



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

NÁSTROJ PRO TVORBU A SPRÁVU OBSAHU WEBOVÝCH PREZENTACÍ

WEB PRESENTATION TOOLS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. IVAN DEKAN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MILOSLAV RICHTER, Ph.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Bc. Ivan Dekan

ID: 47586

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Nástroj pro tvorbu a správu obsahu webových prezentací

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte bezpečný, vícejazyčný systém pro správu webového obsahu (WCMS - web content management system) založený na technologiích PHP, MySQL a JavaScript. Systém vytvořte modulárně, s přehledným uživatelským rozhraním a se správou uživatelských účtů a jejich oprávnění. Vytvořte rozšiřující moduly: novinky, anketa, fotogalerie, kniha návštěv. Převeďte na vzorové aplikaci s návodem pro tvorbu.

DOPORUČENÁ LITERATURA:

Gilmore J.W.: Velká kniha PHP 5 a MySQL - kompendium znalostí pro začátečníky i profesionály. Brno, Zoner press, 2005, ISBN 80-86815-20-X.

Resig J.: JavaScript a Ajax - moderní programování webových aplikací. Brno, Computer Press, 2007, ISBN 978-80-251-1824-5.

Staniček P.: CSS Kaskádové styly - kompletní průvodce. Brno, Computer Press, 2003, ISBN 80-7226-872-4.

Termín zadání: 8.2.2010

Termín odevzdání: 24.5.2010

Vedoucí práce: Ing. Miloslav Richter, Ph.D.

prof. Ing. Pavel Jura, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Diplomová práce se zabývá systémem pro správu webového obsahu (CMS – content management system) s unikátním grafickým prostředím a charakteristickými vlastnostmi cloud computing operačních systémů.

Jádro systému tvoří:

- na straně serveru kód psaný v programovacím jazyce PHP a
- na straně klienta JavaScript s asynchronním přenosem dat (technologie AJAX).

K uložení dat a nastavení systému je použita databáze MySQL. Jádro systému dokáže vytvářet a spravovat libovolný počet uživatelských oken a zajišťuje synchronizaci dat pracovního prostředí mezi klientem a serverem. Systém je programován modulárně - moduly lze rozšířit systém o nové funkce. Systém obsahuje několik často používaných modulů: správa uživatelů, správa obsahu, události, fotogalerie, soutěže a další. K rychlému a snadnému vytvoření nových modulů slouží několik tříd, které umožňují generovat a spravovat základní systémové objekty. Textová část modulů je načítána z jazykového souboru, což umožňuje překlad systému do více jazyků.

Abstract

The content management system with unique graphics environment and characteristic of cloud computing operating systems is major subject of this thesis.

The core of the system consist of:

- PHP code on the server side
- JavaScript with data asynchronous transmission (AJAX technology) on the client side.

MySQL database is used for saving data and system's setting. The core of the system is able to create and manage any number of user's window and also handle synchronization of data between client and server. System is programmed modularly – system is able to expand by modules. System contains several frequently used

modules as: user manager, content manager, events, photogallery, contests, etc. To create new modules quickly and easily are prepared some classes which can manage and generate basic system's objects. Translation of system to any language is possible because of the text part loading from a language file.

Bibliografická citace

DEKAN, I. Nástroj pro tvorbu a správu obsahu webových prezentací . Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 46 s. Vedoucí diplomové práce Ing. Miloslav Richter, Ph.D.

Klíčová slova

cms, systém pro správu obsahu, správa webového obsahu, cloud computing, webový operační systém, webová aplikace, AJAX, redakční systém, kniha návštěv, fotogalerie

Keywords

cms, content management system, cloud computing, web operating system, web application, AJAX, guestbook, photogallery

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Nástroj pro tvorbu a správu obsahu webových prezentací jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne: **24. května 2010**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Miloslavu Richterovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Děkuji také všem, kteří mi pomáhali s překladem a jazykovou úpravou práce.

V Brně dne: **25. května 2009**

.....
podpis autora

Obsah

Obsah	1
1. ÚVOD	3
2. POUŽITÉ TECHNOLOGIE	4
2.1 PHP	4
2.2 MySQL	5
2.3 Apache	6
2.4 JavaScript	7
2.5 AJAX	8
2.6 XHTML	9
2.7 CSS - kaskádové syly	10
3. ISYS	11
3.1 Uživatelské prostředí	11
3.1.1 Hlavní panel	12
3.1.2 Systémové okno	13
3.1.3 Pozadí pracovní plochy	24
3.2 Jádro systému	24
3.2.1 Zdrojový kód	25
3.2.2 Paměť	26
3.2.3 Zpracování stránky	29
3.3 MODULY	31
3.3.1 Systém	32
3.3.2 Správa uživatelů	32
3.3.3 Správa účtu	33
3.3.4 Diskusní fórum	34
3.3.5 Správa stránek	34
3.3.6 Správa obsahu	35
3.3.7 Fotogalerie	35
3.3.8 Novinky	37
3.3.9 Soutěže	37
3.3.10 Anketa	37

3.3.11 Vytvoření nového modulu	37
3.4 Instalace systému	38
3.5 Vzorová aplikace	39
4. DALŠÍ MOŽNOSTI VÝVOJE SYSTÉMU A JEHO ZNÁMÉ NEDOSTATKY	41
5. ZÁVĚR	42
6. SEZNAM LITERATURY.....	44
7. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ.....	45
8. SEZNAM PŘÍLOH	46

1. ÚVOD

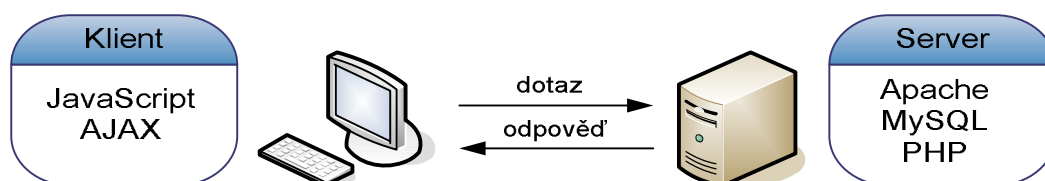
Počet aktivních webových stránek se v posledních 5-ti letech zdvojnásobil. Nemalou část z nich tvoří webové prezentace, ať už firemní či osobní. K zajištění vysoké návštěvnosti je třeba aktualizovat obsah. Od majitelů těchto stránek ale nemůžeme předpokládat programátorské schopnosti a je třeba pro ně vytvořit nástroj, kterým budou schopni snadno a intuitivně spravovat obsah svých stránek. V současné době existuje celá řada systémů pro správu obsahu (CMS – content management system) a redakčních systémů nabízených zdarma i jako komerční řešení. Cílem této práce je vytvořit systém s podobnými vlastnostmi ale s unikátním grafickým prostředím a vlastnostmi cloud computing operačních systémů. Běžný uživatel pracující s počítačem by si jej měl osvojit ve velmi krátkém čase.

