

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

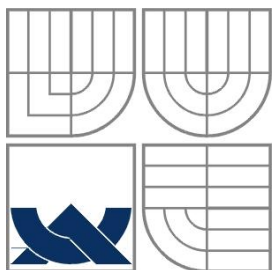
INFORMAČNÍ SYSTÉM PRO VOŠ SOCIÁLNĚ PRÁVNÍ

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

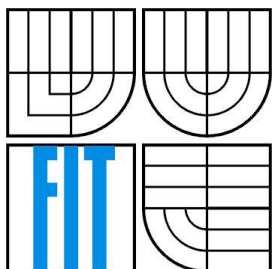
AUTOR PRÁCE  
AUTHOR

ONDŘEJ SVOBODA

BRNO 2009



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 PRO VOŠ SOCIÁLNĚ PRÁVNÍ

INFORMATION SYSTEM FOR SCHOOL OF HIGHER EDUCATION OF SOCIAL WORK AND LAW

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

ONDŘEJ SVOBODA

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. PETR ČÁSTEK

BRNO 2009

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav informačních systémů

Akademický rok 2008/2009

## Zadání bakalářské práce

Řešitel: **Svoboda Ondřej**

Obor: Informační technologie

Téma: **Informační systém pro VOŠ sociálně právní**

Kategorie: Web

Pokyny:

1. Seznamte se s problematikou správy informací ve školách. Prostudujte moderní přístupy a technologie pro vývoj webových aplikací (softwarové architektury, návrhové vzory, programovací jazyky). Podrobně se seznamte s technologií AJAX.
2. Provedte analýzu ve VOŠ sociálně právní.
3. Zaměřte se především řešení registrací, přehledu a hodnocení vyuky. Dále veškeré základní služby poskytované informačními systémy školy.
4. Navrhněte a vytvořte informační systém.
5. Provedte zhodnocení vlastností a případné vylepšení informačního systému společně se zúčastněnými stranami.

Literatura:

- Dokumentace k AJAX dostupná na [www.w3schools.com/ajax/default.asp](http://www.w3schools.com/ajax/default.asp).
- Crane D. a kol.: Ajax in Action. Manning Publications 2005.

Při obhajobě semestrální části projektu je požadováno:

- Zpráva o stavu analytické a přípravné části řešení bakalářské práce.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Částek Petr, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNIČKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2



---

doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

## **Abstrakt**

Tato práce se zabývá návrhem a tvorbou informačního systému pro Vyšší odbornou školu, který bude sloužit jako prostředník mezi studenty a učiteli a usnadní tak jejich komunikaci. Výsledná aplikace je napsána ve skriptovacím jazyce PHP a využívá databázi MySQL, je zde použita i technologie AJAX. Při tvorbě bylo využito paradigmatu objektově orientovaného programování.

## **Abstract**

This bachelor's thesis describes designing and creation of information system for Higher school of education, which will improve communication between students and lectors. Final application is written in PHP scripting language and uses MySQL database and AJAX technology. During implementation object-oriented programming paradigm was used.

## **Klíčová slova**

Informační systém, VOŠ, PHP, MySQL, AJAX, JavaScript, CSS, LDAP, OOP

## **Keywords**

Information system, school, PHP, MySQL, AJAX, JavaScript, CSS, LDAP, OOP

## **Citace**

Svoboda Ondřej: Informační systém pro VOŠ sociálně právní, bakalářská práce, Brno, FIT VUT v Brně, 2009

# Informační systém pro VOŠ sociálně právní

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Částka.

Další informace o provozních záležitostech školy mi poskytla Mgr. Hana Svobodová.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Ondřej Svoboda  
19. května 2009

## Poděkování

Děkuji panu Ing. Petru Částkovi za vedení a podnětné připomínky k mojí práci. Stejně tak patří moje poděkování Evangelické akademii, Vyšší odborné škole sociálně právní, zvláště pak paní ředitelce Mgr. Haně Svobodové za podrobné informace o chodu školy, konzultace a připomínky k mojí práci.

V neposlední řadě bych chtěl poděkovat přítelkyni a rodině za podporu a trpělivost.

© Ondřej Svoboda, 2009

*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

|  |    |
|--|----|
| Obsah.....   | 1  |
| 1 Úvod.....  | 3  |
| 2 Použité technologie a jazyky.....                  | 4  |
| 2.1 HTML.....  | 4  |
| 2.1.1 Vývoj jazyka HTML.....                         | 4  |
| 2.1.2 Popis jazyka.....                              | 5  |
| 2.2 CSS.....   | 6  |
| 2.3 JavaScript.....                                  | 7  |
| 2.4 PHP.....   | 8  |
| 2.4.1 Vývoj jazyka PHP.....                          | 9  |
| 2.5 MySQL.....                                       | 11 |
| 2.6 AJAX.....  | 12 |
| 2.7 LDAP.....  | 13 |
| 3 Návrh systému.....                                 | 14 |
| 3.1 Specifikace a analýza požadavků.....             | 14 |
| 3.1.1 Typy uživatelů.....                            | 15 |
| 3.2 E-R diagram.....                                 | 18 |
| 3.3 Schéma databáze.....                             | 19 |
| 3.4 Uživatelské rozhraní.....                        | 20 |
| 4 Implementace.....                                  | 21 |
| 4.1 Struktura aplikace.....                          | 21 |
| 4.1.1 Objektově orientovaný přístup.....             | 21 |
| 4.2 Použití AJAXu.....                               | 22 |
| 4.2.1 Našeptávač uživatelů.....                      | 23 |
| 4.2.2 Mazání souborů.....                            | 23 |
| 4.3 Předměty.....                                    | 23 |
| 4.4 Správa uživatelů.....                            | 24 |
| 4.4.1 Autentizace uživatelů pomocí LDAP serveru..... | 25 |
| 4.4.2 Ověření funkčnosti elektronické adresy.....    | 25 |
| 4.5 Přejít na nový školní rok.....                   | 25 |
| 4.6 Použité externí knihovny a třídy.....            | 26 |
| 5 Plánovaná rozšíření.....                           | 27 |
| 5.1 Komunikace uživatelů.....                        | 27 |
| 5.2 Tiskové sestavy.....                             | 27 |

|     |                               |    |
|-----|-------------------------------|----|
| 5.3 | Hodnocení výuky a školy ..... | 27 |
| 6   | Závěr .....                   | 29 |
|     | Literatura .....              | 30 |
|     | Seznam příloh .....           | 31 |

# 1 Úvod

Rozvoj počítačů, informačních technologií a hlavně internetu nabral v posledních letech neskutečné obrátky. Nezřídka se tak stává, že máme pocit, jakoby stroje začínali náš život pomalu „řídít“. Ve valné většině případů je to pro nás velmi přínosné a šetří to tak náš čas i energii. Stejná motivace vedla i mne k myšlence tvorby informačního systému pro Evangelickou Akademii, Vyšší odbornou školu sociálně právní, Opletalova 6, Brno.

Moje spolupráce s touto institucí trvá na poli informačních technologií už několik let, a jak jsem víc a více pronikal do způsobu fungování školy respektive způsobu práce některých učitelů, tak jsem čím dál častěji uvažoval o vytvoření informačního systému pro tuto školu. Systému, který by měl hlavně usnadnit komunikaci jednak studentů a učitelů, ale také vytvořit jednotný informační kanál mezi vedením a zaměstnanci.

Předem bych měl uvést, že se jedná o malou školu, kde studuje necelých 100 studentů. Ale i tak se požadavky na systém a nápady na nejrůznější rozšíření rozrostly do celkem velkých rozměrů, které přesahují rámec této bakalářské práce. Budou tedy přidávány postupně při budoucím vývoji.

Tuto technickou zprávu jsem rozčlenil do čtyř okruhů, z nichž každý se týká vždy jedné fáze vývoje systému.

V následující kapitole čtenáře seznámím s technologiemi, které jsem se rozhodnul použít při vývoji mé aplikace. Jsou to vždy volně dostupné a všeobecně velmi oblíbené a používané jazyky a systémy pro tvorbu dynamických webových aplikací. U každého najdete krátkou charakteristiku spojenou většinou s ukázkou kódu nebo použití. Trochu více zde rozebírám nejvýznamnější dva jazyky – HTML a PHP, což jsou hlavní stavební prvky mého informačního systému.

Třetí kapitola je už směřována konkrétně k mému řešení. Věnuje se hned té první fázi života informačního systému, kterou je fáze přípravná a návrhářská. Dozvíte se zde jaké požadavky a co od zavedení informačního systému očekávají na VOŠ. Specifikuji zde hlavní funkce a moduly celé aplikace a nechybí ani návrh struktury databáze.

Implementaci klíčových a v některých ohledech i zajímavých bodů programu shrnuji v následující čtvrté kapitole. Od této části práce nečekejte dokumentaci kódu, nýbrž vysvětlení jen těch základních principů fungování mnou navrženého systému. Hned první podkapitola se zabývá využitím jedné z posledních novinek na poli webových aplikací, kterou je AJAX, který v současné době prožívá opravdu velký rozvoj.

Ne všechny funkce a drobná vylepšení, které vzešly z rozhovorů s vedením školy, se podařily připravit hned do první verze systému. Další rozvoj je nastíněn v předposlední čtvrté kapitole.

Následuje závěr a výčet použité literatury. Jako přílohu přikládám seznam zkratk, CD se zdrojovými kódy a hodnocení mé práce z pohledu VOŠ.



## 2 Použité technologie a jazyky

I když cílem mojí práce není detailní popis jednotlivých programovacích jazyků a jiných technik, rozhodl jsem se čtenáře v této kapitole seznámit se základními informacemi o technologiích použitých při implementaci informačního systému. Jedná se především o jazyky používané v internetu – HTML, JavaScript, PHP atp. Dále pak představím databázový server MySQL a relativně novou techniku pro vývoj interaktivních aplikací – AJAX. Okrajově se poslední podkapitola dotkne i problematiky LDAP serverů, protože je pomocí tohoto protokolu řešena autentizace uživatelů (viz. kapitola 4.4.1).

Všechny mnou použité nástroje jsou volně dostupné, nepodléhají tedy žádné speciální licenční politice.

### 2.1 HTML

**HyperText Markup Language** (volně přeloženo jako hypertextový značkovací jazyk) je základním jazykem internetu. Téměř každá webová stránka je vytvořena pomocí nějaké formy tohoto jazyka. Již samotný název definuje dvě základní a stěžejní vlastnosti: hypertext a univerzálnost. Slovem hypertext rozumějme schopnost vytvořit vzájemné propojení webových stránek, které nás může provázet internetem z jedné stránky na druhou. Univerzálnost spočívá v tom, že programy napsané v jazyce HTML jsou ukládány jako textové soubory – odpadá tedy problém kompilace a přenos mezi platformami. Prostý text je možno číst na všech platformách, takže nezáleží na tom, jestli čtenář vašich stránek používá Windows, Macintosh, Unix nebo si je prohlíží v mobilním telefonu.

Tím se dostáváme k problému prohlížení. Jak jsem již napsal, textový dokument se značkami jazyka HTML si může otevřít každý, ale to není příliš uživatelsky přívětivý způsob prohlížení webových stránek. Tudíž potřebujeme prohlížeč – software – který nám stránku zobrazí a podle HTML značek zformátuje. Problém nastává v tom, jak daný program stránku zobrazí. Existuje mnoho výrobců a ne všichni stejně dodržují standard jazyka. Vše závisí také na typu počítače, monitoru atp. Při tvorbě webových stránek je tedy nutné se zaměřit i na kompatibilitu jednotlivých prohlížečů, respektive platforem [1].

