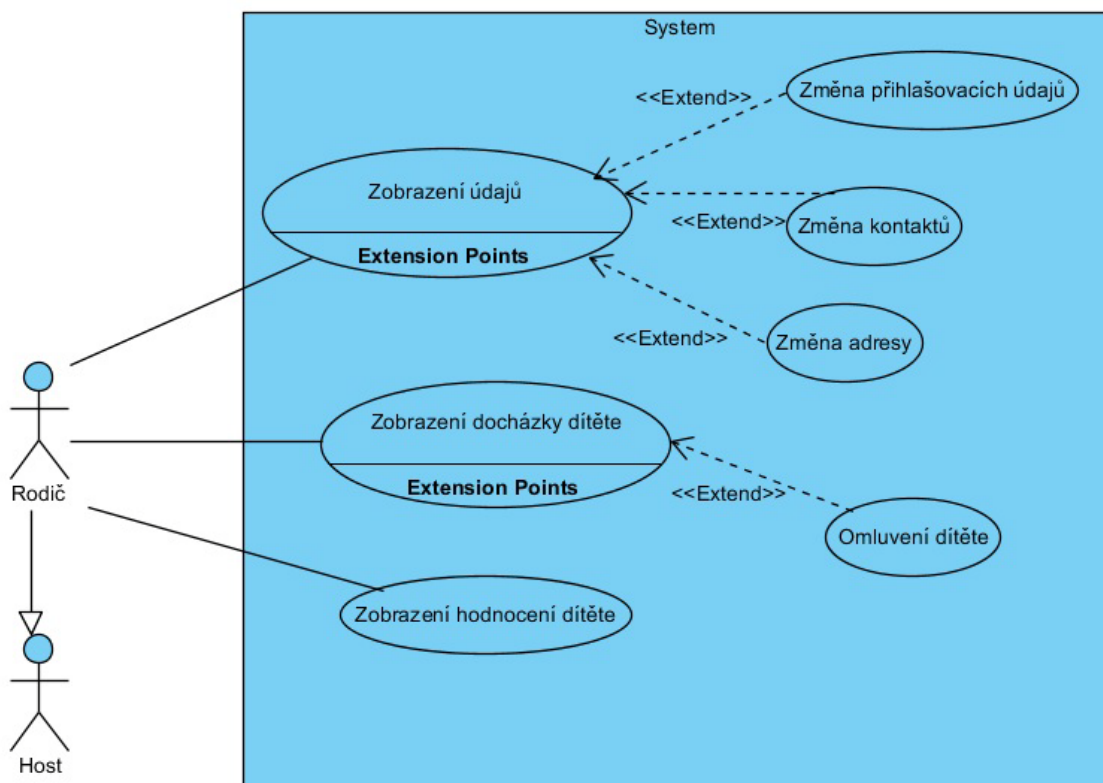
Pro větší přehlednost jsem vytvořil USE-Case diagramy pro všechny uživatele systému:



Obrázek 1 - Use case Host

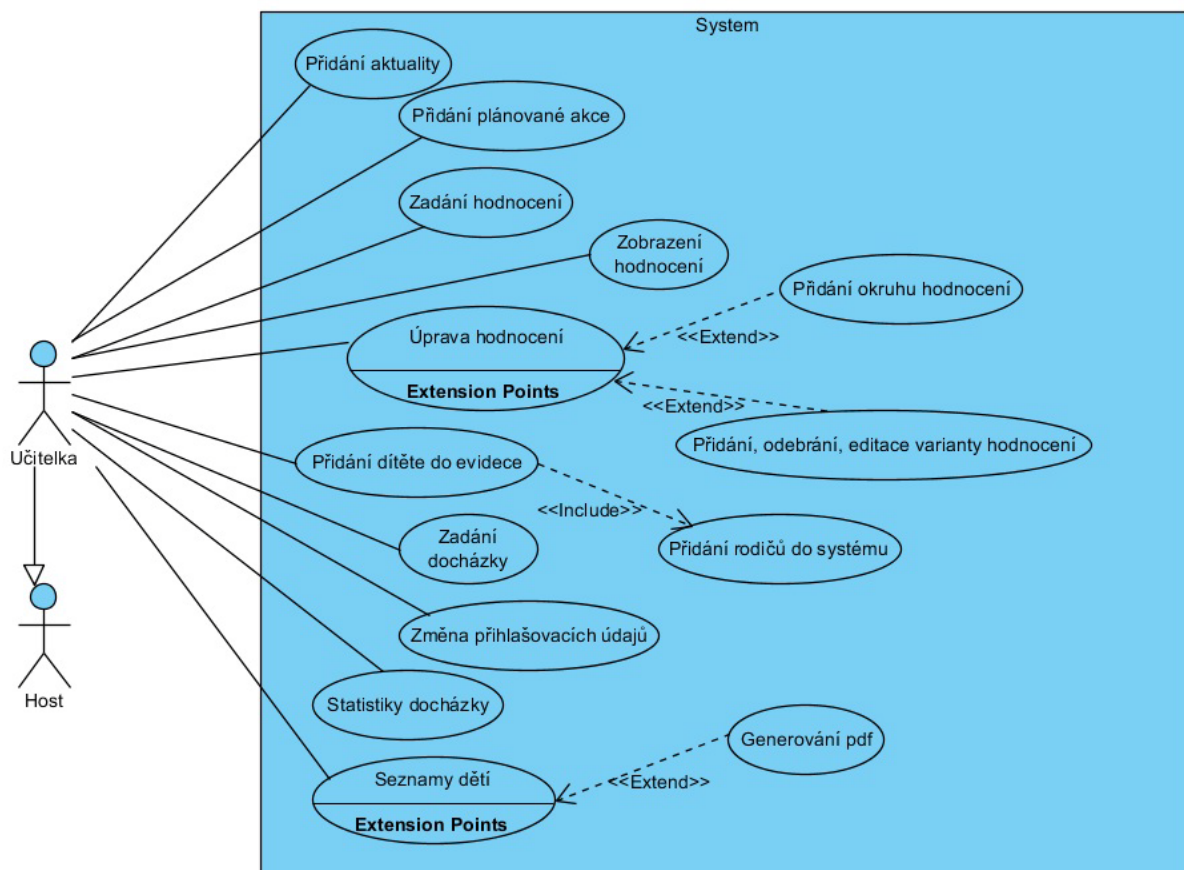
Zde jsou vidět jednotlivé možnosti, co může náhodný návštěvník v systému dělat. Takže si může prohlížet fotky v galerii, může si najít základní informace o MŠ a kontakty na vedení. Také se může podívat co je aktuálně nového ve školce (aktuality), nebo co má školka v budoucnu v plánu (plánované akce). Nic jiného mu nebude bez přihlášení do systému povoleno.