Cloud computing je relativně nový pojem. Jedna z definic říká: „Termín označuje souhrnně technologie a postupy používané v datových centrech a firmách pro zajištění snadné škálovatelnosti aplikací dodávaných přes Internet.“ [4]. V blízké budoucnosti se očekávají velké investice a rozšíření těchto technologií. O běh cloud computing systémů a aplikací se stará server. Stejně tak ukládá veškerá data a dokumenty. Uživatelé stačí internetový prohlížeč, připojení k internetu a může přistupovat ke svým souborům prakticky odkudkoliv. Další výhodou jsou minimální požadavky na hardware počítače uživatele. Na druhou stranu se řeší otázka bezpečnosti, jelikož jsou data často přenášena nešifrovaně a uživatel neví, kde jsou ve skutečnosti uložena. Typickým příkladem je volně dostupný EyeOS [5]. Náš systém ale zaměříme na správu a úpravu obsahu webových stránek a označíme si ho názvem „iSys“. Svými vlastnostmi a uživatelským prostředím se bude blížit cloud computing OS, ale s funkcemi CMS. V současné době na trhu neexistuje (nebo neznám) webový operační systém tohoto zaměření.

Jako příklad použití bude systém iSys předveden na ukázkovém webu pro místní vysokoškolský klub Terč.

2. POUŽITÉ TECHNOLOGIE

Náš systém nebude svou strukturou a funkcemi natolik složitý, abychom museli použít komerční řešení. Vystačíme si tak s open-source technologiemi. Pro svoji širokou podporu ze strany serverů byl použit skriptovací jazyk PHP ve verzi 5.3.0. Přínosem je i podpora objektově orientovaného programování a soubor funkcí, které PHP obsahuje. Spolu s PHP se často k uložení dat instaluje relační databázový systém MySQL. V našem případě jde o verzi 5.1.37. K testovacím účelům poslouží softwarový webový server Apache (v. 2.2.13). Na straně klienta bude ve velké míře využito programovacího jazyka JavaScript a především AJAX, který z JavaScriptu vychází. Obsah stránek bude popsán jazykem XHTML 1.0 Strict a jeho grafická podoba upravena kaskádovými styly – CSS. Ukázkový příklad spolupráce jednotlivých technologií je ukázán na obrázku 2.1.



Obrázek 2.1 – způsob komunikace mezi klientem a serverem

2.1 PHP

Jednoduché statické webové stránky jsou již minulostí a převážná většina internetových aplikací dnes využívá kód spuštěný na serveru. PHP přináší do jinak statických stránek „paměť“ a „vstupní proměnné“ v podobě proměnných v URL adrese. Stránky jsou generovány dynamicky dle aktuálního požadavku uživatele. PHP je nejpoužívanější skriptovací jazyk určený pro vývoj webových aplikací.

Hlavním konkurentem na webovém hostingu je řešení společnosti Microsoft, technologie ASP.NET (ASP – active server pages), součást platformy .NET. Narozdíl od PHP jde o celý framework a ASP.NET tak není závislé na konkrétním

programovacím jazyce. Přestože Microsoft dostatečně podporuje svoji technologii a poskytuje dokumentaci ve formě webového portálu MSDN a .NET Frameworku, zůstaneme u PHP, které je vhodné pro menší a střední projekty a pro naše účely je více než dostačující. Srovnání obou technologií je například v [6]. Výhodou je cena (oproti ASP.NET zdarma) a široká podpora hostingů.

Při vzniku v roce 1994 označovala zkratka PHP Personal Home Page, tedy osobní domovskou stránku. Tento jazyk byl určen pouze k realizaci formulářů na webových stránkách. O rok později byla zveřejněna verze 2, která již dokázala komunikovat s databází a umožnila tvorbu jednoduchých dynamických webových stránek. Význam zkratky se změnil na „Hypertext Preprocessor“. Současná pátá verze tohoto oblíbeného skriptovacího jazyka přidává podporu objektově orientovaného programování a jazyka XML [7]. Další podrobné informace o PHP můžete nalézt např v [1].

Zdrojový kód PHP jazyka se ukládá do souborů *.php. Jednoduchý příklad cyklu a vypsání čísel 1 – 5 je ukázán zde:

```
<?php
    for($i=1;$i<=5;$i++)
    {
        echo $i;
    }
?>
```

2.2 MYSQL

MySQL je volně šířitelný multiplatformní databázový server vytvořený švédskou firmou MySQL AB dnes vlastněný společností Sun Microsystems [8]. V praxi se často používá u webových stránek: publikační a informační portály, inzerce, osobní stránky, seznamky apod. Databázový server vystupuje vůči svému okolí jako služba, která oprávněným klientům a klientským aplikacím poskytuje služby pro přístup k datům a pro jejich správu. Každá databáze se skládá z tabulek, které obsahují řádky a sloupce. Nevýhodou oproti konkurenci byla až do verze 5 absence transakcí a uložení a spuštění procedur. V nové verzi jsou však již tyto vlastnosti implementovány [7].

Databáze MySQL komunikuje s PHP skriptem velmi jednoduchým jazykem SQL. Po připojení a vybrání databáze můžeme jednoduchým příkazem z PHP vytvořit novou tabulku `test` o dvou sloupcích `jmeno` a `prijmeni`.

```
<?php
    $ret = mysql_query(CREATE TABLE `test` (`jmeno` VARCHAR (40)
    NOT NULL, `prijmeni` VARCHAR (40) NOT NULL));
?>
```

Vložení a opětovné načtení hodnot z tabulky provedeme příkazem:

```
<?php
    mysql_query("INSERT INTO test ('Ivan', 'Dekan')");
    $res = mysql_query("SELECT * FROM test");
?>
```

Databáze MySQL podporuje celou řadu datových typů do kterých je možné ukládat i například soubory. Několik často používaných je uvedeno zde:

- INT – rozsah hodnot -2147483648 do +2147483647, bez znaménka 0 až 4294967295
- VARCHAR(délka) - délka řetězce může být v rozsahu 0-255
- TEXT - nebinární řetězce, nerozlišuje velikost znaků
- DATE - datum ve formátu "RRRR-MM-DD" a v rozsahu 1000-01-01 až 9999-12-31
- DATETIME - datum a čas ve formátu "RRRR-MM-DD HH:MM:SS" a v rozsahu 1000-01-01 00:00:00 až 9999-12-31 23:59:59

2.3 APACHE

Apache je volně dostupný multiplatformní HTTP server. Za tímto projektem stojí nezisková organizace Apache Software Foundation. Podle posledních měření obsluhuje přibližně 54% všech existujících webových stránek. Hlavní rysy a výhody:

- jednoduchý, malý, rychlý, bezpečný
- rozšíření o mnoho modulů (šifrování, komprese dat)
- podpora CGI skriptů (PHP)

- autentizace, řízení přístupů
- přepisování URI
- konfigurovatelné logování

Ze stejných důvodů jako skriptovací jazyk PHP byl zvolen Apache oproti softwaru Microsoftu IIS (Internet Information Services), na kterém je v provozu 25% stránek.