V této bakalářské práci jsou všechny stránky testovány v nejnovějších verzích těchto prohlížečů: Internet Explorer, Opera a Mozilla Firefox.

#### 2.1.1 Vývoj jazyka HTML

V roce 1989 pracoval Tim Berners-Lee na propojeném informačním systému pro výzkumné středisko CERN. V té době se obvykle používaly pro tvorbu dokumentů jazyky TeX, PostScript a SGML. Berners-Lee si uvědomoval, že je potřeba vytvořit nějaký jednodušší způsob. Takže v roce

1990 navrhnul základy jazyka HTML a protokolu pro jeho přenos po síti – HTTP. Současně s tím také napsal první webový prohlížeč, který se jmenoval WorldWideWeb.

O rok později CERN zprovoznil svůj web a začaly vznikat další webové prohlížeče, hlavně pro platformy IBM PC a Macintosh. Prvním grafickým byl v roce 1993 program Mosaic.

Následoval rychlý rozvoj webu a začala být potřeba definovat pro HTML standardy [2].

#### **Verze jazyka [2]:**

- Verze 0.9
  - Vydána v roce 1991, nepodporuje grafický režim.
- Verze 2.0
  - Stav jazyka v polovině roku 1994, přidána podpora interaktivních formulářů a grafiky.
- Verze 3.2
  - Plánovaná verze 3.0 nebyla nikdy přijata jako standard, protože byla příliš složitá. Proto vychází až tato verze, kterou vydává (a také i všechny ostatní) organizace W3C 14. ledna 1997. Přidává tabulky, zarovnání textu a stylové elementy.
- Verze 4.0
  - Vychází 18. prosince 1997, přidány nové prvky pro tvorbu tabulek a formulářů. Hlavní novinkou jsou rámce (frames). Snaží se dosáhnout původního účelu – prvky mají vyjadřovat význam, styly mají určovat vzhled.
- Verze 4.01
  - Byla vydána 24. prosince 1999, měla to být poslední verze, plánoval se přechod na XHTML.
- Verze 5
  - Pracovní skupina HTML, která byla založena v roce 2007, si klade za cíl vytvořit novou verzi tohoto jazyka do konce roku 2012. Základem jí mají být Web Applications 1.0 a Web Forms 2.0 ze specifikace WHATWG.

## **2.1.2 Popis jazyka**

Je charakteristický množinou značek (tj. tagů) a jejich atributů, specifických pro danou verzi. Značky jsou obvykle párové, tím určují sémantiku pro text nebo jiný element, který uzavírají. Značky je možné neomezeně zanořovat do sebe. Atributy určují doplňující informace o významu značky (např. atribut `align` u odstavce určuje zarovnání textu) [2].

Dokument HTML je rozdělen do 4 částí:

- definice typu dokumentu – povinná od verze 4.01, direktiva `<!DOCTYPE>`
- kořenový element – uzavírá celý dokument mezi značky `<html>` a `</html>`

- hlavička – obsahuje metadata, která se vztahují k celému dokumentu (např. kódování, jazyk, titulek okna prohlížeče atd.). Je ohraničena značkami <head> a </head>
- tělo dokumentu – vlastní text stránky, ohraničeno značkami <body> a </body>

Příklad dokumentu HTML [2]:

```
<!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>
```

## 2.2 CSS

CSS neboli česky **tabulky kaskádových stylů** řeší jeden ze základních požadavků jazyka HTML a tím je oddělení definic vzhledu stránek od samotného obsahu. Na počátku byly webové stránky formátovány pomocí HTML značek a případně jejich atributů. Tento stav začal být neúnosný v době, kdy vývojáři začali psát čím dál tím složitější dokumenty. Pak se nezdálo, že formátovací značky zabíraly více místa než samotný obsah, což ve výsledku znamenalo narůstající nároky na šířku přenosového pásma, respektive na výkon serverů.

Původní návrh kaskádových stylů vytvořil Håkon Wium Lie, od té doby už organizace W3C standardizovala dvě specifikace – CSS1 a CSS2. A další jsou v přípravě – CSS2.1 a CSS3.

Hlavní výhodou je tedy oddělení obsahu od vzhledu, ale také nesrovnatelně větší možnosti formátování dokumentů pomocí stylů. Dalším velkým přínosem je snadná tvorba a údržba stylů. Když se například rozhodnete změnit barvu a styl písma celého vašeho webu, tak podle původního přístupu byste museli projít všechny HTML dokumenty a upravit stovky značek a jejich atributů. Pokud využijete CSS, tak změníte kód barvy a typ písma na jediném místě – v souboru kaskádových stylů. Z toho plyne značná úspora času a zefektivnění práce tvůrcům webů.

Tímto ale výčet výhod tohoto jazyka nekončí. Je možné nejen formátovat výstup na obrazovku, ale také vzhled dokumentu při tisku (např. můžeme zakázat tisk menu, nebo grafických prvků, které by znepřehledňovaly výstup z tiskárny), projekci, nebo třeba při prohlížení dokumentu na PDA. Tvůrci specifikace mysleli i na postižené osoby, díky tomu je možné nadefinovat styly například i pro hlasový syntetizátor nebo hmatovou čtečku Braillova písma.

V neposlední řadě je možné zkombinovat CSS s JavaScriptem a zařídit tak dynamické chování vašich stránek (např. můžete nastavit barvu pozadí stránky podle denní doby) [3, 4].

Příklad definic stylů [4]:

```
h1 {                                /* vzhled nadpisu první úrovně */
    margin: 5px;                    /* okraj šířky 5 pixelů          */
    font-size: 12pt                /* velikost písma 12 bodů      */
}
p {                                  /* styl odstavce             */
    text-align: center;            /* text centrovat            */
    line-height: 10pt;            /* výška řádku 10 bodů      */
}
```

## 2.3 JavaScript

Objektově orientovaný skriptovací jazyk JavaScript byl vytvořen ve firmě Netscape koncem minulého století. Zpočátku se jmenoval LiveScript, ale pak byl z marketingových důvodů přejmenován. I když s Javou nemá nic společného.

Používá se převážně jako interpretovaný programovací jazyk v internetových prohlížečích. Jeho činnost není prováděna na serveru, nýbrž přímo na klientské stanici. Nejčastěji bývají skripty součástí HTML souboru nebo jsou připojeny v externím souboru. Jejich provádění začíná až po stažení stránky do počítače, případně až po vyvolání nějaké specifické události (např. stisk tlačítka, pohyb myši). Z důvodu ochrany soukromí uživatelů nepodporuje žádné operace se soubory.

Nejčastěji se využívá jako doplněk při tvorbě webových stránek. Dokáže do stránky přidat jistou dynamiku a bezprostřední interakci s uživatelem. Nutno ale upozornit, že jako u většiny webových technologií, tak i tady narážíme na problém (ne)kompatibility jednotlivých prohlížečů. U JavaScriptu to není až tak markantní, ale hlavně při modifikaci objektového modelu DOM se rozdíly najdou. Dalším problémem je skutečnost, že podporu JavaScriptu lze v prohlížeči vypnout. Takže bychom se neměli stoprocentně spoléhat na to, že vše bude fungovat, tak jak jsme si to naplánovali. Například kontrolovat správnost zadaných dat do formuláře pouze JavaScriptem není asi nejlepší nápad. I z tohoto důvodu jsem tento jazyk použil víceméně jako doplněk, pro mírné oživení informačního systému [5, 6].

Příklad jednoduché funkce:

```
function email(user, server, text){
    if (!text){
        text=user+"<img src=\"gui/img/at.gif\" id=\"zav\">"+server;
    }
    document.write("<a href=\"javascript:mailto(\"'+user+'\",'"+server+'
    '\');\">" +text+"</a>");
}
```

Příklady použití JavaScriptu v této práci:

- zobrazení data, hodin a aktuálního svátku podle času na serveru
- skrývání polí formuláře na základě předchozí volby z nabídky radio-tlačítek
- přidávání polí pro přílohy do formuláře
- ošetření mailto odkazů, aby nebyly čitelné pro nejrůznější roboty-spammery
- při každém použití AJAX aplikace (viz. kapitola 2.6)

## 2.4 PHP

Původně označení PHP znamenalo „Personal Home Page“, což byla technologie, kterou v roce 1994 vytvořil Rasmus Lerdorf kvůli sledování návštěvníků svých internetových stránek. S čím dál větším zájmem rostla i využitelnost tohoto skriptovacího jazyka. Po uvedení třetí verze se ujal nový rekurzivní název „PHP: Hypertext Preprocessor“.

Jak již bylo zmíněno, PHP je vložený skriptovací jazyk používaný převážně pro dynamické webové prezentace. Slovo „vložený“ v definici znamená, že kód v tomto jazyce je vložen a přímo interpretován v kódu HTML. Což samozřejmě znamená, že jde o skriptovací, nikoliv programovací jazyk. K definici je třeba ještě dodat, že jde čistě o serverovou technologii, veškerá činnost PHP skriptů probíhá na straně serveru, až její výsledek je nejčastěji ve formátu HTML, XHTML, WML či XML poslán na klientskou stanicí. Zde je už klasickou cestou zobrazen v internetovém prohlížeči, jako např. běžná HTML stránka. Interaktivitou na straně serveru se zásadně liší dynamická prezentace od statické.

Velkou předností této technologie je bezesporu nezávislost na použité platformě. PHP skripty lze prakticky bez úprav spouštět na serverech s operačním systémem Windows, Unix (v mnoha různých variantách Linuxu) nebo Macintosh. Mezi další výhody lze zařadit těsnou integraci s většinou dostupných databázových systémů (MySQL, ODBC, Oracle, MSSQL, atd.), podporu nejrůznějších síťových protokolů (HTTP, SMTP, IMAP, POP3, LDAP, atd.) a soubor nejrůznějších knihoven a frameworků pro práci např. s archivy nebo obrázky. To vše i díky velmi volné licenční politice a veřejnému zdrojovému kódu. PHP se za dobu své existence stalo již jakýmsi standardem a

ve spojení s databází MySQL a webovým serverem Apache jej nabízí valná většina hostingových firem.

Syntaxe jazyka je inspirována programovacími jazyky Perl, Pascal, C a Java. Jako velmi zajímavé vlastnosti, které bych na tomto místě chtěl zmínit, a které při implementaci velmi zjednodušují práci je dynamická typovost jazyka a práce s poli. První z nich umožňuje určit typ proměnné až v okamžiku přiřazení. Díky tomu jsou v PHP dva operátory k porovnání rovnosti == a ===. Při použití prvního dochází před samotným porovnáním ke konverzi datových typů podle potřeby, u druhého operátoru ke konverzi nedochází a porovnání má kladný výsledek pouze v případě, že oba operandy jsou shodného datového typu. Práce s poli je v každém jazyce velmi specifická. V případě PHP je možné použít číselné indexování polí (podobně jako v jazyce C) nebo lze využít pole jako hash-mapu. Obě předcházející metody lze i vzájemně kombinovat, takže jedno pole může být indexováno oběma způsoby najednou. Pole jsou heterogenní, to znamená, že jedno pole může mít prvky různých typů. Jak je vidět, tak možnosti polí jsou velmi rozsáhlé, je ale potřeba různé kombinace datových typů a indexování používat obezřetně, protože případná chyba se projeví až za chodu. Odhalení příčiny nemusí být zrovna jednoduché. Oba předcházející syntaktické prostředky si ukážeme na jednoduchém příkladu jazyka PHP [10, 11, 12].

```
// Do proměnné je možné dát pole, které obsahuje jak čísla, tak znaky či
další pole
$pole = array('a', 'b', 1, 2, array('první' => 'podpole', 'vytištěno'));

// Vytiskne obsah proměnné pole
print_r($pole);

// Test porovnání
$cislo = 100;
$retez = '100';

// Toto porovnání ('==') platí díky automatické typové konverzi
if ($retez == $cislo)
{
    echo 'Jsou stejné';
}

// Ale porovnání pomocí '===' neplatí, neboť nejsou stejné typy
if ($retez === $cislo)
{
    echo 'To by nešlo';
}
```

*Ukázka kódu v jazyce PHP [10]*

## 2.4.1 Vývoj jazyka PHP

Od počátku existence se o vývoj PHP staral pouze Rasmus Lerdorf. V roce 1997 si nové verze jazyka všimli Andy Gutmans a Zeev Suraski, kteří hledali nástroj, který by jim usnadnil vývoj

řešení elektronického obchodování. Zjistili ale, že jazyku chybí řada podstatných funkcí. Spojili se s Lerdorfem a rozhodli se jej zcela přepracovat. Tak vznikla verze PHP 3.