**Obrázek 2 - Use case rodič**

Rodič má stejné možnosti jako host. Po přihlášení do systému má další možnosti. A to zobrazení jeho osobních údajů, které o něm školka vede. Samozřejmě má možnost si změnit své přihlašovací údaje (přihlašovací jméno a heslo). Také má možnost změnit kontaktní údaje, email nebo telefon. V případě přestěhování může také zaktualizovat adresu, na kterou se přestěhoval. Může také sledovat docházku svého dítěte, ale není mu umožněno prohlížet si docházku dětí jiných rodičů. Může si také zobrazit, jak a kdy bylo jeho dítě hodnoceno, samozřejmě nemůže sledovat hodnocení jiných dětí.



**Obrázek 3 - Use case učitelka**

Učitelka má samozřejmě stejné možnosti jako náhodný návštěvník, ale také má mnoho dalších variant. Může si změnit přihlašovací údaje do systému, přidávat aktuality na hlavní stránku, plánované akce. Může upravovat kritéria hodnocení dětí, může přidávat okruhy hodnocení. Přidávat, editovat a rušit varianty hodnocení v okruzích hodnocení. Může také přidávat děti do evidence, to je spojené s přidáním rodičů dítěte do systému. Učitelka má možnost zadávat docházku a prohlížet její statistiku. V aplikaci jsou také vedeny seznamy dětí, které lze třídit podle několika kritérií (datum narození, abecedně) a roztřídit podle přiřazených tříd, je také umožněno generovat tyto seznamy jako dokumenty pdf.

Administrátor má stejné možnosti jako učitelka, s tím že má možnost přidávat další pracovnice mateřské školy.

## 4 Použité technologie

Vzhledem k co nejjednoduššímu přístupu rodičů do aplikace jsem se rozhodl použít HTML a vystavit stránky na internetu. Ke stylování stránek jsem použil CSS stylování, databázi jsem vybral MySQL, protože je na naprosté většině hostingů, a je velice dobře zdokumentována. Pro dynamičnost samotných stránek jsem zvolil PHP, a to z podobných důvodů jako databázi MySQL. Vybrané technologie jsou na drtivé většině hostingů, a je okolo nich velice početná komunita se spoustou návodů a tutoriálů.

### 4.1 HTML

#### 4.1.1 Trocha historie

HTML [6] neboli HyperText Markup Language, je značkovací jazyk, nikoliv programovací jak si lidé často myslí. HTML byl navržen za účelem vytvoření jednoduššího jazyka pro tvorbu elektronických dokumentů, než do té doby používané TeX<sup>1</sup>, PostScript<sup>2</sup> nebo SGML<sup>3</sup>. V roce 1990 byl představen spolu s protokolem HTTP a prvním prohlížečem v CERNU<sup>4</sup>.

#### 4.1.2 Koncepce

HTML dokumenty jsou složeny z takzvaných elementů, to znamená z počáteční značky (uzavřené do závorek <>), samotné hodnoty elementu a ukončovací značky (uzavřené </>).

##### Například:

```
<b>Ahoj</b>
```

Tento příklad znamená, že text Ahoj (hodnota elementu) bude zobrazen tučně, značka <b> je počáteční tag elementu a značka </b> je koncový tag. Jednotlivé elementy mohou obsahovat další, ale nesmějí se navzájem křížit.

##### Dobře:

```
<p><b>Nějaký text</b></p>
```

##### Spatně:

```
p><b>Nějaký text</p></b>
```

---

<sup>1</sup> Sázeční systém ze 70. let 20. století. Naprogramovaný profesorem Donaldem Knuthem

<sup>2</sup> Programovací jazyk určený ke grafickému popisu tisknutelných dokumentů, vyvinutý v roce 1985 firmou Adobe Systems

<sup>3</sup> SGML (z ang. *Standard Generalized Markup Language*) univerzální značkovací jazyk

<sup>4</sup> CERN (z franc. *Conseil Européen pour la recherche nucléaire*), česky Evropská organizace pro jaderný výzkum.

### 4.1.3 Struktura stránky

HTML dokument má přesně nadefinovanou strukturu:

- Deklarace DTD<sup>5</sup>,
- Kořenový element html. Je povinný, ale při jejich absenci si je prohlížeč umí, doplnit sám,
- Hlavička, element head. Obsahuje informace týkající se celého dokumentu (název dokumentu, autora, jazyk dokumentu, styly zobrazení),
- Tělo dokumentu, element body. Obsahuje už samotný obsah dokumentu.

**Příklad:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
  <!-- toto je komentář -->
  <head>
    <title>Titulek stránky</title>
  </head>
  <!-- tělo dokumentu -->
  <body>
    <h1>Nadpis stránky</h1>
    <p>Toto je tělo dokumentu</p>
  </body>
</html>
```

## 4.2 CSS

CSS [7] (z ang. Cascading Style Sheets) neboli kaskádové styly. Je to jazyk pro popis vzhledů dokumentů napsaných v HTML (případně xHTML<sup>6</sup> nebo XML<sup>7</sup>). V podstatě byly navrženy pro oddělení datové části dokumentu od části popisující jeho vzhled.

### 4.2.1 Syntaxe

Syntaxe se skládá ze selektoru a bloku deklarácí, kde blok deklarácí obsahuje vlastnost poté dvojtečku a hodnotu vlastnosti a ukončovací středník.

**Příklad:**