2.4 JAVASCRIPT

V době kdy se začalo prosazovat CGI, vydal Netscape Communications Corporation (dříve Sun and Andreessen) nový jazyk, původně určený pro set-top boxy, s názvem Java. Nabídl ihned podporu tohoto jazyka ve svém, v té době dominantním, prohlížeči Netscape Navigator. Záměr byl stejný jako v případě CGI, učinit web více dynamický. Tak vznikly aplety – malé aplikace umístěné v obsahu webových stránek. V inkriminované době byly skvělým způsobem, jak vytvářet dynamické webové aplikace. Přestože se aplety v některých oblastech hojně rozšířily, měly i své negativní vlastnosti. Vývojáři často museli řešit kompatibilitu jednotlivých verzí Javy. Nefunkčnost apletů na odlišných verzích Javy a chybně naprogramované aplety zamezily masivnímu rozšíření.

Pro snadnější programování apletů vytvořil Netscape skriptovací jazyk JavaScript (Microsoft po té vytvořil svůj vlastní skriptovací jazyk – VBScript.). Za návrhem a implementací stál Brendana Eiche. Prvotně byl JavaScript určen pouze k dynamické změně tagů, po čase se ale ukázalo, že je možné celou stránku považovat za objekt a zrodil se DOM (Document Object Model). DOM představuje plně objektově orientovanou reprezentaci webové stránky, kterou je možné dynamicky modifikovat skriptovacím jazykem, jako je JavaScript a VBScript. O standartizaci DOMu se postaralo konsorcium W3C (World Wide Consortium) a asociace ECMA (European Computer Manufacturers Association) ratifikovala JavaScript. Dodržení standardů tak zaručuje stejnou funkčnost apletů ve všech prohlížečích dodržující tyto standardy [9].

Jak už bylo řečeno, JavaScript je stejně jako PHP skriptovací jazyk ale s tím rozdílem, že JavaScript se spouští na straně klienta. Tím jsou dána některá omezující bezpečnostní omezení. Nesmí například číst ani zapisovat do souborového systému nebo spouštět programy. Syntaxe kódu je podobná jazykům C/C++/Java. Kód lze vložit přímo do stránky nebo do vlastního souboru a uvést na něj pouze adresu. Jednoduchá ukázka vytvoření funkce a události v JavaScriptu je uvedena níže. Po kliku na odkaz zahlásí prohlížeč uživateli chybovou hlášku.

```
<html>
  <head>
    ...
    <script language="JavaScript" type="text/javascript">
      function chyba()
      {
        alert("chyba!");
      }
    </script>
  </head>
  <body>
    ...
    <a href="" onclick="chyba();">klikni zde</a>
  </body>
</html>
```

Zde bych ještě připomněl jednu, pro nás důležitou, vlastnost JavaScriptu. Ačkoliv dnešní internetové prohlížeče plně podporují JavaScript, uživatel ho může z jakýchkoliv důvodů zakázat. V našem systému iSys je JavaScript stěžejní technologie a bez ní **nebude** systém správně nebo vůbec pracovat.

2.5 AJAX

AJAX ve své podobě existuje už od dob prohlížeče Internet Explorer 5. Tehdy byla uvedena první technologie patřící dnes pod termín AJAX, objekt XMLHttpRequest (XHR). Další technologie, které AJAX drží, jsou JavaScript, DOM, CSS. Rozšíření se ale tato technika dočkala až posledních letech. Pojem AJAX poprvé použil v únoru 2005 Jesse James Garrett ve své práci „AJAX: A new Approach to Web Application“, kde rozebírá, jak se pomalu stírají rozdíly mezi klasickými klientskými aplikacemi a webovými aplikacemi. O prosazení v praxi se

postaral až Goole, přesněji řečeno služby Google Maps a Google Suggest, kde mají JavaScript velice v oblibě. Dalším příkladem může být u mnoha vyhledávačů tzv. našeptávač, kdy se při vyplňování hledaného výrazu zobrazuje nabídka výrazů podobných [9].

Princip AJAXu spočívá v odeslání dat na server, které uživatel nějakým způsobem modifikoval, poté obdržení odpovědi (XHR) a dojde k patřičné změně dokumentu (JavaScript, DOM, CSS), aniž by došlo k znovunačtení celé stránky. Pro vývojáře webových aplikací je toto velmi silný nástroj. Pokud zvládneme vytvářet dynamické webové aplikace, které dokáží samy asynchronně komunikovat se serverem, tak proč nezkusit naprogramovat webový operační systém. A jsem u cloud computingu.

Snad jedinou nevýhodou je absence ukládání stránek do historie prohlížeče během změny obsahu stránky AJAXem. Je to logický důsledek toho, že nedochází k načtení celé stránky. Někteří uživatelé pak můžou být zmateni. Existují ale techniky jak tuto slabinu odstarnit.

2.6 XHTML

Předchůdce XHTML, jazyk HTML, měl za úkol definovat vazby mezi webovými dokumenty a především popsat jejich strukturu. Určit, která část dokumentu je nadpisem, která tabulkou, odstavcem a který úsek tvoří seznam a co je jeho položkou. Způsob zobrazení zpočátku určovalo pouze výstupní zařízení (prohlížeč, tiskárna, projektor). S rostoucími potřebami vývojářů a uživatelů vzniklo výrobcí prohlížečů množství rozšíření jazyka HTML nad rámec standardu. HTML se rozpadlo do několika druhů podle různých verzí prohlížečů. Do HTML postupně přibývalo velké množství formátovacích značek a tagů. Spolu s narůstajícím počtem formátovacích značek v dokumentu se vytrácela jejich struktura. Negativní dopad mají i na velikost dokumentu. Můžou představovat až 80% jejich datového objemu.

Z těchto důvodů vznikly nové standardy XHTML a CSS, které rozdělují dokument na popis struktury a popis vzhledu. Vývoj HTML byl ukončen a nejnovější verze XHTML 1.1 všechny formátovací a podobná řešení zakazuje a jsou

považována za chybu. Důležitou podmínkou pro úspěch je dnes dobrá podpora standardů ze strany internetových prohlížečů [3].

Příklad jak zapsat nadpis s odstavcem, který mimo textu obsahuje i obrázek:

```
<h1>nadpis</h1>
<p>
text v odstavci

</p>
```

2.7 CSS - KASKÁDOVÉ SYLY

Jak už bylo zmíněno v předchozím bodě, jazyk Tabulek kaskádových stylů (CSS) je určený k formátování dokumentů. Definuje způsob jejich prezentace na koncových zařízeních (lze např. vytvořit rozdílný styl pro zobrazení na monitoru a pro tisk), popisuje podobu stránek a styl jednotlivých prvků aniž by ovlivňoval obsah dokumentu. Kaskádové styly se staly v návrhu stránek opravdovou revolucí.