#### **2.4.1.1 PHP 3**

Nový tříčlenný tým autorů chtěl zdůraznit, že PHP už není jen pro osobní potřebu, změnili tedy význam zkratky PHP. Gutmans a Suraski navrhli a implementovali nové rozhraní API, které umožňovalo tvorbu dalších rozšíření, např. pro podporu databází nebo ke kontrole pravopisu. Díky tomu se k projektu PHP připojilo mnoho dalších vývojářů, kteří se starali o tyto navazující technologie.

#### **2.4.1.2 PHP 4**

V roce 1998 se autoři ohlédli za svou práci v projektu PHP a přišli k závěru, že mohli skriptovací stroj napsat mnohem lépe. Rozhodli se pro jeho přepracování. Hlavní novinkou, jež výrazně zvýšila výkon jazyka, byla změna přístupu k samotnému zpracování skriptů. Ve verzi 3 se skript zpracovával již během čtení, načež v nové verzi se autoři přiklonili k přístupu překlad-zpracování. Skriptovací stroj překládal vstupní skripty do interního byte-kódu, který pak byl interpretován pomocí stroje Zend Engine, což bylo srdce prostředí PHP 4.

Mezi další vylepšení této verze byla změna API s lepším výkonem, vrstva serverové abstrakce umožnila provozovat skriptovací stroj na nejrůznějších webových serverech. Byly představeny superglobální proměnné `$_GET`, `$_POST`, `$_SESSION` atd. aby mohla být ve výchozím nastavení zakázána direktiva `register_globals`, což byla v mnoha aplikacích velká bezpečnostní díra, jelikož autoři těchto aplikací správně neověřovali vstupní proměnné.

#### **2.4.1.3 PHP 5**

Zatím poslední verze jazyka PHP přinesla revoluční změny hlavně v objektově orientovaném přístupu, který byl v podstatě od verze 3 prakticky nezměněn. Dále obsahuje množství nových funkcí, které z PHP tvoří rozhodující platformu pro vývoj webových aplikací. Nejvíce se to odráží v práci s XML dokumenty, které jsou důležitým prostředníkem mezi různorodými systémy.

Hlavní slabinou na poli objektově orientovaného programování byla v předchozích verzích hlavně sémantika kopírování objektů, která byla stejná pro objekty i klasické datové typy. Při přiřazení proměnné, která odkazovala na objekt, došlo k vytvoření kopie tohoto objektu, což bylo pro programátory nejen nečekané chování (ve srovnání např. s jazykem Java), ale hlavně to ovlivňovalo výkon celé aplikace a občas to vedlo i k nevysvětlitelnému chování a chybám. Bylo tak nutné takovéto proměnné předávat explicitně odkazem, což také zneprůhledňovalo celý kód. Ve verzi 5 už je celá infrastruktura OOP přepsána tak, aby proměnné pracovaly s odkazy na objekty, k duplikaci objektu dochází jen při použití klíčového slova `clone`.

Seznam nových funkcí v této verzi je velmi rozsáhlý, tudíž vybírám jen ty nejpoužívanější a nejzajímavější z nich. Podrobný popis najdete v dokumentaci na oficiálních stránkách projektu PHP [www.php.net](http://www.php.net) [11].

- Modifikátory `public`, `private`, `protected` pro metody a vlastnosti tříd
- Unifikovaný název konstruktoru a destrukturu objektu - `__construct()`, `__destruct()`
- Byla zavedena rozhraní, třída může být odvozena jen od jednoho předka, může však implementovat libovolný počet rozhraní (stejně jako v jazyce Java)
- Operátor `instanceof`, konečné metody a třídy, statické metody a členy tříd, abstraktní metody a třídy

## 2.5 MySQL

Jeden z nejoblíbenějších a ve spojení s PHP asi i nejpoužívanějších databázových systémů s veřejně dostupným zdrojovým kódem. Hlavně při použití v malých a středních projektech je životaschopným konkurentem nesrovnatelně dražších produktů, jakými jsou např. Oracle nebo MS SQL server. Stejně jako tomu bylo u jazyka PHP i MySQL nabízí výborný výkon, přenositelnost a hlavně minimální provozní náklady. Lze jej provozovat na nejrůznějších platformách počínaje MS Windows a nejrůznějšími distribucemi Linuxu konče.

Tento systém byl vyvinut a je udržován švédskou společností MySQL AB. Jedná se o SŘBD určený pro relační databáze. Jde tedy o relační databázi, což je systém založený na vzájemně provázaných tabulkách. Každá tabulka obsahuje řádky (tj. záznamy), některé sloupce v ní mohou být chápány jako cizí klíče (odkazy na záznamy v jiných tabulkách), což je ve svém důsledku interpretováno jako relace mezi záznamy, použijeme-li matematickou terminologii [12].

Rozlišujeme čtyři typy vztahů (= relací):

- Mezi tabulkami není žádný vztah.
- 1:1 – záznamu v tabulce A odpovídá právě jeden záznam v tabulce B a naopak. V praxi se tato relace používá spíše ojediněle, protože je vhodnější tabulky sloučit do jedné.
- 1:N – nejčastější typ vztahu – jednomu záznamu z A odpovídá více záznamů v B – např. vztah třída-student (v jedné třídě studuje více studentů).
- M:N – přiřazuje více záznamům z A několik záznamů v B. V praxi se tento vztah transformuje na dva vztahy typu 1:N, které odkazují do pomocné (vazební) tabulky.

Databázový systém MySQL je stejně jako PHP dostupný i se zdrojovými kódy, takže jej můžeme nejen bezplatně používat, ale také upravovat. Existují ale výjimky, kdy je nutné za jeho použití



zaplatit. Především tehdy, účtujeme-li si za jeho distribuci nebo za začlenění do licencovaného produktu peníze [12, 13].

Samotný systém se skládá z několika částí:

- Server MySQL – démon `mysqld`, který spouští a udržuje databáze.
- Klient MySQL – aplikace `mysql`, jež je rozhraním především pro správu serveru.
- Další podpůrné nástroje – např. MySQL Administrator – grafické prostředí pro správu lokálního serveru, ale i vzdálenou administraci databází a jejich tabulek.

## 2.6 AJAX

Další možnost jak vytvářet všestranné a interaktivní webové aplikace, které jsou stále více uživatelsky přívětivé. AJAX je spíše technikou nežli speciální technologií. Spojuje v sobě v podstatě všechny dříve zmíněné technologie a jazyky. Základem je JavaScript ve spojení s některým serverovým programovacím jazykem (v našem případě PHP), ideálně ještě napojeným na databázový server.

Takto vytvořené aplikace využívají asynchronní přenos dat mezi prohlížečem a webovým serverem pomocí objektu v JavaScriptu zvaném `XMLHttpRequest`. Pro výměnu informací se serverem se nejčastěji užívá formát zpráv XML, i když lze data posílat i v jiných formátech (např. HTML, JSON, prostý text apod.).

Program využívající techniku AJAXu lze poznat podle toho, že běží lokálně na počítači klienta, který na základě podnětu uživatele (nemusí tomu ale tak být vždy, může to být i automaticky generovaný podnět na základě jiných událostí, např. časovaná událost) dynamicky mění obsah stránky bez nutnosti aktualizace celého dokumentu. Typickým využitím jsou různé našeptávače u formulářových polí.

Nejsilnější stránkou je právě ono aktualizování obsahu na pozadí, zároveň to v některých případech může znamenat i značné komplikace. Hlavní problém je v tom, že uživatelé jsou zvyklí na to, že webové aplikace fungují na principu požadavek/odpověď. Díky AJAXu však toto omezení zcela odpadá. Je proto třeba AJAX využívat v rozumné míře, tak abychom nezmátli uživatele naší aplikace.

Hlavním matoucím elementem a také jednou z nevýhod je nemožnost použití tlačítek Zpět/Vpřed v internetovém prohlížeči. Stejně tak není možné uložit si aktuální stránku do oblíbených položek, nebo poslat odkaz kamarádovi. Tato omezení jsou pochopitelná a vychází ze samotného principu této techniky. Jelikož obsah stránky změněný pomocí AJAXu se netváří jako nová stránka, takže ani v adresním řádku se pochopitelně nic nemění. Není možné se tedy k předcházejícímu obsahu zase jednoduše vrátit, jedinež za použití další aplikační logiky, což s sebou nese nemalou režii [7, 8].

Termín AJAX se poprvé veřejně objevil v dubnu 2005 v článku Jesse James Garretta, nazvaném Ajax: A New Approach to Web Applications (Ajax: Nový přístup k webovým aplikacím). Myšlenky, na kterých je AJAX založen, jsou však výrazně starší. Mezi začátky lze zařadit zavedení elementu IFRAME v Microsoft Internet Exploreru 3.0 z roku 1996, elementu LAYER v Netscape Navigatoru 4.0 z roku 1997 (tento element byl opuštěn na počátku vývoje Mozilly). Také Macromedia Flash od verze 4 umožňoval komunikaci se serverem na pozadí, bez překreslení stránky [9].

## 2.7 LDAP

Jelikož je v informačním systému použita autentizace uživatelů prostřednictvím LDAP serveru, tak se v krátkosti zmíním i o tomto protokolu.

LDAP je protokol sloužící k ukládání a prohlížení záznamů na adresářovém serveru. Data jsou na serveru uspořádána do stromové struktury. Každá položka v adresáři tvoří samostatný záznam na dané pozici, stejně jako tomu je u klasické adresářové struktury na disku. Protokol LDAP byl vytvořen jak již samotný název napovídá jako odlehčená verze odvozená od protokolu X.500.

Samotný adresář na serveru umožňuje ukládání různých typů dat, je však optimalizován především pro záznamy, které se moc často nemění. Na rozdíl od databázových serverů nepodporuje složité transakce a nezajišťuje integritu dat. Je tedy vhodný pro ukládání záznamů v předem specifikovaném formátu. Každý záznam musí být instancí některé třídy z předem definovaného schématu (jméno, příjmení atp.) nebo je možné vytvořit si i schéma vlastní [14].

Pro práci s adresářem je zapotřebí klientská aplikace, která provádí v zásadě dvě operace:

- Autentizační – přihlášení uživatele k LDAP serveru.
- Aktualizační a dotazovací – adresářový server umožňuje několik základních operací se záznamy (vyhledávání, porovnávání, přidávání, mazání, úprava), jejich provádění je závislé na právech přihlášeného uživatele.

V mém bakalářském projektu využívám pouze operaci autentizační přes zabezpečený kanál TLS/SSL. Jako server mi slouží Novell NetWare. Dále se o celém procesu autentizace rozepisuji v kapitole 4.4.1.