```
body {
  background-color: white;
  color: black;
  padding: 10px !important;
}
```

<sup>5</sup> DTD (z ang. *Document Type Definition*) definice typu dokumentu

<sup>6</sup> xHTML (z ang. *extensible hypertext markup language*) rozšiřitelný hypertextový značkovací jazyk

<sup>7</sup> XML (z ang. *Extensible Markup Language*) rozšířený značkovací jazyk

## 4.2.2 Selektory

V CSS existuje mnoho druhů selektorů:

- `body` – selektor pro všechny výskyty elementu `body`,
- `body p` – selektor pro všechny výskyty elementu `p`, nacházející v elementu `body`,
- `.trida` – selektor pro všechny elementy, které mají pomocí atributu `class` nastavenou třídu `trida`,
- `#id` – selektor pro všechny elementy, které mají pomocí atributu `id` nastavený `id` na `id`,
- `sel1, sel2, sel2` – takzvané seskupení selektorů, kdy následná deklarace bude platit pro všechny.

## 4.2.3 Připojení kaskádových stylů do HTML stránky

Pro připojení CSS stylů do samotného dokumentu existují 3 možnosti:

- takzvaný přímý zápis, kdy samotné vlastnosti vzhledu zapíšeme přímo do atributu `style`,
- zápis stylů do elementu `style`, tyto styly pak budou dostupné v celé stránce,
- připojení externího souboru, asi nejčastější způsob.

## 4.3 PHP

PHP [8] (z ang. Hypertext Preprocessor) neboli, Hypertextový preprocesor je skriptovací jazyk pro tvorbu skriptů, prováděných na straně serveru. Umožňuje, aby stránky, byly dynamické, zpracovávají se pomocí nich data z formulářů, připojuje s pomocí nich databáze a pracuje se s ní a ještě spousty dalších věcí.

**Příklad syntaxe:**