Poslední uvedenou specifikací je CSS2.1 a vývojáři už několik let čekají na uvedení CSS3, které přinese mnoho nových vylepšení a možností. Očekávat můžeme modularizaci kaskádových stylů, nové selektory, pseudotřídy a vlastnosti. Standardizaci zajišťuje opět organizace W3C.

Každé pravidlo v tabulce kaskádových stylů obsahuje vždy dvě informace. Za prvé je to selektor a za druhé blok deklarací, který obsahuje seznam vlastností a jejich hodnoty. Příklad nastavení velikosti fontu a barvy pozadí u nadpisů první úrovně (v XHTML značeno „h1“) :

```
h1 { font-size:14px; background-color:white; }
```

Styly můžeme umístit přímo do hlavičky dokumentu nebo zde pouze uvést odkaz na externí soubor typu *.css. Je to praktičtější řešení, které dovoluje změnou jednoho souboru změnit vzhled celých webových stránek.

```
<head>
  <style type="text/css">
    h1 { font-size:14px; background-color:white; }
  </style>
  <link href='main.css' type='text/css' rel='stylesheet'>
</head>
```

3. ISYS

System iSys je navrhnutý jako interaktivní webový systém pro správu obsahu. Interaktivitu zaručuje JavaScript, technika AJAX popsaná v kapitole 2.5 a částečně i kaskádové styly CSS. Ke komunikaci mezi uživatele a systémem dochází prostřednictvím systémových oken. Pro názornost je na obrázku 3.1 ukázáno přihlašovací okno do systému.

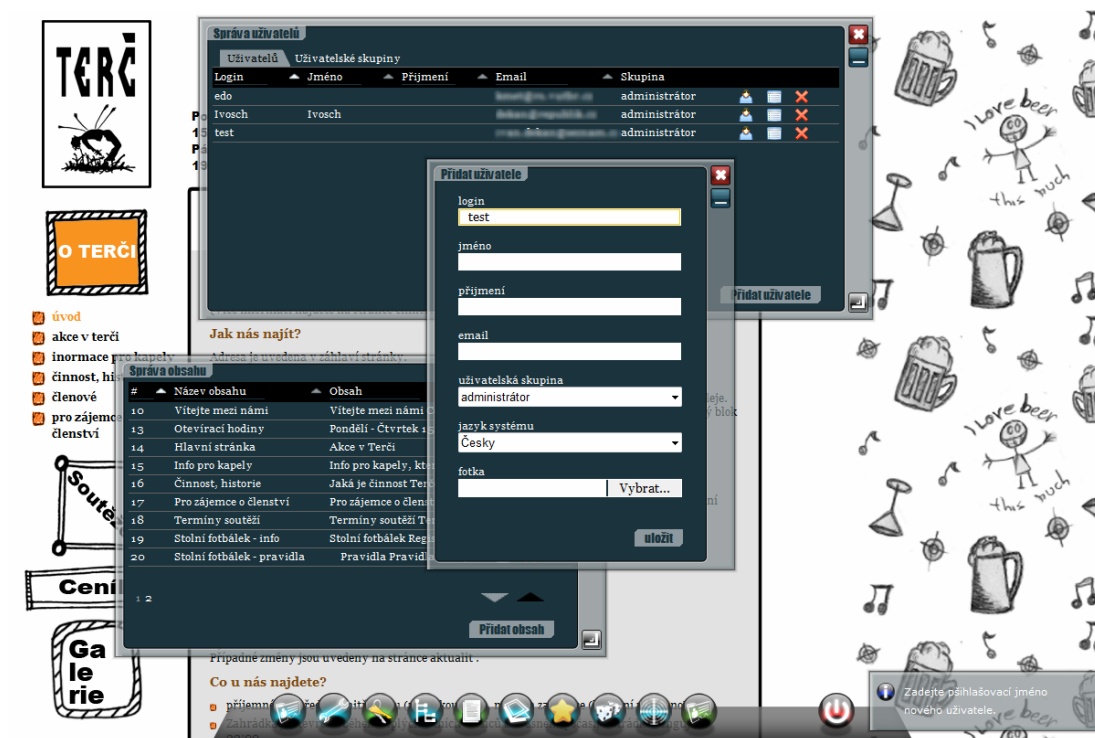


Obrázek 3.1– přihlašovací okno do webového systému

3.1 UŽIVATELSKÉ PROSTŘEDÍ

Na úvod se pokusím přiblížit uživatelské prostředí a základní vlastnosti systému. Jakmile provede uživatel jakoukoliv změnu na své pracovní ploše či v systémovém okně (může třeba přetáhnout okno do jiného místa nebo vyplnit některé textové pole), jsou změny ihned odeslány na server (AJAX) kde dojde k jejich uložení. Tímto způsobem se zajišťuje synchronizace systémových dat. Prochází-li uživatel webovou stránku na pozadí plochy nebo se jí pokusí obnovit, načte se mu pracovní plocha ve stejném stavu. Ukázka pracovní plochy je na obrázku 3.2. Naprogramované systémové funkce na straně klienta (JavaScript, AJAX) dokáží reagovat na podněty uživatele a upravovat obsah pracovní plochy dle aktuálního požadavku. Přenos dat mezi klientem a serverem technikou AJAX je signalizován

animací v levém dolním rohu. Pravý dolní roh je určený pro nápovědu a chybová oznámení.



Obrázek 3.2 – pracovní plocha iSys, otevřeno několik oken a na pozadí ukázková webová stránka VŠK Terč

3.1.1 Hlavní panel

Součástí pracovní plochy je i hlavní panel. Ten je napevno umístěn ve spod obrazovky a kromě trvale umístěných zástupců obsahuje zástupce aktuálně otevřených oken. Kliknutím na zástupce dojde k otevření okna a podbarvení zástupce zelenou barvou. Je-li okno otevřené, přenesse se do popředí. Je-li okno minimalizované, maximalizuje se.

Dojde-li k otevření nebo zavření okna, aktualizují se zástupci na hlavním panelu. Přidat, odebrat a nastavit pořadí zástupců na hlavním panelu může uživatel v nastavení účtu. Zástupci jsou na panel umístováni zleva. Na pravé straně hlavního panelu je zástupce pro odhlášení ze systému.