## 3 Návrh systému

Tato kapitola vás seznámí především s prvotní přípravou celého projektu, strukturou aplikace a rozdělením uživatelských rolí.

Celá tato fáze projektu trvala v podstatě celý zimní semestr letošního akademického roku. Bylo potřeba prodiskutovat a hlavně ve VOŠ vysvětlit význam a přínos této aplikace pro jejich práci. Z čehož následně vznikla následující specifikace požadavků.

V prvopočátcích práce jsem se já sám i vedení školy seznamoval s různými informačními systémy vysokých škol, na kterých studují lidé z mého okolí. Jednak jsem chtěl trochu načerpat inspiraci, ale hlavně jsem chtěl vedení VOŠ představit různé přístupy v komunikaci se studenty. Měl jsem možnost porovnat systémy naší fakulty, Masarykovy Univerzity a Právnické fakulty Univerzity Palackého v Olomouci.

### 3.1 Specifikace a analýza požadavků

Vyšší odborná škola sociálně právní má požadavek na vytvoření webového informačního systému zaměřeného hlavně na usnadnění komunikace mezi studenty a učiteli. Není potřeba, aby systém vedl celkovou školní agendu, na to škola využívá jiný software. O uživateli tedy stačí uchovávat pouze základní osobní a kontaktní údaje. Je nutné implementovat kvalitní vyhledávací a filtrovací subsystém pro snadné vyhledání požadované osoby.

Škola má pouze tři třídy v jednom studijním oboru. V každém ročníku studuje cca 30 studentů, do budoucna se nepočítá s navyšováním tohoto počtu, v systému tak budou třídy definovány staticky.

Systém musí umožnit práci se školními aktualitami. Zaměstnanci s dostatečným oprávněním budou moci vytvářet aktuality pro různé skupiny uživatelů. Budou zde aktuality veřejné, které budou dostupné pro nepřihlášeného uživatele a v budoucnu budou zobrazovány na oficiálních stránkách školy. Dále pak aktuality určené pro studenty a zvláště pak zprávy dostupné pouze pro zaměstnance školy. K aktualitám se bude připojovat neomezený počet příloh.

Hlavním úkolem informačního systému bude informovat studenty o zapsaných předmětech. Každý předmět má jednoho nebo maximálně dva vyučující, kteří budou spravovat informace o daném předmětu. Rozlišujeme tři typy předmětů: povinný, volitelný a povinně volitelný. Většina předmětů je povinných, volitelných absolvují studenti během studia zhruba jen 4. Každý předmět má svůj syllabus a je k němu možné přidávat interní zprávy a přikládat soubory se studijními materiály. Dále je možné k předmětům přidávat události (zkoušky, úkoly) a jejich varianty. Přihlašovat se na ně mohou studenti buď sami, automaticky, nebo je ručně přihlásí jeden z učitelů předmětu. Studenti zde pak budou odevzdávat výstupy své domácí práce a zjistí zde i její ohodnocení.

Hodnocení ve škole nemá žádný jednotný řád, tak jak ho známe např. z naší fakulty. Každý učitel si v předmětu definuje svoje vlastní pravidla a uděluje studentům za úkoly nebo zkoušky buď body, nebo známky. Tuto „volnost“ je nutné zachovat i do budoucna. Stejně tak celkové hodnocení z předmětu není možné díky výše popsanému nikterak automatizovat. Učitelé jej budou zadávat ručně na konci zkuškového období.

Systém musí běžet na dostupném školním zařízení. K dispozici je tedy server s operačním systémem Windows, na kterém běží kromě jiného webová prezentace školy. Je zde nainstalován webový server Apache 2.2, PHP verze 5.2.6 a MySQL server 5.0.45. Dále je ve školní síti přítomen server Novell NetWare, kde mají uživatelé svoje osobní profily. K autentizaci do informačního systému by bylo vhodné tento server využít, aby uživatelé mohli používat jednotné přihlašovací údaje jako pro přihlášení do sítě.

### 3.1.1 Typy uživatelů

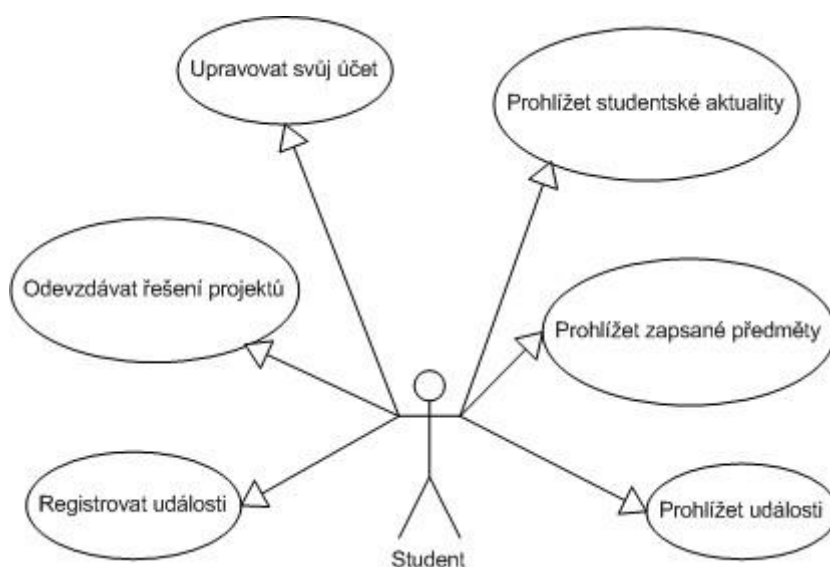
V systému jsou čtyři typy uživatelů, odstupňované podle oprávnění k operacím, které mohou provádět. Uvedu jejich výčet i s příslušnými use-case diagramy, které demonstrují výčet povolených operací.

#### 3.1.1.1 Nepřihlášený uživatel

Může pouze prohlížet veřejné aktuality na titulní stránce informačního systému, žádné další operace neprovádí.

#### 3.1.1.2 Student

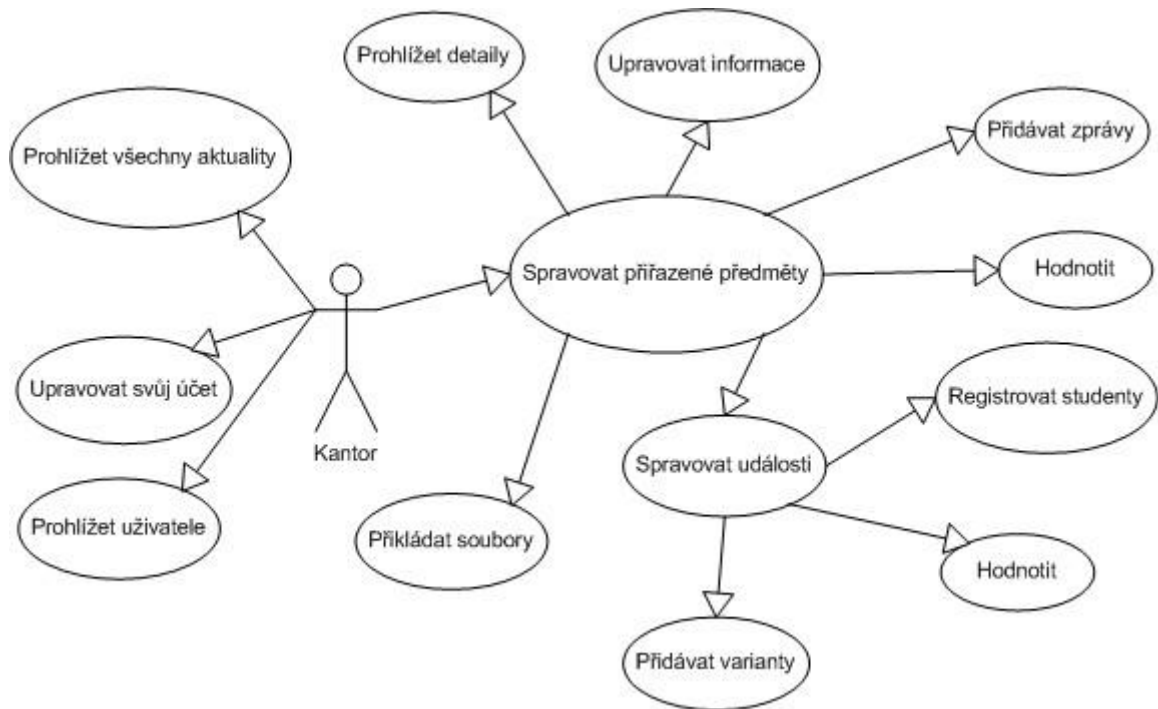
Má nejnižší oprávnění. Kromě studentských a samozřejmě i veřejných aktualit se dostane pouze k informacím a událostem týkajících se předmětů, které má zapsané.



Obrázek 3.1: Use-case diagram: Student

### 3.1.1.3 Kantor

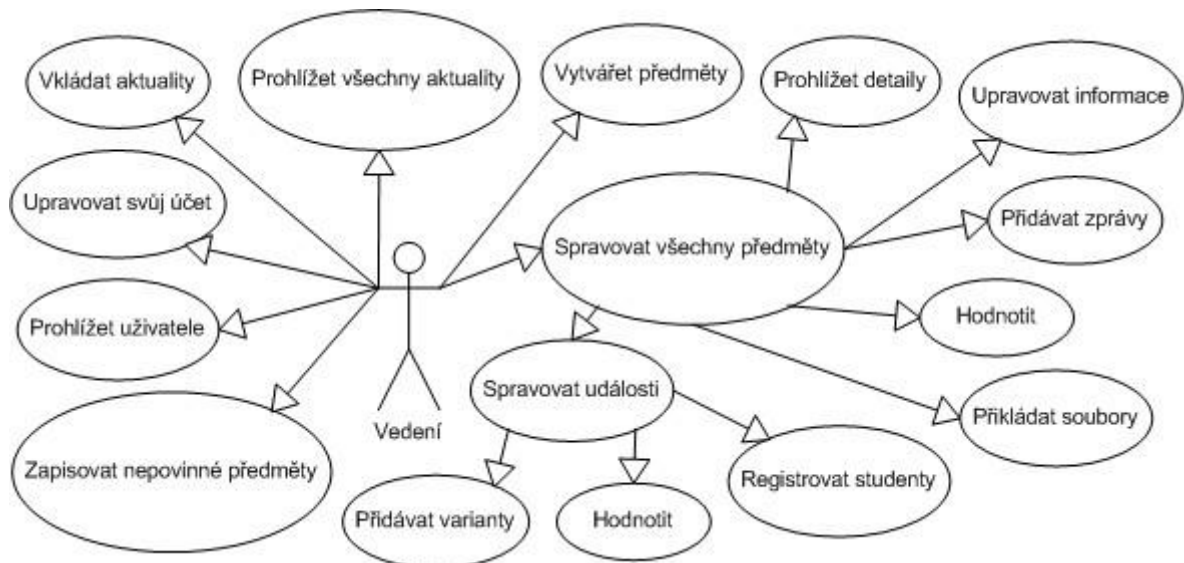
Pro čtení má již k dispozici všechny typy aktualit včetně interních-zaměstnaneckých. Spravuje předměty jemu přiřazené. Může prohlížet osobní údaje všech uživatelů.