```
<?php
    echo "Ahoj, světe!";
?>
```

## 4.4 MySQL

MySQL [9] je databázový systém, vytvořený firmou MySQL AB, tato firma je ze Švédska. Což dává odpověď, proč některé věci jsou defaultně nastaveny na švédštinu. MySQL databáze je velice oblíbená především protože je zadarmo, a je používána především programátory webových stránek. Také je velice snadno instalovatelná, jak pod windows, tak pod linux, za zmínku stojí především balíček LAMP<sup>8</sup> pro linux.

MySQL databáze byla vytvářena s ohledem především na rychlost zpracování požadavků, proto zde do nedávné minulosti chyběli například pohledy, triggerů nebo uložené procedury.

---

<sup>8</sup> LAMP (z *Linux*, *Apache*, *MySQL*, *PHP*), balíček pro instalaci potřebného svobodného softwaru pro tvorbu dynamických stránek pod linuxem

## 5 Vývoj aplikace

### 5.1 Struktura aplikace

Pro samotnou aplikaci jsem zvolil vývoj takzvaný sestavování aplikace z kousků. Neboli jsou vytvořeny funkce pro vložení hlavičky, patičky a menu uživatele. Pro každou stránku je samostatný soubor \*.php ve kterém je napsána pouze jeho funkcionalita a jsou připojeny funkce pro hlavičku, patičku a menu.

### 5.2 Vzhled aplikace



Obrázek 4 - Vzhled aplikace

Na předchozím obrázku je vidět vzhled aplikace (její úvodní obrazovku), v horní části je logo aplikace, pod ním je hlavní menu, nyní se nacházíme na záložce „O Školce“, dále je možnost si vybrat mezi dalšími různými záložkami (Dokumenty, Plánované akce, Fotoalbum, Okénko pro rodiče, Kontakty). Pod tímto menu se záložkami se nachází, další odkazy již spojené se záložkou, na které se aktuálně nacházíme, v našem případě jsou zde odkazy „O Školce“, který nás zavede na základní charakteristiku školky, a jídelníček, který, jak sám název napovídá, nás zavede na jídelníček. Poté následuje samotný obsah

záložky z hlavního menu, u nás jsou zobrazeny aktuality o dění ve školce. A nakonec v pravé části se nachází menu uživatele, kde je nyní formulář pro přihlášení.

### 5.3 Adresářová struktura

Aplikace je logicky roztríděná do adresářů, které shromažďují podobnou funkcionalitu. V kořenovém adresáři jsou položky hlavního menu, s výjimkou položky galerie, která je specifická a je ve vlastním adresáři, a to:

- index.php
- ProRodice.php
- PlanovaneAkce.php
- Kontakty.php
- Dokumenty.php

Jednotlivé adresáře:

- AktualityAkce – o tomto adresáři jsou soubory \*.php které zajišťují výpis aktualit a plánovaných akcí a jejich úpravu.
- Deti – zde je soubor, který generuje seznamy dětí a zajišťuje generování pdf těchto seznamů.
- Dochazka – tady je skrytá funkcionalita zajišťující zadávání docházky do systému, omlouvání dětí a generování statistik docházky.
- Dokumety – zde je soubor s informacemi o školce, a jednotlivé dokumenty ke stažení, které jsou k dispozici z různých míst stránek.
- Edit – zde je ukryta funkcionalita, která mění různé údaje, například změna přihlašovacích údajů, změna adresy a změna kontaktů.
- Hodnoceni – tato složka slučuje vše, co se točí okolo hodnocení, ať jeho zadávání, úpravu nebo prohlížení.
- Jidelnicek – zde je vše potřebné pro zobrazení a úpravě jídelníčku.
- Skripty – vše co se nevešlo jinde, jsou zde přístupy do databáze, skripty pro ověření uživatelů, autorizace uživatelů nebo menu pro uživatele.
- Tridy – vše co se třídí. Vytváření tříd, seznamy tříd a podobně.
- Uzivatele – zde jsou skripty pro přidávání uživatelů do systému, zobrazení jejich profilových informací a další nezbytné náležitosti.
- Galerie – všechny soubory týkající se galerie
- Images – soubory s obrázky použitých na stránkách, nikoliv však s obrázky z galerie.
- Tnymce – wysiwyg editor<sup>9</sup>, určený pro pohodlnější zadávání aktualit a plánovaných akcí

---

<sup>9</sup> Wysiywg edit (akronym z „*What you see is what you get*“), označuje způsob editace dokumentů, při které je verze zobrazená na obrazovce vzhledově totožná s výslednou verzí dokumentu.





### 5.4.1 Popis tabulek

Zde se budu věnovat popisu jednotlivých tabulek a jejich atributů.

#### Lide

Základní tabulka, zde jsou společné informace o osobách zavedených v systému a zároveň je jim zde přidělen jednoznačný identifikátor *id*.

- *Id* (**Int**), primární klíč.
- *Jmeno* (**Varchar**).
- *Prijmeni* (**Varchar**).
- *Datum\_narozeni* (**Varchar**).
- *Mesto* (**Varchar**).
- *Ulice* (**Varchar**).
- *CP* (**Int**).
- *PSC* (**Int**).

#### Uzivatele

Zde jsou všechny potřebné informace pro přihlášení uživatelů. Uživatelé se zde identifikují pomocí cizího primárního klíče *Id\_lide* z tabulky Lide.

- *Id\_lide* (**Int**), primární cizí klíč z tabulky Lide.
- *Nick* (**Varchar**), přihlašovací jména.
- *Heslo* (**Varchar**), heslo sloužící k přihlášení.
- *Role* (**Varchar**), role uživatele v systému.
- *Platnost\_od* (**Date**), od kdy se může uživatel do systému přihlásit.
- *Platnost\_do* (**Date**), platnost do kdy se uživatel může do systému přihlásit.

#### Rodice

Seznam všech rodičů v systému, identifikují se pomocí primárního cizího klíče *Id\_lide* z tabulky Lide.

- *Id\_lide* (**Int**), primární cizí klíč z tabulky Lide.
- *Telefon* (**Int(9)**), telefon, omezená délka na 9 číslic.
- *Email* (**Varchar**).

#### Deti

Tabulka všech dětí, identifikují se pomocí primárního cizího klíče *Id\_lide* z tabulky Lide.

- *Id\_lide* (**Int**), primární cizí klíč z tabulky Lide.
- *Datum\_reg* (**Date**), datum zadání dítěte do systému.

## Rodice\_a\_deti

Vazební tabulka, která obsahuje 3 cizí klíče, a to *Id\_rodic*, *Id\_dite* a *Id\_vztah*. Kde *Id\_rodic* a *Id\_dite* jsou primárními klíči. Tato tabulka má za účel „spojit“ rodiče se svými dětmi, nebo případně osoby oprávněné k vyzvednutí dětí

- *Id\_rodic* (**Int**), primární cizí klíč z tabulky Rodice.
- *Id\_dite* (**Int**), primární cizí klíč z tabulky Deti.
- *Id\_vztah* (**Int**), cizí klíč z tabulky Vztah.

## Vztah

Číselník vztahů

- *Id* (**Int**), primární klíč.
- *Nazev* (**Varchar**), název vztahu.

## Reg\_do\_tridy

Tabulka, ve které je dětem přiřazovaná třída, kde jednoznačným identifikátorem registrace je sloupeček *Id*.

- *Id* (**Int**), primární klíč.
- *Od* (**Date**), od kdy je dané dítě registrované do určité třídy.
- *Do* (**Date**), od kdy je dané dítě registrované do určité třídy.
- *Id\_dite* (**Int**), cizí klíč z tabulky Deti.
- *Id\_tridy* (**Int**), cizí klíč z tabulky Tridy.

## Tridy

Číselník tříd, kde *Id* je jednoznačná identifikace třídy, *Nazev* je pojmenování třídy a *Id\_ucitele* je cizí klíč z tabulky Uzivatele, který určuje třídní učitelku.

- *Id* (**Int**), primární klíč.
- *Nazev* (**Varchar**).
- *Id\_ucitele* (**Int**), cizí klíč za tabulky Uzivatele.

## Dochazka

V této tabulce se eviduje veškerá docházka dětí do školky, sloupeček *Datum*, je primární klíč, určuje datum kdy docházky, *Id\_dite* určuje jaké dítě a *Id\_typDoch*, cizí klíč z tabulky *Typ\_dochazky* určuje, jak dítě bylo ve školce.