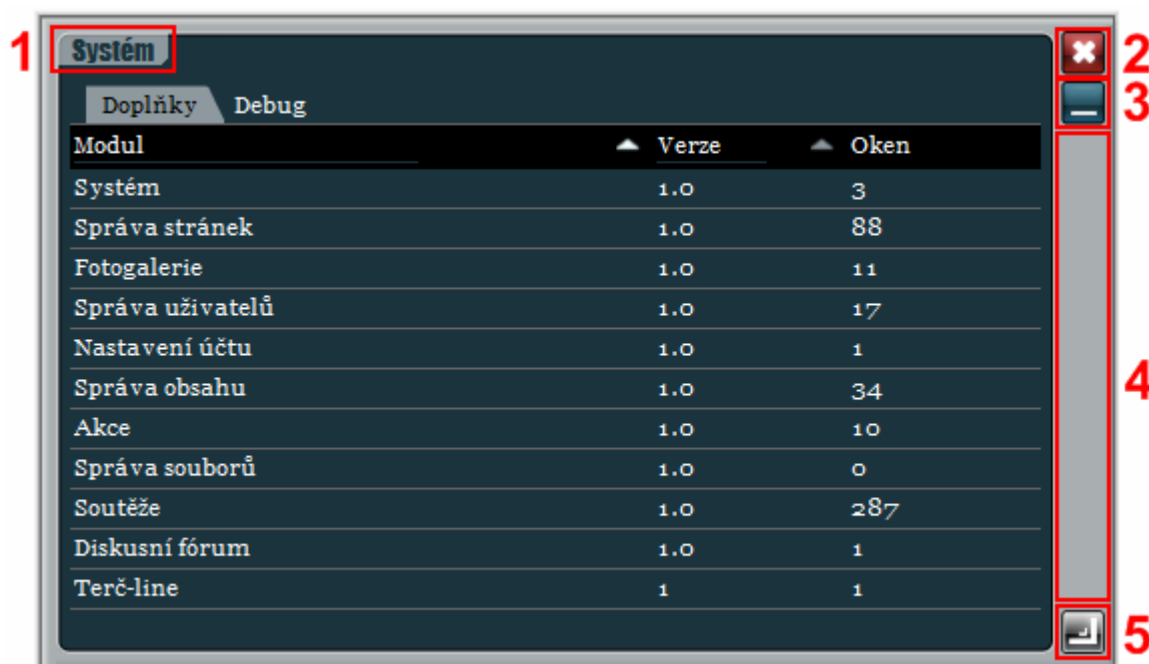
K hlavnímu panelu se vztahuje databázová tabulka *icons*. Obsahuje nastavení a řazení zástupců jednotlivých uživatelů. Její konfigurace je uvedena na obrázku 3.3.

Jméno	Typ	Další vlastnosti	Komentář
Primární index	id	unique	
unique	user,window	unique	
id	int(10) unsigned	auto_increment	ID nastavení
user	int(10) unsigned		ID uzivatele
window	int(10) unsigned		ID systemoveho okna
sort	int(10) unsigned		poradi zastupce

Obrázek 3.3 – databázová tabulka *icons*

3.1.2 Systémové okno

Systémové okno je základním stavebním elementem celého systému iSys. Svými vlastnostmi a chováním se podobá klasickým oknům operačních systémů PC. Stejně jako v OS Windows jej můžeme otevřít, přesunout, změnit jeho velikost, minimalizovat nebo zavřít. Popis okna je ukázán na obrázku 3.4.

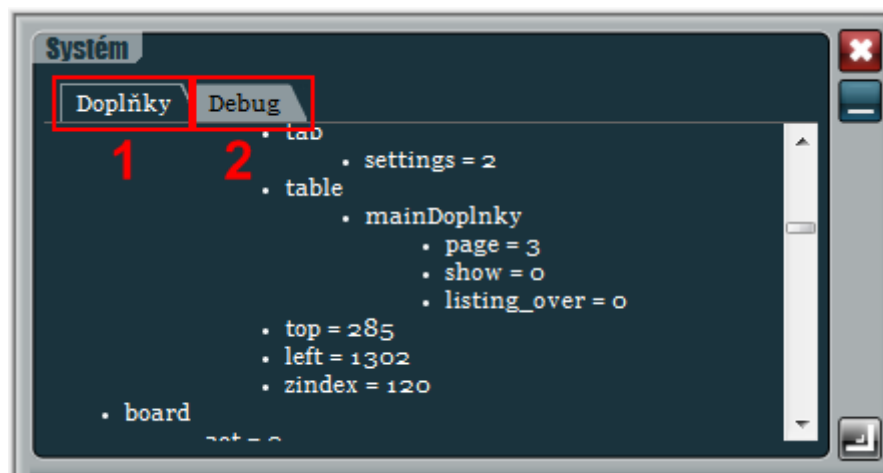


Obrázek 3.4 – systémové okno iSys

- 1) název okna
- 2) tlačítko zavřít
- 3) tlačítko minimalizovat
- 4) lišta k uchycení a přesunutí okna (metodou Drag & Drop)
- 5) tlačítko ke změně velikosti okna (metodou Drag & Drop)

3.1.2.1 Záložky

Smyslem záložek je umístění většího množství informací do jednoho systémového okna. K vytváření záložek slouží třída *TAB* uložená v souboru *system/classes/class_tab.php*. Stejně jako třídy modulů, tak i všechny třídy generující uživatelské rozhraní dědí metody a vlastnosti z třídy *WINDOW*. Příklad aktivní (2) a neaktivní (1) záložky je ukázán na obrázku 3.5.



Obrázek 3.5 – aktivní (2) a neaktivní (1) záložka

Třída *TAB* obsahuje následující metody:

`function __construct($id, $name, $width="")` – konstruktor třídy,
`id` – ID okna, `name` – ID skupiny záložek – jedinečné v rámci okna, `width` – určuje šířku záložek, pokud se parametr vypustí, záložky budou mít plnou šířku

`function addTab($id,$name)` – přidá novou záložku, `id` – id záložky – jedinečné v rámci skupiny záložek, `id` záložky zároveň určuje jméno metody, která generuje její obsah, `name` – jméno záložky

`function endTab()` – ukončí skupinu záložek

V každém okně je možné inicializovat libovolný počet skupin záložek. Přechod mezi záložkami zajišťuje tzv. scroll efekt. Po kliknutí na neaktivní záložku vyše prohlížeč (AJAX) dotaz na server, kde se ověří práva uživatele a je-li ověření v pořádku, odešle se zpět obsah žádané záložky. Jakmile se obsah vloží do skryté záložky, dojde k animaci záložek a zobrazení nového obsahu. Příklad kódu záložek z obrázku 3.5 v modulu *SYSTEM*:

```
$k = new TAB($this->id, "settings");  
$k->addTab("mainDoplňky", "Doplňky");  
$k->addTab("debug", "Debug");  
$k->endTab();
```

3.1.2.2 Tabulky

Tabulky slouží k výpisu uložených dat. Základní funkce tabulky jsou:

- vyhledávání
- seřazování
- stránkování

Ukázka výpisu systémových modulů v tabulce je na obrázku 3.6. U každého sloupce v tabulce lze povolit vyhledávání a seřazování. Podmínkou je načítání hodnot přímo z databáze MySQL, protože třízení a vyhledávání je implementováno do SQL dotazu. Seřazovat lze sloupce kliknutím na ikonu 1 (obrázek 3.6). Nadpis sloupců ve kterých je povoleno vyhledávání je podtržen. Klepnutím na tento nadpis se uživateli aktivuje kurzor s textovým polem, do kterého se zadává hledaný text. Smazáním textu se vyhledávání zruší. Jestliže se záznamy nevejí na jednu stránku tabulky, umístí se pod tabulku stránkování

(3 a 4 na obrázku 3.6). Přejít je možné po jedné stránce pomocí šipek (4) nebo kliknutím na číslo požadované stránky (3), kdy jsou případné mezistránky během animace přechodu vynechány. Aktualizace a načítání obsahu tabulky během seřazování, vyhledávání a stránkování je prováděno stejně jako u záložek prostřednictvím AJAXu.