Obrázek 3.2: Use-case diagram: Kantor

### 3.1.1.4 Vedení

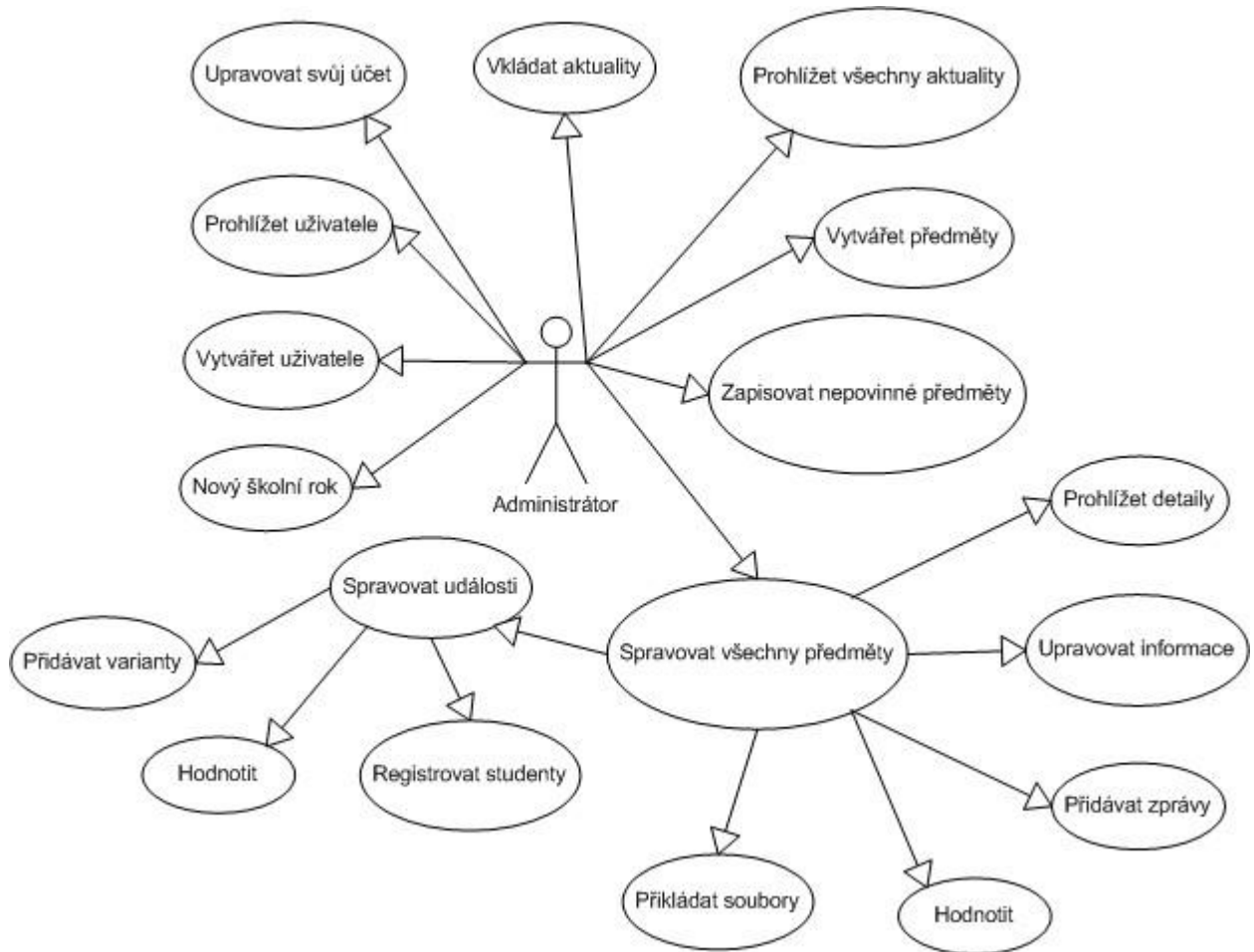
Osoba z vedení školy, kromě všeho výše zmíněného, může aktuality i vytvářet. Dále je mu umožněno vytvářet nové předměty, přiřazovat k nim kantory a zapisovat studentům nepovinné předměty. Má přístup ke všem předmětům v systému, tak jakoby mu byly přiřazené.



Obrázek 3.3: Use-case diagram: Vedení

### 3.1.1.5 Administrátor

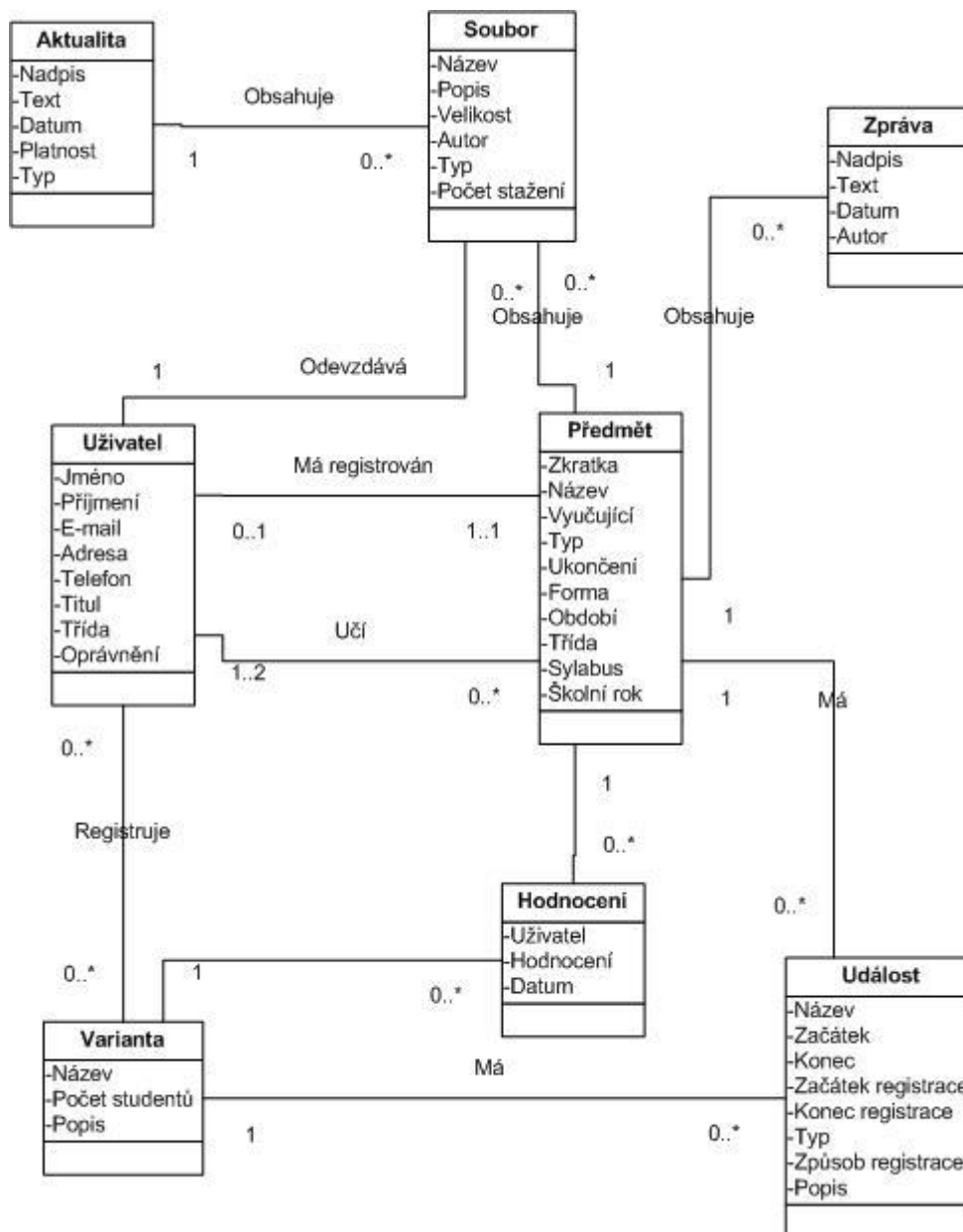
Nejvyšší možné oprávnění. Má přístup ke všem předešlým operacím. Navíc může přidávat nové uživatele a editovat všechny uživatelské účty. Spravuje veškerá nastavení systému, včetně přechodu na nový školní rok.



Obrázek 3.4: Use-case diagram: Administrátor

## 3.2 E-R diagram

Entity relationship diagram na obrázku 3.5 popisuje vztahy entit (= objektů v systému) a jejich integritní omezení.

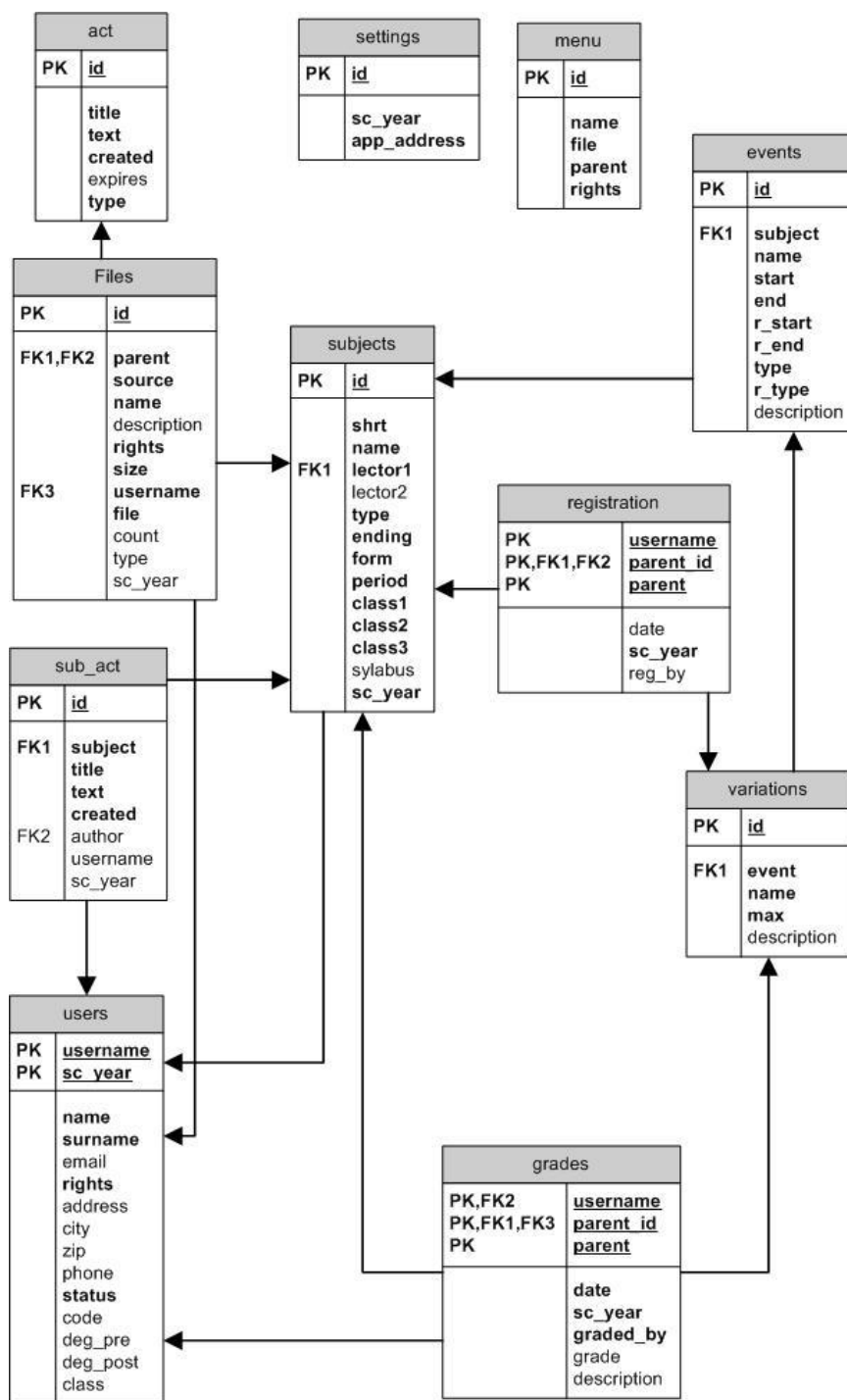


Obrázek 3.5: E-R diagram

### 3.3 Schéma databáze

V současné době je v databázi 11 tabulek, z čehož jsou dvě pomocné, které obsahují nastavení systému a obsah menu, které se automaticky generuje v uživatelském rozhraní.