- *Datum* (**Date**), primární klíč.
- *Id\_dite* (**Int**), primární cizí klíč z tabulky Deti.
- *Id\_typDoch* (**Int**), cizí klíč z tabulky *Typ\_dochazky*.

## Typ\_dochazky

Číselník typu docházek

- *Id* (**Int**), primární klíč.
- *Nazev* (**Varchar**), název docházky (například dopoledne nebo celý den).

## Omluvenky

Tabulka všech omluvenek, kde *Id* je jednoznačný identifikátor, *Id\_dite* označuje dítě, které je omluveno, *Id\_omlouval* označuje uživatele, který dítě omluvil. *Od* a *Do* označuje období na kdy je dítě omluveno a *Duvod* popisuje důvod omluvy.

- *Id* (**Int**), primární klíč.
- *Id\_dite* (**Int**), cizí klíč z tabulky Deti.
- *Id\_omlouval* (**Int**), cizí klíč z tabulky Uzivatele.
- *Od* (**Date**).
- *Do* (**Date**).
- *Duvod* (**Varchar**).

## Hlavicka\_hodnoceni

V této tabulce jsou uvedeny základní informace ke každému hodnocení. *Id* jednoznačný identifikátor hodnocení v rámci systému, *Id\_dite* označuje hodnocené dítě, *Id\_hodnotil* označuje hodnotitele, *K\_datu* označuje, k jakému datu se hodnocení vztahuje, *Datum\_hodnoceni* je skutečné datum zápisu hodnocení

- *Id* (**Int**), primární klíč.
- *Id\_dite* (**Int**), cizí klíč z tabulky Deti.
- *Id\_hodnotil* (**Int**), cizí klíč z tabulky Uzivatele.
- *K\_datu* (**Date**).
- *Datum\_hodnoceni* (**Date**).

## Hodnoceni

Vazební tabulka, ve které se uchovává veškeré hodnocení dětí v rámci systému. *Id\_hlavicky* označuje, k jaké hlavičce se záznam vztahuje, *Id\_variatny* označuje, co hodnotíme a *Id\_stupne* označuje, jak hodnotíme

- *Id\_hlavicky* (**Int**), primární cizí klíč z tabulky Hlavicka\_hodnoceni.
- *Id\_variatny* (**Int**), primární cizí klíč z tabulky Varianty\_hodnoceni.
- *Id\_stupne* (**Int**), cizí klíč z tabulky Stupne\_hodnoceni.

## Stupne hodnoceni

Číselník stupňů hodnocení, kde *Id* je jednoznačný identifikátor, *Nazev* je název hodnocení a *Stupen* je stupeň hodnocení

- *Id* (Int), primární klíč.
- *Nazev* (Varchar).
- *Stupen* (Int).

## Variaty\_hodoceni

Číselník variant hodnocení, kde *Id* je jednoznačný identifikátor, *Nazev* je název hodnocení, *platnost\_Od* a *platnost\_Do* označuje dobu platnosti hodnocení a *Id\_okruhHod* označuje, jakému okruhu varianta hodnocení náleží.

## Okruh\_hodnoceni

Číselník okruhů hodnocení, kde *Id* je jednoznačný identifikátor a *Nazev* je název okruhu.

- *Id* (Int), primární cizí klíč.
- *Nazev* (Varchar).

## Planovane\_akce

Tato tabulka obsahuje seznam plánovaných akcí, kde *Id* je jednoznačný identifikátor akce, *Datum\_konani* obsahuje informaci o datu konání dané akce, *Text* je samotné sdělení, *Nadpis* je nadpis akce a *Id\_vlozil* obsahuje informaci, kdo panovanou akci přidal.

- *Id* (Int), primární klíč.
- *Datum\_konani* (Date).
- *Text* (Varchar).
- *Nadpis* (Varchar).
- *Id\_vlozil* (Int), cizí klíč z tabulky Uzivatele.

## Aktuality

Tato tabulka obsahuje všechny aktuality školky, kde *Id* je jednoznačný identifikátor aktuality, *Datum\_vlozeni* obsahuje informaci o datu vložená aktuality, *Text* je samotné sdělení, *Nadpis* je nadpis aktuality a *Id\_vlozil* obsahuje informaci kdo panovanou akci přidal.

- *Id* (Int), primární klíč.
- *Datum\_vlozeni* (Date).
- *Text* (Varchar).
- *Nadpis* (Varchar).
- *Id\_vlozil* (Int), cizí klíč z tabulky Uzivatele.

## 5.5 Přidání dítěte do systému

Formulář pro vkládání dítěte do systému je jeden ze základních stavebních kamenů systému. Pro vložení dítěte je určena podmínka že dítě musí mít alespoň jednoho rodiče bydlící na stejné adrese jako dítě.

Všechna políčka ve formuláři jsou povinná, vyjma položek druhého rodiče po odškrtnutí checkbox Druhý rodič. Před vložení dat do systému, se zkontrolují všechna povinná políčka formuláře, jestli odpovídají potřebnému formátu. Až poté když všechna políčka odpovídají, je dítě spolu s jeho rodičem (rodiči) vloženo do systému. V případě že už některá osoba je v systému zavedena (uživatel je veden v systému, když odpovídá jméno, příjmení, adresa a datum narození zadané ve formuláři s daty v systému), nekládá se znovu, například když nové dítě má rodiče, kteří už figurují v systému je vloženo pouze nový záznam pro dítě a jako rodiče se použijí už existující záznamy ze systému. Ihned po vložení do systému se rodič může přihlásit, pod jeho defaultně nastavenými přihlašovacími údaji (přihlašovací jméno je příjmení rodiče, heslo je jméno). Dítě je okamžitě zaregistrováno do vybrané třídy a je možno mu ihned zadat docházku.

[Zpět na seznam](#)

Údaje o dítěti		Adresa	
Jméno:	<input type="text"/>	Ulice:	<input type="text"/>
Příjmení:	<input type="text"/>	Číslo popisné:	<input type="text"/>
Datum narození:	<input type="text"/>	Město:	<input type="text"/>
		PSČ:	<input type="text"/>
1. Rodič			
Jméno:	<input type="text"/>	Příjmení:	<input type="text"/>
Telefon:	<input type="text"/>	Datum narození:	<input type="text"/>
Email:	<input type="text"/>		
<input checked="" type="checkbox"/> Druhý rodič			
2. Rodič			
Jméno:	<input type="text"/>	Příjmení:	<input type="text"/>
Telefon:	<input type="text"/>	Datum narození:	<input type="text"/>
Email:	<input type="text"/>		
Klokánci	▼	Registrovat do třídy	<input type="button" value="Pridej dítě"/>

**Menu uživatele**  
Vítejte: masarka  
**Dochazka**  
**Vlož aktualitu**  
**Vlož plánovanou akci**  
**Vlož učitele**  
**Hodnoceni**  
**Seznam tříd**  
**Seznam dětí**  
**Uprav jídelníček**  
**Profil**  
**Odhlášení**

Obrázek 6 - formulář pro vložení dítěte do systému

## 5.6 Seznamy dětí

Na následujícím obrázku je zobrazen vygenerovaný seznam dětí roztříděný podle tříd a se zobrazenými podrobnostmi.

**Přidání dítěte do evidence**

Volby zobrazení

- Seřadit abecedně
- Seřadit podle data narození
- Rozdělit podle tříd
- Podrobnosti
- Generovat jako pdf

Zobraz

**Menu uživatele**

Vítejte: masarka

- Docházka
- Vlož aktualitu
- Vlož plánovanou akci
- Vlož učitele
- Hodnocení
- Seznam tříd
- Seznam dětí
- Uprav jídelníček
- Profil
- Odhlášení

**Seznam dětí k datu: 05.05.2010**

**Klokánci**

	Jméno	Datum narození	Adresa
1	Váňa Jiří	01.05.2007	Sportovní 531 Račim 28103
2	Vaničková Ilona	02.02.2007	Na Skále 2511 Račim 28103
3	Vnuk Luboš	13.12.2006	Nad 72 Chotutice 28105

**Veverky**

	Jméno	Datum narození	Adresa
1	Vaniček Tomáš	24.04.2007	Na Skále 2511 Račim 28103
2	Petrová Veronika	21.04.2006	Průběžná 52 Chotutice 28105

Obrázek 7 - Seznam dětí

Nerozepisuji zde celou funkcionalitu generování seznamu dětí, ale chtěl bych se zaměřit pouze na jeden případ, a to seznam dětí roztříděný podle tříd a vygenerovaný jako pdf. O samotné generování pdf dokumentu se stará volně šiřitelná třída mpdf, která pro vytvoření dokumenty pdf potřebuje pouze proměnou ve které je uložena stránka v html kódu.

Funkce pro vygenerování jakékoliv stránky jako pdf:

```
function tiskniPdf($html) {

    include("../mpdf/mpdf.php");
    $mpdf=new mPDF('utf-8');
    $stylesheet = file_get_contents('../print.css');
    $mpdf->WriteHTML($stylesheet,1);
    $mpdf->useLang = true;
    //$mpdf->debug = true;
    $mpdf->WriteHTML($html, 2);
    $mpdf->Output();
    exit;

}
```

Jak je zde vidět tato třída také podporuje css stylování pdf dokumentů, a nemá jako jedna z mála problémy s českými znaky.

Problematiku generování pdf tedy máme úspěšně zvládnutou, teď se podíváme na samotné generování seznamu dětí. O to se stará tato funkce:

```
function VypisPodleTridyPdf($html) {

    $tridy = dotaz('SELECT * FROM `tridy`');
    while($radek = mysql_fetch_array($tridy)) {
        $id_tridy = $radek["Id"];
        $html = $html.'<H2><a
href=" ../Tridy/SeznamTrid.php">'. $radek["Nazev"]. '</a></H2>';
        if(isset($_POST["SeradDatumNarozeni"])) {
            $html = vypispdf($html, "dotaz");
        }else {
            $html = vypispdf($html, "dotaz");
        }
    }
    return $html;
}
```

U této ukázky zdrojového kódu jsem byl nucen mírně upravit zdrojový kód samotné funkce a samotné selecty nahradit pouze slovem dotaz, protože tyto selecty jsou velice rozsáhlé a kód by jenom zneřehlednili, pro samotné pochopení funkcionality však nejsou potřebné.

Samotná funkce je velice jednoduchá, v cyklu projíždí všechny třídy ve školce, do proměnné \$html ukládá jejich názvy a volá další funkci, která má za úkol vpisovat do proměnné \$html samotný seznam dětí. Funkce vypispdf se volá s různými selecty do databáze a to podle toho zda si uživatele přeje ještě děti seřadit podle data narození.



Funkce vypisPdf pouze provede dotaz předaný do ní druhým parametrem a vepíše do proměnné \$html tabulku s dětmi, pokud si uživatele přál výpis podrobností, v této fázi vývoje pouze adresu, vepíší se i podrobnosti. Pro úplnost přikládám celou funkci vypispdf:

```
function vypispdf($html, $dotaz) {

    $dети = dotaz($dotaz);
    $i = 1;
    $html = $html.'<table id="Tabulka">';
    $html = $html.'<tr><th></th><th>Jméno</th><th>Datum narození</th>';

    if(isset($_POST["Podrobnosti"])) {
        $html = $html.'<th>Adresa</th></tr>';
    }else $html = $html.'</tr>';
    while($radek = mysql_fetch_array($dети)) {
        $datumNarozeni = strtotime($radek["Datum_narozeni"]);
        $datumNarozeni = strftime("%d.%m.%Y", $datumNarozeni);
        $html = $html.'<tr><td>'. $i.'</td><td>';
        $html = $html.'<a
href="../Uzivatele/InformaceOditeti.php?dite_id='.$radek["id"].'">'. $radek["prijmeni"]." ". $radek["jmeno"]."</a>";
        $html = $html.'</td><td>'. $datumNarozeni.'</td>';
        if(isset($_POST["Podrobnosti"])) {
            $html = $html.'<td>'. $radek["ulice"].' '. $radek["cp"].'
'. $radek["mesto"].' '. $radek["psc"].'</td>';
        }else {
            $html = $html.'</tr>';
        }
        $i++;
    }
    $html = $html.'</table>';
    return $html;
}
```

Po provedení těchto funkcí se zavolá funkce, tiskni pdf s parametrem \$html a dokument ve formátu pdf se zobrazí na obrazovce.

## 5.7 Statistiky docházky

Statistiky docházky je jedna z nejdůležitějších částí aplikace, lze zobrazovat statistiky docházky celé školky, tak jednotlivých tříd. Jak lze vidět z následujícího obrázku, docházku lze zobrazit mezi jakýmikoliv daty. Při výčtu jednotlivých dat jsou vynechány víkendy.

V tabulce zleva je vidět název dne, datum, počet dětí ve školce (po kliknutí na číslo, se zobrazí jmenný seznam dětí s jejich účastí ve školce daný den) a poté jsou zobrazeny počty dětí v jednotlivých variantách docházky.

Zadat docházku    **Statistiky jednotlivých tříd**

Zobrazit docházku mezi daty

Od data 20.03.2010    K datu 06.04.2010    Zobrazit

Den	Datum	Počet dětí ve školce	Dopoledne	Celý den	4 hodiny
úterý	6.04.2010	3	2	0	1
pondělí	5.04.2010	4	1	2	1
pátek	2.04.2010	4	2	1	1
čtvrtek	1.04.2010	3	1	1	1
středa	31.03.2010	3	1	2	0
úterý	30.03.2010	2	1	1	0
pondělí	29.03.2010	3	3	0	0
pátek	26.03.2010	0	0	0	0
čtvrtek	25.03.2010	2	1	0	1
středa	24.03.2010	2	0	2	0
úterý	23.03.2010	2	0	2	0
pondělí	22.03.2010	2	2	0	0

Průměrná hodnota za období: 2.5

**Menu uživatele**

Vítejte: masarka

- Docházka
- Vlož aktualitu
- Vlož plánovanou akci
- Vlož učitele
- Hodnocení
- Seznam tříd
- Seznam dětí
- Uprav jídelníček
- Profil
- Odhlasení

Obrázek 8 - Statistiky docházky

Pro názornost zde uvedu některé zajímavé funkce. Pro rozlišení názvu dne v týdnu jsem vytvořil jednoduchou funkci `cesky_den`, pro kterou je jediným parametrem pořadový den v týdnu, a funkce vrátí český název dne.

```
function cesky_den($den) {
    static $nazvy = array('neděle', 'pondělí', 'úterý', 'středa',
        'čtvrtek', 'pátek', 'sobota');
    return $nazvy[$den];
}
```

Zajímavý problém ještě vzniknul při určování od kdy, do kdy se statistiky vlastně mají zobrazit.

```
$Od = strtotime($Od);
$Do = strtotime($Do);

$pocetDni = ($Do - $Od)/60/60/24;
$odData = floor((time() - $Do)/60/60/24);

$Od = StrFTime("%d.%m.%Y", $Od);
$Do = StrFTime("%d.%m.%Y", $Do);
```

První dva řádky převedou textový řetězec na datum ve vteřinách od roku 1970. Proměnná \$pocetDni udává kolik dní je mezi daty a proměnná \$odData udává o kolik dní zpět datum uložený v proměnné \$Do, je vzdálený od dnešního data.

Proměnná \$odData je velice důležitá pro samotné vypisování datumů v cyklu, a používá se pro určení, který datum se vlastně použije.