Obrázek 3.6 – tabulka s možnostmi třídit (1), vyhledávat (2) a stránkovat (3 a 4)

Metody třídy *TABLE*:

`function __construct($id, $name, $o_height = 0, $o_width = 0, $dynamic = 1, $line_height = 20)` – konstruktor třídy, `id` – ID okna, které obsahuje tabulku, `name` – ID tabulky (musí se shodovat se jménem metody, ve které je tabulka zapsána), `o_height` – výchozí výška tabulky (0 – plná výška v okně, kladná hodnota - nastaví výšku v px, záporná hodnota - udá rozdíl mezi výškou tabulky a okna), `o_width` – výchozí šířka tabulky, `dynamic` – změna velikosti tabulky spolu s oknem (0/1), `line_height` – výška řádku [px]

`function where($add = "")` – vrací část MySQL dotazu k filtrování záznamů, `add` – upřesňující podmínky

`function limit()` – vrací část MySQL dotazu určující stránku

`function sort()` – vrací část MySQL dotazu určující řazení

`function addColumn($id, $name, $width=0, $search=0, $sort=0, $show="")` – přidává sloupec do tabulky, `id` – ID sloupce, `name` – nadpis sloupce, `width` – šířka sloupce (zadáva se včetně jednotek [px/%]), `search` – aktivuje vyhledávání ve sloupci (0/1), `sort` – aktivuje řazení ve sloupci (0/1), `how` – nastaví výchozí řazení záznamů v tabulce podle tohoto sloupce (ASC - vzestupně/DESC - sestupně)

`function startTable($cnt)` – začátek tabulky – vykreslí hlavičku, `cnt` – celkový počet záznamů v tabulce

`function addLine($line)` – vloží řádek do tabulky, `line` – pole hodnot sloupců

`function endTable()` – ukončí tabulku

`function setParam($key="", $val="")` – slouží k uložení proměnné

`function getParam($key="")` – slouží k načtení proměnné

Z důvodu volání a načítání obsahu tabulky z AJAXu, musí být kód tabulky zapsán ve vlastní metodě a název metody se musí shodovat s názvem tabulky v konstruktoru třídy *TABLE*. Uvedu zde proto celou metodu, která vykresluje tabulku „mainDoplňky“ z obrázku 3.6:

```
function mainDoplňky() {  
    global $db;
```



```
// nová tabulka
$t = new TABLE($this->id,"mainDoplňky",-25);

// přidání sloupců
$t->addColumn("title_" . $_SESSION['cfg']['lg'],
    $this->lg[802], "60%", 1, 1);
$t->addColumn("version", $this->lg[803], "20%", 0, 0);
$t->addColumn("id", $this->lg[804], "20%", 1, 1);

// celkový počet záznamů
$cnt = $db->countRows("SELECT * FROM modules".
    $t->where());

// začátek tabulky
$t->startTable($cnt);

// načtení záznamů z databáze
$temp = $db->all("SELECT * FROM modules" . $t->where() .
    $t->sort() . $t->limit());

if ($temp)
    // pro každý záznam se vloží řádek do tabulky
    foreach ($temp as $line){
        $cnt = $db->one("SELECT COUNT(id) FROM windows
            WHERE id_module = '" . $line['id'] . "'");

        $t->addLine(array($line["title_" . $_SESSION['cfg']['lg']],
            $line["version"], $cnt));
    }

$t->endTable();
}
```

Třída *TABLE* obsahuje několik proměnných. Zmínil bych jen jeden důležitý:
show. Podle tří možných hodnot určuje způsob vykreslení:

- 0 – vykreslí celou tabulku
- 1 – vykreslí jen řádky tabulky (stránkování)
- 2 – vykreslí řádky tabulky a stránkování (vyhledávání a řazení)

Proměnná *show* je nastavována vždy z AJAXu jakmile uživatel použije některou funkci tabulky. Následně je zaslán požadavek na vykreslení (obnovu) jejího obsahu.

3.1.2.3 Formuláře

Formuláře slouží k ukládání, změně nebo odstraňování záznamů. Práci s formuláři zajišťují v systému iSys dvě třídy. Za prvé je to třída *FORM*, která vykresluje jednotlivé prvky formuláře a za druhé třída *CHECK*, která kontroluje správnost odeslaných údajů.

Některé metody třídy *FORM*:

`function __construct($id)` – konstruktor třídy, `id` – ID okna, které obsahuje formulář

`function startForm($form, $data="")` – začátek formuláře, `form` – ID formuláře (jedinečné v rámci okna), `data` – předáním hodnoty "data" lze formulářem odesílat soubory (metoda *file*)

`function startTableForm($form)` – je nutné volat, pokud tabulka vytvořená třídou *TABLE* obsahuje formulářová pole a začátek formuláře je umístěn v jiné metodě, `form` – ID již dříve vytvořeného formuláře

`function file($t, $class='', $value2='', $help='')` – vykreslí vstupní pole pro upload souboru, `t` – ID pole (jedinečné v rámci formuláře), `class` – CSS třída pole, `value2` – výchozí hodnota pole, `help` – nápověda, která se zobrazí po kliknutí do pole

`function text($t, $class='', $value2='', $help='', $attr='')` – vykreslí jednořádkové textové vstupní pole, `t` – ID pole, `class` – CSS třída, `value2` – výchozí hodnota, `help` – nápověda, `attr` – další atributy vstupního pole

`function textarea($t, $class='', $width='', $height='', $value2='', $help='')` – vykreslí víceřádkové textové pole, stejné

parametry jako metoda `text`, navíc je parametr `width` (šířka v px) a `height` (výška v px)

`function tiny($t,$class='', $width='', $height='', $value2='', $help='')` – vykreslení wysiwyg editoru TinyMCE (kapitola 3.1.2.4.), parametry shodné s metodou `textarea`

`function pass($t,$class='', $value2='', $help='')` – vykreslí vstupní textové pole s hvězdičkami (použití u hesel), shodné parametry s metodou `text`

`function select($t,$class='', $values = array(), $value="", $help='')` – vykreslí roletkové menu s možností výběru, `t` – ID pole, `class` – CSS třída, `values` – pole hodnot nabídky, `value` – výchozí hodnota, `help` – nápověda

`function checkbox($t, $value="", $attr="")` – vykreslí zaškrtačací políčko, `t` – ID pole, `value` – výchozí hodnota (0/1), doplňující atributy

`function hidden($t,$value='')` – vloží do formuláře skryté pole, `t` – ID pole, `value` – hodnota pole

`function submit($t,$value, $class="")` – vykreslí tlačítko pro odeslání formuláře, `t` – ID pole, `value` – popisek tlačítka, `class` – CSS třída

`function getSorterDown($value)` – vrátí ikonu k seřazování – zvýší pořadí položky o 1, `value` – ID položky, která má být posunuta

`function getSorterUp($value)` – vrátí ikonu k seřazování – sníží pořadí položky o 1, `value` – ID položky, která má být posunuta

function endForm() – ukončení formuláře

Jakmile je formulář odeslán, dochází k načtení příslušného *post* souboru modulu a zpracování odeslaných údajů. Ke zpracování je možné použít třídu *CHECK*. O chybě ve formuláři rozhoduje proměnná *err* třídy *CHECK*. Po vytvoření instance je nastavena na nulu a veškeré chybové texty formuláře jsou smazány.