Celkový model databáze je na obrázku 3.6. Skript pro vytvoření této databáze je přiložen na CD v adresáři db.



Obrázek 3.6: Schéma databáze

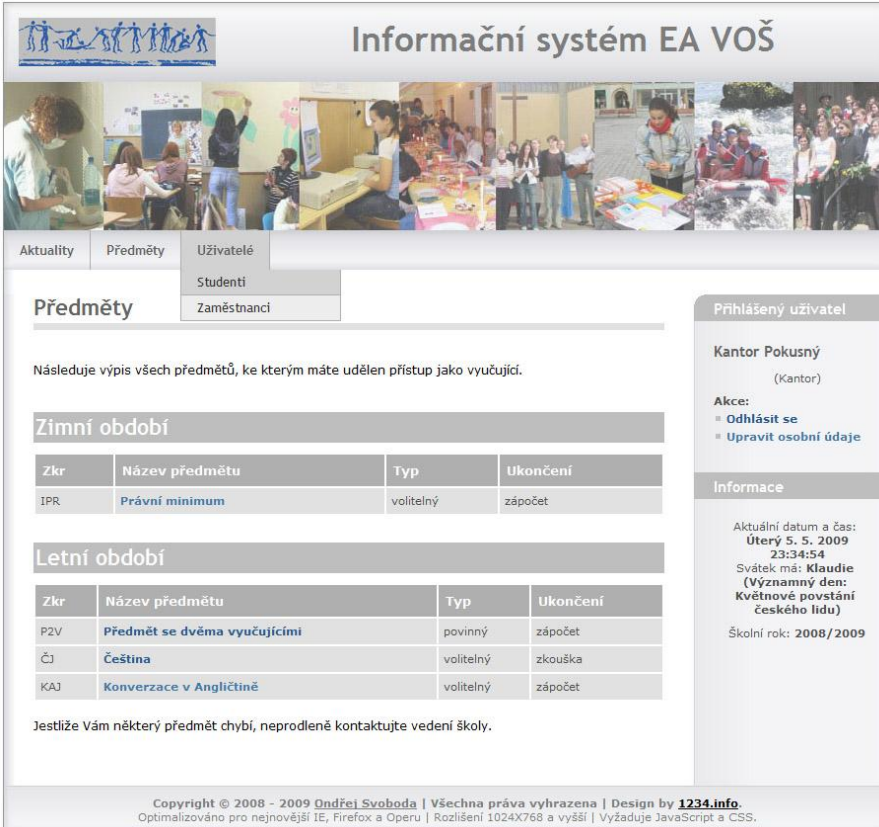


## 3.4 Uživatelské rozhraní

Jsem zastáncem tvrzení, že uživatelské rozhraní musí být především jednoduché, účelné a intuitivně ovladatelné. Až poté přicházejí na řadu estetické aspekty. Jelikož ale nejsem grafik a moje kreativita v tomto směru je značně omezená, rozhodl jsem už při tvorbě webových stránek pro tuto školu, využít nějaké volně dostupné šablony. Stejný design jsem použil i nyní, nejenom aby bylo vše jednotné, ale také proto, že jsem s ním byl velmi spokojen.

Zmíněnou šablonu je možné najít na adrese [www.1234.info](http://www.1234.info), jedná se open source projekt, který je možno jakkoliv upravovat. Autor na výše zmíněné adrese vytvořil již pět velmi zdařilých vzorů stránek. Já jsem si vybral ten s názvem „Multiflex 3“. Ukázku můžete vidět na obrázku 3.7. Celé uživatelské prostředí je vytvořeno pomocí kaskádových stylů. Já jsem si jej v některých detailech mírně poupravil a obohatil jej o několik JavaScriptových funkcí.

Všechny stránky v informačním systému jsem pečlivě testoval ve třech nejrozšířenějších internetových prohlížečích, kterými jsou: Internet Explorer (verze 7 a 8), Mozilla Firefox (verze 3.0.10) a Opera (verze 9.64). V některých případech bylo potřeba kód stránek upravovat, aby byly ve všech těchto prohlížečích zobrazeny alespoň trochu podobně. A hlavně aby všechny funkce zůstaly zachovány, ať už si aplikaci otevřete v jakémkoliv z nich. Největší problémy byly u některých funkcí JavaScriptu, samotné stránky už se zobrazují v nejnovějších verzích tří zmíněných prohlížečů velmi podobně.



**Informační systém EA VOŠ**

Aktuality | Předměty | Uživatelé

Studenti | Zaměstnanci

**Předměty**

Následuje výpis všech předmětů, ke kterým máte udělen přístup jako vyučující.

**Zimní období**

| Zkr | Název předmětu | Typ       | Ukončení |
|-----|----------------|-----------|----------|
| IPR | Právní minimum | volitelný | zápočet  |

**Letní období**

| Zkr | Název předmětu               | Typ       | Ukončení |
|-----|------------------------------|-----------|----------|
| P2V | Předmět se dvěma vyučujícími | povinný   | zápočet  |
| ČJ  | Čeština                      | volitelný | zkouška  |
| KAJ | Konverzace v Angličtině      | volitelný | zápočet  |

Jestliže Vám některý předmět chybí, neprodleně kontaktujte vedení školy.

**Přihlášený uživatel**

Kantor Pokusný  
(Kantor)

Akce:

- Odhlásit se
- Upravit osobní údaje

**Informace**

Aktuální datum a čas:  
Úterý 5. 5. 2009  
23:34:54  
Svátek má: **Klaudie**  
(Významný den:  
Květnové povstání  
českého lidu)

Školní rok: 2008/2009

Copyright © 2008 - 2009 Ondřej Svoboda | Všechna práva vyhrazena | Design by 1234.info  
Optimalizováno pro nejnovější IE, Firefox a Operu | Rozlišení 1024X768 a vyšší | Vyžaduje JavaScript a CSS.

Obrázek 3.7: Ukázka uživatelského rozhraní

## 4 Implementace

Po předchozí přípravné a plánovací fázi už následuje samotná tvorba informačního systému. V jeho implementaci jsem postupoval systematicky. Vždy jsem vytvořil jednu ucelenou část nebo celý modul, který jsem náležitě otestoval a prokonzultoval se zástupci školy. Po případných úpravách jsem pokračoval v dalším vývoji.

V tomto systému práce ostatně hodlám pokračovat i v budoucím zdokonalování a přidávání nových funkcí. Těch nápadů a plánů je už teď celá řada, s těmi nejvýznamnějšími vás seznámím hned v následující kapitole.

Tato kapitola detailněji popisuje implementaci některých základních a zajímavých funkcí systému.

### 4.1 Struktura aplikace

Aplikace je logicky rozdělena na tři části. Prezentační, aplikační a datovou logiku.

Prezentační část informačního systému najdeme v podadresáři `gui`. Jsou zde skripty obsahující jednotlivé části výsledných stránek – hlavička, patička atp. Také zde najdeme kaskádové styly a soubory s JavaScriptovými funkcemi.

V kořenovém adresáři je hlavní jádro systému – aplikační logika. Tyto skripty obsluhují všechny funkce aplikace. Spojují tak v sobě obě zbývající části. Jsou v nich vytvářeny jednak objekty tvořící výslednou stránku, ale také funkce, které získávají potřebná data z databáze.

Téměř všechny databázové dotazy jsou přehledně uloženy zvlášť v souboru `SQL.php` v podadresáři `lib`. Samotné napojení na databázi zajišťuje třída `Db`. Parametry databázového spojení jsou uloženy v souboru `db_settings.php`.

#### 4.1.1 Objektově orientovaný přístup

Při programování aplikace jsem využil objektově orientovaného přístupu. Všechny třídy jsou uloženy v samostatných souborech v adresáři `classes`.

Vytvořil jsem třídy pro všechny tři vrstvy aplikace, jejichž funkce je často provázaná:

- Třídy prezentační logiky
  - `Page` – Pomocí něj se tvoří celá výsledná stránka, která je po provedení všech operací odeslána uživateli metodou `show()`. Dále poskytuje metody pro nastavení titulku okna (`setTitle()`), obsahu stránky (`setContent()`), přidat další obsah (`addContent()`), umístit stránkování (`addPaging()`), atd.

- `Table` – Podpora pro tvorbu tabulek. Metody pro nastavení popisku (`setLabel()`), přidání řádku (`addRow()`), konverze tabulky do HTML kódu (`toHTML()`), atp.
- `Form` – Podobně jako u předchozí třídy, se pomocí této tvoří uživatelské formuláře. Pomocí metody `addItem()` lze přidávat jednotlivé položky formuláře, tlačítka (`addButton()`) a samozřejmě převést výsledek do HTML (`toHTML()`).
  - Od této třídy jsou odvozeny dva speciální formuláře – `LoginForm` (přihlašovací formulář) a `SearchUserForm` (vyhledávací formulář). U těchto typů formulářů nebylo možné využít předchozího přístupu, kvůli jinému designu na stránce.
- `Error` – Zobrazí uživateli chybovou stránku. Např. když nemá dostatečné oprávnění pro prováděnou operaci.
- Třídy aplikační logiky
  - `User` – Obsahuje všechny atributy uživatele, které načítá (metoda `load()`) a ukládá (`save()`) do databáze.
  - `Subject` – Informace o konkrétním předmětu, opět zde funguje stejný princip komunikace s databází.
  - `Event` – Událost předmětu.
  - `Auth` – Bezpečnostní třída. Pomocí ní je řešeno přihlašování (`login()`) a odhlašování (`logout()`) uživatelů. Dále poskytuje metody pro ověření oprávnění uživatele a stavu jeho uživatelského účtu. Pracuje s využitím `SESSION`.
- Třídy datové logiky
  - `Db` – Zajišťuje komunikaci s databází. Metoda `query()` vyhledá požadovaný dotaz v souboru `SQL.php` a dotáže se databáze, přičemž chrání dotaz před sql injection pomocí funkce PHP `mysql_real_escape_string()`. Výsledek dotazu lze prozkoumat pomocí funkce `getRow()`, můžeme získat počet vrácených řádků (`getRowCount()`) nebo u dotazů typu `INSERT` zjistit ID naposledy vloženého řádku (`getLastId()`).

## 4.2 Použití AJAXu

Jak jsem již napsal v kapitole 2.6 AJAX je potřeba používat rozumně a na vhodných místech. Já jsem se tohoto tvrzení držel i ve své práci a této techniky jsem (zatím) využil jen ve dvou následujících případech.

## 4.2.1 Našeptávač uživatelů

Je využit vždy, když je potřeba, aby uživatel do formuláře vyplnil jméno nebo uživatelské jméno některého uživatele. Například u vytváření předmětu nebo vyhledávání informací o uživatelích. V takovém případě našeptávač nabídne uživateli, po zadání 1-N písmen, možnosti z databáze.

Na straně klienta běží pouze zobrazovací logika napsaná v JavaScriptu. Veškeré vyhledávací operace obstarává skript `autoSuggest.php` už na straně serveru. Podle předaných parametrů metodou `GET` hledá zadaná písmena buď v celém jméně uživatele, nebo v jeho uživatelském jméně. V některých případech je také potřeba vyhledávání omezit jen na některé typy uživatelů. Např. u předmětů se vyplňují jejich vyučující, bylo by jistě nežádoucí uživateli napovídat i jména studentů. Toho je dosaženo použitím vhodné klauzule `WHERE` u databázového dotazu.