```
$dnes = mktime(0, 0, 0, date("m"), date("d")-($i+$odData), date("Y"));
```

Tento řádek vyrobí z dnešního data, datum, které potřebuje, kde \$i je počítáč cyklů, ve kterém právě jsme a proměnná \$odData udává, jak daleko od dnešního data máme skončit. Například když dnes bude 21. 4. 2010, budeme ve třetím opakování cyklu (\$i = 2) a proměnná \$odData bude mít hodnotu 3 v proměnné \$dnes bude datum 16. 4. 2010.

## 5.8 Zadávání hodnocení dítěte

Formulář pro zadávání hodnocení dítěte, je celý generovaný dynamicky, podle toho zda varianty hodnocení mají k aktuálnímu datu platnost nebo ne. Samotné varianty jsou rozříděny podle příslušnosti do jednotlivých okruhů. Každou variantu hodnocení lze hodnotit v rozmezí 1 až 5, tedy stejně jako ve škole, 1 výborně a 5 nedostatečně. Data z formuláře se odesílají pomocí metody GET a jednotlivé varianty se identifikují pomocí jejich id. Na následujícím obrázku lze vidět vygenerovaný formulář (generován 26. 4. 2010) pro zapsání hodnocení.

[Zpět na seznam](#)

**Tomáš Vaniček**  
Hodnoceni k datu

Jméno hodnocení	1	2	3	4	5
<b>Dítě a jeho tělo</b>					
zachovává správné držení těla	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
správné dýchání	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
pohybuje se koordinovaně	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
házi a chytá míč	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
napodobuje pohyb podle vzoru	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
zvládá stolování, oblékání, obouvání	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
udržuje pořádek	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Jazyk a řeč</b>					
pojmenovává většinu věcí a dějů v okolí	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
správně vyslovuje	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
mluví gramaticky správně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

**Menu uživatele**  
Vítejte: masarka

- Docházka
- Vlož aktualitu
- Vlož plánovanou akci
- Vlož učitele
- Hodnocení
- Seznam tříd
- Seznam dětí
- Uprav jídelníček
- Profil
- Odhlášení

Obrázek 9 - Zadávání hodnocení

Při zpracování těchto dat se znovu provede kontrola, zda id varianty předané metodou GET z formuláře je skutečně id varianty, která je aktuálně platná. Poté se již zavolají příslušné funkce pro vložení hlavičky hodnocení a variant hodnocení.

Ukázka zdrojového kódu pro vložení hodnocení do databáze:

```
if($_GET["datum"] != '') {
    if(JeDatum($_GET["datum"])) {

        $varianty = dotaz("Select varianty_hodnoceni.id,
varianty_hodnoceni.Nazev from varianty_hodnoceni where CURDATE() BETWEEN
varianty_hodnoceni.Platnost_od AND varianty_hodnoceni.Platnost_do");
        $datum = $_GET["datum"];
        $idHlavicky = vložHlavickuHodnoceni($Id_dite, $datum);
        while($radek = mysql_fetch_array($varianty)) {
            $idSel = $radek["id"];
            if(isset ($_GET[$idSel])) {
                vložHodnoceni($idHlavicky, $idSel,
$_GET[$idSel]);
            }else {
                echo '<p id="chyba">Chyba!! Špatné zadané
hodnocení</p>';
                break;
            }
        }
    }else {
        echo '<p id="chyba">Špatný formát data</p>';
    }
}else {
    echo '<p id="chyba">Chyba špatné zadané datum</p>';
}
}
```

Funkce zjistí, zda je nastavené datum a zda má správný formát, poté se z databáze vyberou aktuálně platné varianty hodnocení a v cyklu se všechny projdou a z metody GET se vybere správný stupeň hodnocení a vloží se do databáze.

Zdrojový kód funkce vložHlavickuHodnoceni:

```
function vložHlavickuHodnoceni($idDite, $kDatu) {
    return insert("INSERT INTO `skolka`.`hlavicka_hodnoceni` (`Id`,
`Id_dite`, `Id_hodnotil`, `K_datu`, `Datum_hodnoceni`) VALUES (NULL,
'".$idDite."', '$_SESSION[id].', STR_TO_DATE('".$kDatu."',
'%d.%m.%Y'), CURDATE());");
}
```

Zdrojový kód funkce vložHodnoceni:

```
function vložHodnoceni($idHlavicky, $idVar, $stupen) {
    insert("INSERT INTO `skolka`.`hodnoceni` (`Id_hlavicky`,
`Id_varianty`, `Id_stupne`) VALUES ('".$idHlavicky."', '".$idVar."',
'".$stupen."');");
}
```

Obě předchozí funkce jsou pouze jednoduché inserty do databáze, a byly vytvořeny pouze za účelem zpřehlednění kódu.

## 6 Zabezpečení aplikace

Z pohledu bezpečnosti byly na aplikaci kladeny tyto nároky:

- Neautorizovaný uživatel se nedostane k neveřejnému obsahu,
- Rodiče si mohou prohlížet pouze informace o svých dětech.

Pro identifikaci uživatelů jsem využil session, která se nastaví po úspěšné autorizaci uživatele, a jsou v ní uloženy základní informace o přihlášeném uživateli jeho Id a jeho role v systému.

### 6.1 Role uživatelů

V systému jsou v tento moment vedeny 3 role:

- Rodič,
- Učitel,
- Admin.

Tyto role jsou určeny, k rozpoznávání kam, přihlášený uživatel má přístup. Proto jsem si napsal několik funkcí, které za mne určí, zda má uživatel do dané sekce stránek přístup nebo ne, zde je příklad jedné z nich:

```
function pristupUcitel()
{
    $vysl = false;
    if(isset ($_SESSION['role']))
    {
        if($_SESSION['role'] == 'Admin' || $_SESSION['role'] == 'Ucitel')
        {
            $vysl = true;
        }
    }
    return $vysl;
}
```

Tato jednoduchá funkce zkoumá, zda je session zavedena, v případě že ano zda přihlášený uživatel splňuje požadavky na přístup, v tomto případě zda má roli v systému Admin nebo Ucitel. Pokud ano, funkce vrátí true, pokud ne false.

## 6.2 Hesla v databázi

Pro zabezpečení přihlašovacích údajů jsem zvolil hašovací funkci MD5<sup>10</sup>, která vytvoří z hesla otisk (hash<sup>11</sup>) dlouhý 128 bitů, ze kterého už zpětně nelze heslo dostat zpět. Proto při ověřování správnosti zadaného hesla se toto heslo musí funkcí MD5 zahašovat a poté až porovnat hash ze zadaného hesla tak hash uložený v databázi.

## 6.3 Formuláře

Zpracování a ověření správnosti dat předávaných pomocí formulářů je další důležitou problematikou všech formulářových aplikací. Moje práce toho není výjimkou. Před každým zpracováním dat předaných z formulářů ověřuji, zda byla všechna políčka vyplněná a zda odpovídají formátu, který požadují. Pokud některá z požadovaných informací není buď vyplněná, nebo má špatný formát, na obrazovce se objeví chybové hlášení a formulář se znovu zobrazí. Pro kontrolu správného formátu informací čtených z formulářů jsem použil takzvané regulární výrazy.

### 6.3.1 Regulární výrazy

Regulární výraz (regular expression), je speciální řetězec znaků, který představuje určitý vzor (masku) pro textové řetězce. Regulární výrazy se proto nejčastěji používají ke kontrole dat zadávaných ve formulářích (například e-mailová adresa či PSČ).

Pro představu přikládám jednu z funkcí kontrolující emailovou adresu:

```
function JeEmail ($email) {  
    return preg_match( "^.+@.+\\.\\.+$$", $email );  
}
```

Kde funkce `preg_match` je právě php funkce pro kontrolu regulárních výrazů, kde prvním parametrem funkce je samotný regulární výraz a parametrem druhým je textový řetězec, který chceme zkontrolovat. Tato funkce vrátí `true`, když formát řetězce odpovídá regulárnímu výrazu nebo `false` když neodpovídá.

---

<sup>10</sup> MD5 - Message-Digest algorithm, hašovací algoritmus vytvořený v roce 1991

<sup>11</sup> Hash – výstup z hašovacího algoritmu, též označovaný jako otisk, výtah nebo miniatura

## 7 Závěr

Cílem bakalářské práce bylo vytvořit internetovou aplikaci usnadňující komunikaci mezi rodiči dětí a MŠ, což se povedlo a aplikace je „života schopná“. Všechny potřebné funkce jsou funkční a byly hrubě otestovány (programátorem). Aplikace je připravena k nasazení na webový server k testovacímu provozu. Na aplikaci je stále velmi mnoho práce, ať už dodělat rozbalující se menu uživatele a tím zpřehlednit celou aplikaci, nebo upravit galerii, aby byla více intuitivní a podporovala zobrazení ve složkách.

Během testování se určitě objeví chyby či nedodělky, které mě jako autoru celého programu unikly. Proto aplikace není určitě ve finální podobě a jsem připraven přizpůsobit ji požadavkům uživatelů. Aplikace byla vyvíjena ve Firefoxu, následně byla optimalizována pro Internet Explorer.

Závěrem bych chtěl říct, že aplikace je rozsáhlá a některá funkčnost komplikovaná. Se samotnou aplikací jsem spokojen a práce na ní mi přinesla mnoho potřebných zkušeností, které mi budou užitečné. Jsem velice zvědav, co přinese její spuštění do reálného provozu.



## 8 Použité zdroje:

- [1] *Zákon č. 101/2000 Sb.*[online]. 2000 [cit. 2010-04-12]. Dostupné z www: <<http://www.zakonycr.cz/seznamy/101-2000-Sb-zakon-o-ochrane-osobnich-udaju-a-o-zmene-nekterych-zakonu.html>>.
- [2] Mgr. Ivana Štěpánková. *Ochrana osobních údajů.*[online]. 2000 [cit. 2010-04-12] Dostupné z WWW: <[http://osz.cmkos.cz/CZ/Z\\_tisku/Bulletin/11\\_2000/members\\_ochrana.html](http://osz.cmkos.cz/CZ/Z_tisku/Bulletin/11_2000/members_ochrana.html)>.
- [3] Vladimír Šmíd *Ochrana osobních údajů a informační systémy - dva roky v nových podmínkách.* [online]. 2002? [cit. 2010-04-13]. Dostupné z www <<http://www.fi.muni.cz/~smid/ts02od.html>>.
- [4] *Zákon č. 177/2001 Sb., kterým se mění zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně některých zákonů, ve znění zákona č. 227/2000 Sb., a zákon č. 65/1965 Sb., zákoník práce, ve znění pozdějších předpisů.*[online]. 2001 [cit 2010-14-12]. Dostupné z www <<http://www.gov.cz/wps/portal/s.155/6966/s.155/701?!=177/2001%20Sb.>>.
- [5] *HyperText markup language.* [online]. Wikipedia. Naposledy editováno 23. 3. 2010 [cit. 2010-04-20]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/HyperText\\_Markup\\_Language](http://cs.wikipedia.org/wiki/HyperText_Markup_Language)>.
- [6] *Document Type Definition.* [online]. Wikipedia. Naposledy editováno 5. 4. 2010 [cit. 2010-04-20]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Document\\_Type\\_Definition](http://cs.wikipedia.org/wiki/Document_Type_Definition)>.
- [7] *Kaskádové styly.* [online]. Wikipedia. Naposledy editováno 15. 4. 2010 [cit 2010-04-20]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/wiki/Cascading_Style_Sheets)>.
- [8] *PHP.* [online]. Wikipedia. Naposledy editováno 12. 4. 2010 [cit 2010-04-20]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/PHP>>.
- [9] *MySQL.* [online]. Wikipedia. Naposledy editováno 7. 4. 2010 [cit 2010-04-20]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [10] *Message-Digest algorithm.* [online]. Wikipedia. Naposledy editováno 7. 1. 2010 [cit 2010-04-23]. Dostupné z WWW: <[http://cs.wikipedia.org/wiki/Message-Digest\\_algorithm](http://cs.wikipedia.org/wiki/Message-Digest_algorithm)>.
- [11] Miroslav Pecka. *Základy regulárních výrazů.* [online]. 2005? [cit 2010-04-23]. Dostupné z WWW: <<http://www.regularnivyrazy.info/regularni-vyrazy-zaklady.html>>.