Příklad jednoduchého formuláře z modulu *user_manager* k vytvoření nového uživatelského účtu:

```
function add(){  
  
    // nový formulář, v proměnné $this->id je uloženo ID  
    // aktuálního okna  
    $w = new FORM($this->id);  
    // začátek formuláře s ID "adduser"  
    $w->startForm("adduser", "data");  
    echo "<div class='window-outer-form'>";  
    // pole $this->lg obsahuje texty načtené z jazykového  
    // souboru - vypíše popisek pole  
    echo $this->lg[128]."<br />";  
    // vloží vstupní textové pole označené s ID "login"  
    $w->text("login", "login", "", $this->lg[301]);  
    echo "<br /><br />".$this->lg[129]."<br />";  
    $w->text("name", "login", "");  
    echo "<br /><br />".$this->lg[130]."<br />";  
    $w->text("surname", "login", "");  
    echo "<br /><br />".$this->lg[131]."<br />";  
    $w->text("email", "login", "", $this->lg[302]);  
    echo "<br /><br />".$this->lg[132]."<br />";  
    $w->select("group", "login2", getGroup(), "");  
    echo "<br /><br />".$this->lg[133]."<br />";  
    // vloží menu s výběrem jazyka, přednastaví aktuálně  
    // používaný jazyk  
    $w->select("lang", "login2", getLang(),  
        $_SESSION['cfg']['lg']);  
    echo "<br /><br />".$this->lg[134]."<br />";  
    // vloží pole pro upload souboru (fotku uživatele)  
    $w->file("photo", "win-photo", "", $this->lg[303]);  
    echo "<br /><br />";  
    echo "</div>";  
    // vloží tlačítko odeslání  
    $w->submit("ok", $this->lg[135]);  
}
```

```
// konec formuláře  
$w->endForm();  
  
}
```

Výše zmíněný formulář zpracovává *post* skript modulu *user_manager* následovně:

```
// nová kontrola formuláře s ID "adduser"  
$ch = new CHECK("adduser");  
  
// kontroluje minimální délku 3 znaků pole s ID "login"  
$ch->minLen("login",3);  
// kontroluje zda se již položka nenachází v databázi  
$ch->isNotInDB("login","users","nick");  
// kontroluje přenos souboru  
$ch->file("photo","data/user-photo/",  
        "gif,jpg,jpeg,png",0);  
  
if (!$ch->err){  
    // formulář je bez chyby, dojde k vytvoření nového  
    účtu  
    ...  
}
```

Třída *CHECK* má mimo kontroly vstupních polí i několik dalších funkcí. Jelikož dochází v průběhu editace formulářů k jejich ukládání na server do *\$_SESSION*, mohou být data z formuláře načtena i odsud metodou *getValues*. Dále zajišťuje řazení položek v tabulce. Přehled a popis některých metod třídy *CHECK* je zde:

function isNumeric(\$item) – kontroluje, zda je ve vstupním poli zadané číslo, *item* – ID vstupního pole

function isInDB(\$item,\$table,\$column) – kontroluje, zda se zadaná položka nachází už v databázi, *item* – ID vstupního pole, *table* – jméno databázové tabulky, *column* – jméno sloupce v databázové tabulce

`function minLen($item,$minlen)` – kontroluje minimální počet znaků,
`item` – ID vstupního pole, `minlen` – minimální počet znaků

`function isEmail($item)` – kontroluje, zda je zadaný řetězec platnou
emailovou adresou, `item` – ID vstupního pole

`function file($item,$dir="", $exts = "", $mb=0)` – zajišťuje
přenos souborů, `ID` vstupního pole, `dir` – relativní cesta k adresáři k uložení souboru,
`exts` – povolené přípony souboru – oddělují se čárkami, `mb` – povinnost vložení
souboru (0/1)

`function getValues($filter="")` – vrátí pole hodnot formuláře uložených
v `$_SESSION` (tj. ty, které byly změněny), `filter` – omezuje výběr hodnot podle ID
vstupního pole

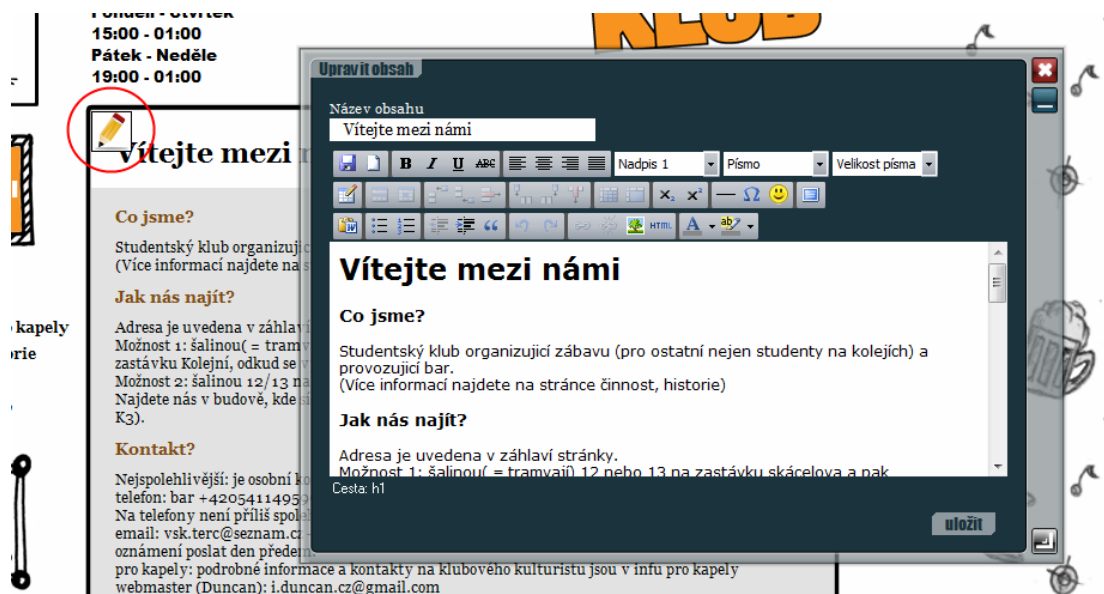
`function sorter($table, $wh = "")` – zajišťuje řazení položek uložených v databázi,
`table` – jméno databázové tabulky, `wh` – upřesňující podmínka MySQL dotazu

3.1.2.4 wysiwyg editor TinyMCE

TinyMCE je jeden z nejvíce používaných volně dostupných webových wysiwyg editorů textu. Zkratka wysiwyg pochází z anglického výrazu: What you see is what you get. V češtině můžeme použít označení vizuální editor textu. Uživatelům dovoluje editovat a formátovat text podobným stylem, jako ve standardním textovém editoru. Prostředí editoru TinyMCE a funkce lze upravovat libovolně. Dostupná je i česká lokalizace. Editor a zdrojové kódy jsou napsány v jazyce JavaScript. Ukázkou editoru lze vidět na obrázku 3.7. Domovská stránka je <http://tinymce.moxiecode.com/>.