Výsledek vyhledávání obsahuje pro přehlednost maximálně 5 položek, které posílá zpět na klientskou stanici ve formátu XML. Tato data už převezme a zobrazí klientský JavaScript, který čeká na odpověď serveru.

## 4.2.2 Mazání souborů

Jedná se o mazání souborů přiložených k aktualitám, předmětům, nebo odevzdaných studentem jako řešení projektu (úkolů). Po kliknutí na odkaz „odstranit“ u výpisu souborů je na pozadí vyslán požadavek na server, který je zpracován skriptem `deleteFile.php`. Tento ověří oprávnění uživatele k požadovanému souboru. Pokud vše proběhne správně, soubor je smazán z databáze a klientovy je poslána informace o úspěšnosti/neúspěšnosti operace.

Po úspěšném provedení operací na serveru JavaScriptová funkce `deleteFileHandler()` odstraní z výpisu smazaný soubor pomocí funkce `removeChild()`. V opačném případě je uživateli zobrazeno chybové hlášení.

## 4.3 Předměty

Momentálně nejdůležitějším modulem je správa předmětů a jejich událostí. V současné verzi zápis nepovinných předmětů provádějí uživatelé s oprávněním vedení a vyšší. Je to kvůli malému počtu těchto typů předmětů. Studenti jich za celou dobu studia absolvují zhruba jen čtyři. Uvidíme jak se tato forma „zápisu“ osvědčí v praxi.

Správu předmětů zajišťují soubory `subjects.php` a `subDetail.php`. První z nich uživateli zobrazí seznam dostupných předmětů. U studentů seznam zapsaných předmětů, u učitelů předměty, které učí. Uživatelé s oprávněním vedoucí a vyšší mají v druhé tabulce na této stránce

k dispozici seznam všech předmětů na škole. Stránka s detailními informacemi umožňuje učitelům přidávat zprávy, soubory a události. Dále může upravovat informace k předmětu (např. sylabus).

Události a jejich varianty zpřístupňuje soubor `events.php`. U nich rozlišujeme tři typy: zkouška, úkol a ostatní. Rozdíl je v tom, že u zkoušky a typu ostatní není studentům umožněno odevzdávat soubory – výstupy jejich domácí práce. Dále je u událostí možné nastavit začátek a konec a začátek a konec registrace, která může být provedena automaticky, kantorem, nebo studentem. Studentům jsou zobrazovány pouze již započaté události, tzn. ty, jejichž datum začátku je nižší než aktuální datum.

Události jsou variantní – každá musí mít alespoň jednu variantu, na kterou je možné studenty (nechat) zaregistrovat. U automatického způsobu registrace je přípustná pouze jedna varianta. Případné řešení studenta je možné stahovat zvlášť, pro celou variantu, nebo celou událost dohromady. Všechny soubory v dané skupině jsou zkomprimovány metodou ZIP. Obsah `.zip` souboru je hierarchicky členěn, pro lepší orientaci, podle variant (pokud stahujeme řešení celé události). Soubory jednoho uživatele jsou opět uloženy v jednom adresáři.

Hodnocení učitel zadává pro každou variantu zvlášť a také ukládá výslednou známku z předmětu. Vždy je zobrazen seznam studentů, u každého z nich je možno zadat libovolnou hodnotu a krátký vysvětlující popis hodnocení. Záleží jen na domluvě, jestli se jedná o bodovou interpretaci, nebo číslo (či písmeno) reprezentuje známku.

## 4.4 Správa uživatelů

Probíhá na dvou úrovních. Seznam uživatelů je jednak na LDAP serveru, který zajišťuje autentizaci do školní sítě, a také v databázi informačního systému, která obsahuje další údaje uživatele (bydliště, e-mail, telefon atd.). Společným klíčem těchto úložišť, je uživatelské jméno, které musí být na obou místech shodné.

Kromě autentizace je zásadní operací nad množinou uživatelů vyhledávání. Toto usnadňuje již zmíněné našeptávání, dále je možné uživatele filtrovat podle typu na studenty a zaměstnance. U studentů je možné další třídění podle tříd.

Databázi uživatelů mohou prohlížet jen kantoři a osoby s vyšším oprávněním. Upravovat a přidávat údaje může pouze administrátor, každý uživatel si jinak spravuje pouze svůj účet.

Speciální operací, kterou může administrátor vykonat je blokování/odblokování účtu. Blokovánému uživateli pak není umožněno přihlášení do systému. Využil jsem této možnosti místo úplného mazání kvůli uchování historie, kterou bude možné v některé z následujících verzí prohlížet a také kvůli konzistenci databáze.

## 4.4.1 Autentizace uživatelů pomocí LDAP serveru

Jak už bylo zmíněno, autentizace probíhá ve dvou fázích. Nejprve je uživatel ověřen vůči LDAP serveru. K tomu slouží v PHP celá knihovna funkcí. Já jsem použil tři z nich:

- `ldap_connect()` – Jak název napovídá tak tato funkce slouží k připojení k LDAP serveru. Nicméně není to tak úplně pravda. Tato funkce pouze inicializuje spojení, ke skutečnému připojení dochází až při prvním volání následující funkce.
- `ldap_bind()` – Ustanovuje spojení s LDAP serverem na základě zadaných parametrů (kontextu a hesla).
- `ldap_errno()` – Zjistí chybový kód poslední LDAP operace.

Kvůli vlastnostem funkce `ldap_connect` nezjistíme, jestli připojení k serveru proběhlo úspěšně. A až po zavolání funkce `ldap_bind`, která vrátí `FALSE`, víme, že nastal problém. Pomocí třetí zmíněné funkce, `ldap_errno`, teprve ověříme, jestli je problém ve spojení nebo v nesprávně zadané kombinaci uživatelské jméno-heslo.

Pokud předchozí ověření dopadlo úspěšně, musíme ještě ověřit, zda zadaný uživatel má v informačním systému vytvořen účet a jestli není zablokován. Potom jej můžeme úspěšně přihlásit. Veškerá komunikace mezi LDAP a webovým serverem probíhá zabezpečeným SSL spojením.

## 4.4.2 Ověření funkčnosti elektronické adresy

Jelikož studenti nemají ve škole zřízenou vlastní elektronickou adresu a pro komunikaci používají svou vlastní, bylo potřeba implementovat mechanismus ověření funkčnosti této adresy.

Držel jsem se již zavedených postupů, jak je známe z nejrůznějších serverů. Po vytvoření nového uživatele, nebo po odblokování jeho účtu je „donucen“ vyplnit/upravit některé osobní údaje včetně e-mailové adresy při prvním přihlášení do systému. Po ověření platnosti těchto dat, je na zadanou adresu odeslána testovací zpráva s odkazem pro aktivaci účtu. To stejné platí i pro situaci, když si uživatel jeho adresu změní.

Dokud není účet aktivován, nemůže uživatel provádět žádné operace, kromě změny e-mailu a znovu odeslání zprávy.

## 4.5 Přejít na nový školní rok

Tato operace patří výhradně pod správu administrátora. V podstatě to znamená pouze přidat jeden řádek do tabulky `settings`. Nicméně aby byl celkový proces usnadněn, může si administrátor vybrat, které uživatele a předměty přenesou do následujícího školního roku. Není pak nutné je všechny znovu zadávat.

Pohledem na obrázek 3.6 zjistíte, že uživatel je definován nejen svým uživatelským jménem, ale také školním rokem. Stejná situace platí i u předmětů. Zvolené položky pro přenos jsou tedy zkopírovány znovu do stejných tabulek, s tím rozdílem, kterým je již změněný školní rok. U všech studentů prvního a druhého ročníku také dochází při kopírování k inkrementaci čísla ročníku. Těm, kterým se tento „postup“ v praxi nezdařil, je nutné nastavit zpět nižší ročník v osobním profilu po kompletním přechodu.

## 4.6 Použité externí knihovny a třídy

Při implementaci informačního systému jsem využil tři externí knihovny nebo třídy. Jednalo se vždy o doplňkové funkce, které usnadnily práci mě, nebo vylepšily a doplnily funkcionalitu pro uživatele.

Všechno to jsou doplňky s volně šiřitelným zdrojovým kódem. Originální balíky, které jsem použil, jsou uloženy adresáři `lib` na přiloženém CD.

Následuje jejich výčet a krátký popis.

### **PHPMailer**

Třída usnadňující práci s odesíláním elektronických zpráv prostřednictvím PHP. Zatím jsem ji sice použil pro posílání testovacích zpráv při prvním přihlášení nového uživatele, ale s hlavním využitím se počítá v budoucím vývoji. V dohledné době bude totiž implementována možnost odesílání skupinových e-mailů včetně příloh. Což za pomoci standardních funkcí jazyka PHP je velmi nepohodlné a neefektivní naprogramovat.

Podrobnější informace získáte na stránkách projektu: <http://phpmailer.codeworxtech.com>.

### **FCKeditor**

WYSIWYG editor pro webové stránky napsaný v jazyce JavaScript s integračními nástroji pro různé další platformy – např. PHP. V mé aplikaci jej využívám jako vstup pro texty aktualit, popisků, sylabů u předmětů atd. Je možné definovat nejen vzhled, ale i soubor operací, které uživateli chceme poskytnout.

Stránky projektu: <http://www.fckeditor.net>.

### **AJAX Auto Suggest**

Klientská část našeptávače, která zajišťuje pouze zobrazování našeptávaných výrazů. Veškerou ostatní logiku našeptávače jsem již programoval já v jazyce PHP.

Nejen že skvěle vypadá, ale také nabízí mnoho konfiguračních možností. Lze jej ovládat jak pomocí myši, tak i pomocí šipek na klávesnici.

Stránky projektu: [http://www.brandspankingnew.net/archive/2007/02/ajax\\_auto\\_suggest\\_v2.html](http://www.brandspankingnew.net/archive/2007/02/ajax_auto_suggest_v2.html).

## 5 Plánovaná rozšíření

Jak jsem již napsal v úvodu práce, vývoj informačního systému je tak trochu běh na dlouhou trať. Již teď mám v hlavě plno nápadů, které bych chtěl do něj doplnit. Další požadavky postupně přicházejí ze strany školy a hlavně budou přicházet během testování a ostrého provozu. Funkce budu postupně doplňovat a rozšiřovat.

Ze všech těch nápadů a požadavků uvádím nejvýznamnější z nich, které bych chtěl implementovat ještě během letošního léta.

### 5.1 Komunikace uživatelů

Tento pojem představuje vytvoření komunikačního modulu, který by umožňoval posílat elektronické zprávy rovnou z prostředí aplikace jednomu i více uživatelům. Odpadne tak nutnost kopírovat seznamy adres z informačního systému do poštovního klienta.

Jedním kliknutím tak bude možné odeslat zprávu např. všem studentům daného předmětu, nebo studentům přihlášeným na určitý projekt nebo zkoušku.

Dále bude umožněno vytvářet si vlastní ručně definované seznamy příjemců, opět s použitím AJAXového našeptávání.

Přítomnost této funkcionality je hlavním důvodem, proč jsem pro odesílání e-mailů použil knihovnu PHPMailer, která mi implementaci velmi usnadní.

### 5.2 Tiskové sestavy

Je potřeba také vytvořit systém tiskových výstupů, tak aby bylo možné vytisknout v rozumném formátu (bez obrázků, barevných popisků atd.) nejrůznější údaje a hlavně seznamy.

Bude se jednat hlavně seznam studentů přihlášených na zkoušku nebo projekt. Dále také závěrečnou zprávu z předmětu, která bude obsahovat studenty předmětu i s jejich výsledným hodnocením. Tak, aby jej vyučující jen podepsal a odevzdal k založení. Učitelům tak odpadne významné množství práce, což je bude motivovat k využívání této aplikace.

### 5.3 Hodnocení výuky a školy

Tuto funkcionality je potřeba doimplementovat do konce zimního období v příštím školním roce. Bude umožňovat administrátorovi vytvářet libovolné dotazníky ohledně předmětů a celé školy, jejichž výstup bude využit pro evaluační zprávu školy.



K vytvoření tohoto modulu bude potřeba přidat několik tabulek do databáze a také navrhnout a implementovat celý systém tvorby dotazníků. Bude se využívat DOM funkcí v JavaScriptu, možná i AJAX zde najde své uplatnění.

Výstupem by měly být tabulky s hodnotami, ale hlavně přehledné grafy. V tomhle by mohla pomoci knihovna pro tvorbu grafů JpGraph (<http://www.aditus.nu/jpgraph>). Také i zde by byl žádoucí tisknutelný výstup výsledků.

Druhým využitím tohoto modulu by mohlo být i zavedení anket u předmětů. Vyučující každého předmětu by si mohl vytvořit dotazník, který by studenti vyplnili. Umožnilo by to tak zpětnou odezvu ze strany studentů, což je mnohdy velmi žádoucí.

## 6 Závěr

Cílem mé bakalářské práce bylo vytvořit informační systém pro Vyšší odbornou školu sociálně právní, Opletalova 6, Brno, tak aby usnadnil komunikaci se studenty a přehledně na jednom místě zpřístupňoval všechny důležité informace.

V druhé kapitole jsem popsal mnou použité technologie. Jejich seznam byl v podstatě předem dán současným hardwarovým a softwarovým vybavením školy. Téměř všechny tyto jazyky a techniky jsem znal z předchozího studia a také z některých externích projektů. Nebylo tak nutné se s nimi již blíže seznamovat, nebo se je učit. I když v některých oblastech jsem si musel svoje znalosti zdokonalit a rozšířit. Hlavní neznámou při přípravě projektu byla komunikace s LDAP serverem, což byla pro mě nová záležitost. I toto se ale podařilo zvládnout a jsem rád, protože to mému systému přidalo na profesionalitě a hlavně na uživatelské přívětivosti.

V další kapitole jsem popsal asi nejdůležitější část vývoje každého informačního systému – modelování. Jak je známo, tak dobrý model dokáže usnadnit mnoho práce při programování. Nic není ale bez chyb, a jelikož jsem během implementace narazil mnohdy i na úplně nové skutečnosti, bylo nutné občas původní model trochu upravit. Nejhorší ovšem je zásadní změna původních požadavků, která dokáže udělat v již částečně hotovém projektu velké nepříjemnosti. To se mi naštěstí nestalo.

Čtvrtá kapitola shrnuje implementaci význačných funkcí systému. Nezacházel jsem zde do přílišných detailů, ty si může čtenář nastudovat přímo ze zdrojových souborů na přiloženém CD. Kód jsem se snažil přehledně formátovat a komentovat, tak aby se v něm dalo dobře zorientovat.

Všechna plánovaná rozšíření z páté kapitoly bych chtěl dokončit do konce srpna tohoto roku, aby mohly být nasazeny začátkem následujícího školního roku do ostrého provozu. Nejsou to rozhodně ani zdaleka všechna vylepšení, která jsou v plánu. Dalším je například přítomnost fotky u každého uživatele a možnost tak zobrazovat a tisknout třídní alba. Což je nyní velmi využíváno v papírové podobě. Učitelé mají ve sborovně plakáty s fotkami a jmény všech studentů, umožňuje jim to jednoduše přiřadit ke jménu obličej, když paměť selhává.

Jelikož byla aplikace vyvinuta speciálně pro tuto školu a dané technické vybavení, nebyl kladen důraz na přenositelnost a snadnou instalaci. Máte ovšem možnost vyzkoušet si demoverzi aplikace na internetu. Adresu i přístupová hesla jednotlivých uživatelů najdete na přiloženém CD v souboru `readme.txt`.

Tvorba této aplikace byla zatím mým největším projektem a zpočátku i velkou výzvou. Určitě mě obohatila o nové zkušenosti hlavně ve vztahu se zadavatelem, což je zásadní věc v obchodním životě, která se nedá ve školních podmínkách prakticky vůbec natrénovat. Už nyní se rýsuje možná spolupráce s partnerskou školou v Brně, která potřebuje vytvořit informační systém na evidenci studijních materiálů. Pokud vše dobře dopadne, tak zde zhodnotím cenné zkušenosti z tohoto bakalářského projektu.

# Literatura

- [1] CASTRO, Elizabeth. HTML, XHTML a CSS : názorný průvodce tvorbou WWW stránek. Brno: Computer Press, a.s., 2007. 438 s. ISBN 978-80-251-1531-2.
- [2] Wikipedie: Otevřená encyklopedie: HyperText Markup Language [online]. c2009 [citováno 8. 04. 2009]. Dostupný z WWW: [http://cs.wikipedia.org/w/index.php?title=HyperText\\_Markup\\_Language&oldid=3781987](http://cs.wikipedia.org/w/index.php?title=HyperText_Markup_Language&oldid=3781987)
- [3] CYROŇ, Miroslav. CSS - kaskádové styly: praktický manuál. 1. vyd. Praha: Grada Publishing a.s., 2006. 340 s. ISBN 80-247-1420-5.
- [4] Wikipedie: Otevřená encyklopedie: Cascading Style Sheets [online]. c2009 [citováno 8. 04. 2009]. Dostupný z WWW: [http://cs.wikipedia.org/w/index.php?title=Cascading\\_Style\\_Sheets](http://cs.wikipedia.org/w/index.php?title=Cascading_Style_Sheets)
- [5] ŠKULTÉTY, Rastislav. JavaScript : Programujeme internetové aplikace. 2. aktualiz. vyd. Brno: Computer Press, a.s., 2004. 224 s. ISBN 80-251-0144-4.
- [6] Wikipedie: Otevřená encyklopedie: JavaScript [online]. c2009 [citováno 8. 04. 2009]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=JavaScript&oldid=3806344>
- [7] ASLESON, Ryan, T.SCHUTTA, Nathaniel. AJAX: Vytváříme vysoce interaktivní webové aplikace. 1. vyd. [s.l.] : [s.n.], 2006. 272 s. ISBN 80-251-1285-3.
- [8] LACKO, Luboslav. AJAX: Hotová řešení. 1. vyd. Brno: Computer Press, a.s., 2008. 269 s. ISBN 978-80-251-2108-5.
- [9] Wikipedie: Otevřená encyklopedie: Asynchronous JavaScript and XML [online]. c2009 [citováno 8. 04. 2009]. Dostupný z WWW: [http://cs.wikipedia.org/w/index.php?title=Asynchronous\\_JavaScript\\_and\\_XML&oldid=3637015](http://cs.wikipedia.org/w/index.php?title=Asynchronous_JavaScript_and_XML&oldid=3637015)
- [10] Wikipedie: Otevřená encyklopedie: PHP [online]. c2009 [citováno 3. 05. 2009]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=PHP&oldid=3848813>
- [11] GUTMANS, Andi, SAETHER BAKKEN, Stig, RETHANS, Derick. Mistrovství v PHP 5. Bogdan Kiszka. Brno : Computer Press, 2005. 656 s.
- [12] ULLMAN, Larry. PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. 1. vyd. Brno : Computer Press, 2004. 534 s. ISBN 80-251-0063-4.
- [13] Wikipedie: Otevřená encyklopedie: Relační databáze [online]. c2009 [citováno 3. 05. 2009]. Dostupný z WWW: [http://cs.wikipedia.org/w/index.php?title=Rela%C4%8Dn%C3%AD\\_datab%C3%A1ze&oldid=3760214](http://cs.wikipedia.org/w/index.php?title=Rela%C4%8Dn%C3%AD_datab%C3%A1ze&oldid=3760214)
- [14] Wikipedie: Otevřená encyklopedie: LDAP [online]. c2009 [citováno 3. 05. 2009]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=LDAP&oldid=3861893>

# Seznam příloh

Příloha 1: Použité zkratky

Příloha 2: Hodnocení školy

Příloha 3: CD se zdrojovými kódy

# Příloha 1: Použité zkratky

**VOŠ** – Vyšší odborná škola

**HTML** – HyperText Markup Language

**CSS** – Cascading Style Sheets

**PHP** – PHP: Hypertext Preprocessor (původně Personal Home Page)

**AJAX** – Asynchronous JavaScript and XML

**LDAP** – Lightweight Directory Access Protocol

**HTTP** – HyperText Transfer Protocol

**W3C** – World Wide Web Consortium

**WHATWG** – Web Hypertext Application Technology Working Group

**XML** – eXtensible Markup Language

**JSON** – JavaScript Object Notation

**OOP** – Objektově orientované programování

**SŘBD** – Systém řízení báze dat

**DOM** – Document Object Model

# Příloha 2: Hodnocení školy

EVANGELICKÁ AKADEMIE BRNO



Evangelická akademie,

Vyšší odborná škola sociálně právní

602 00 Brno, Opletalova 6, tel/fax: 542 221 741

V Brně dne 9. května 2009

## Bakalářská práce „Informační systém pro VOŠ sociálně právní“

Autor bakalářské práce s názvem „Informační systém pro VOŠ sociálně právní“ Ondřej Svoboda spolupracuje s naší školou již několik let. Jako externí zaměstnanec zaváděl do provozu školy systém pro evidenci žáků SAS a rovněž je autorem internetových stránek školy. S radostí jsme přijali jeho nabídku na vytvoření informačního systému pro naši školu.

Evangelická akademie, Vyšší odborná škola sociálně právní vzdělává v akreditovaném programu Sociálně právní činnost budoucí sociální pracovníky. Studium na vyšší odborné škole je organizováno v rámci terciárního vzdělávání a jeho forma se v mnohém podobá vysokoškolskému systému. Během studia se předpokládá, že se studenti připravují poměrnou částí formou samostudia, konzultují s vyučujícími problematiku jednotlivých disciplín jak ústní formou, tak možností dálkového přístupu na školní server.

Informační systém podstatně zefektivní systém kontaktů s vyučujícími, předávání informací o průběhu studia včetně zkoušek a zápočtů, zadávání úkolů. Informační systém jsme ve škole několikrát s autorem bakalářské práce konzultovali tak, aby vyhovoval požadavkům školy. Do konce letošního školního roku proběhne testovací fáze a od září nového školního roku 2009/2010 bude využíván naplno. Počítáme také s tím, že by se informační systém v dalším období rozšířil o další operace a funkce.

Informační systém je velkým přínosem pro provoz naší školy.

Mgr. Hana Svobodová

ředitelka školy

## Příloha 3: Obsah příloženého CD

Příložené CD obsahuje zdrojové kódy aplikace, tento text ve formátech PDF a DOC a také použité externí knihovny. Má následující adresářovou strukturu (zjednodušeno):

- is/
  - classes/
    - zdrojové soubory tříd aplikační a datové logiky
  - db/
    - install.sql – instalační skript databáze
  - gui/
    - prezentační logika aplikace
  - lib/
    - SQL.php – obsahuje SQL dotazy
    - externí knihovny a třídy v originálním „balení“
  - + zdrojové soubory aplikační logiky
- text/
  - db.doc
  - db.pdf
- readme.txt – základní informace o aplikaci, přístupové údaje do testovací verze