3.1.3 Pozadí pracovní plochy

Na pozadí pracovní plochy je umístěna spravovaná webová stránka, kterou je možné libovolně procházet stejně jako bychom přihlášení nebyli. Pokud ale jsme, na vybraných místech jsou umístěny ikony úpravy obsahu (obrázek 3.4). Po kliknutí na ikonu se otevře okno a obsah můžeme snadno měnit.



Obrázek 3.7 – úprava obsahu editorem TinyMCE přímo z internetové stránky

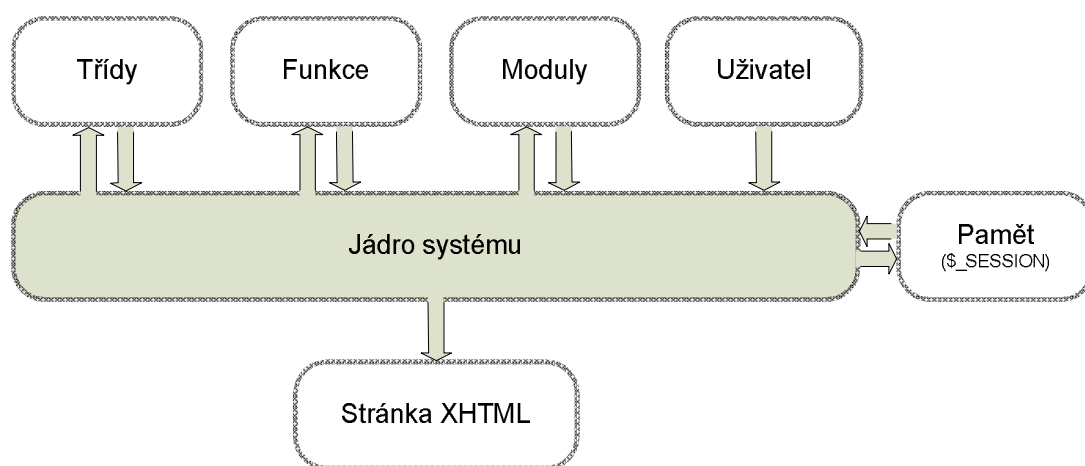
3.2 JÁDRO SYSTÉMU

Jádro systému zajišťuje základní funkce a provoz systému. Jeho hlavní činnosti jsou :

- Autentizace uživatele a přidělení práv
- Správa uživatelského rozhraní (hlavní panel a systémová okna)
- Komunikace s klientem prostřednictvím AJAXu
- Práce s moduly

Na obrázku 3.5 je blokově zobrazeno schéma iSys systému. Ve skutečnosti jsou ale bloky daleko více provázány. Třídy a funkce poskytují jádru nástroje

k generování obsahu. Těchto nástrojů využívají moduly, které obstarávají obsah systémových oken a rozšiřující funkce systému. Podrobně jsou všechny moduly popsány v kapitole 3.3. Dočasná data jsou ukládána v PHP do pole `$_SESSION` (kapitola 3.2.2). Dle požadavků uživatele se pak generuje obsah XHTML stránky.



Obrázek 3.8 – blokové schéma systému

3.2.1 Zdrojový kód

Jádro systému tvoří soubor funkcí uložených v souboru `system/main/window.php`. Ve zkratce zde uvedu několik funkcí a jejich popis:

`function wMainbar()` – vykreslí hlavní panel. Funkce je volána vždy od klienta technikou AJAX a to vždy po načtení stránky a nebo po otevření nebo zavření okna.

`function wSys()` – vykreslí uživatelské prostředí. Volá se přímo z XHTML šablony – vkládá se na začátek dokumentu, hned za značku `<BODY>`.

`function wInit()` – vypíše do stránky kód JavaScriptu, který po načtení stránky inicializuje uživatelské prostředí. Volá se přímo z XHTML šablony – vkládá se na konec dokumentu, hned před značku `</BODY>`.

`function wOpen($id)` – otevře systémové okno podle zvoleného ID.

`function wClose($id)` – zavře systémové okno

`function wLoad($id)` – vykreslí šablonu systémového okna a nahraje jeho obsah

`function cLoad($id, $fc2="")` – nahraje obsah okna nebo jen jeho část, `fc2` – určuje jaká metoda modulu se má vykonat

`function rLoad()` – načte informace o uživateli a přiřadí práva

3.2.2 Paměť

Dočasnou paměť systému představují relace - session v PHP. Session pracují na jednoduchém principu. Každému uživateli, který vstoupí na stránku, je přidělen identifikátor – session ID (SID). PHP potom implicitně předává SID ve všech odkazech (buďto jako parametr v URL adrese nebo uloží SID do cookie). Jakmile je relace vytvořena (provede se příkazem `session_start()`), je zpřístupněná superglobální proměnná `$_SESSION`. K ní je možné v jakémkoliv místě skriptu přistupovat a ukládat řetězce. Data ze `$_SESSION` jsou ukládána na serveru, mezi uživatelem a serverem se přenáší pouze SID. Tímto způsobem řeší PHP bezstavovost protokolu HTTP [10].

V rámci systému je definováno několik proměnných v poli `$_SESSION`. Jejich přehled je v tabulce 3.1. Například jméno uživatele je uloženo v proměnné `$_SESSION['user']['name']`. Vlastní strukturu v relaci mají i systémová okna. Pro každé okno je vytvořeno pole `$_SESSION['window']['ID_okna']`. Přehled atributů systémových oken je v tabulce 3.2.

klíč \$_SESSION	popis proměnné
user, id	ID přihlášeného uživatele
user, name	jméno přihlášeného uživatele
user, surname	příjmení přihlášeného uživatele
user, nick	přihlašovací jméno
user, email	email
user, photo	cesta k souboru fotografie uživatele
user, logged	uživatel přihlášen (0/1)
user, group	ID uživatelské skupiny přihlášeného uživatele
cfg, lg	nastavený jazyk systému (cz, en, ..)
cfg, lgs	pole všech dostupných jazyků systému
sys	pole nastavení systému (databázová tabulka <i>system</i>)
board, aot	nastavení hlavního panelu vždy navrchu (0/1)
board, width	šířka hlavního panelu v px
board, zindex	zindex (atribut CSS) hlavního panelu
settings, tooltip	zobrazování nápovědy (0/1)
settings, showerror	zobrazování chyb (0/1)
path	pole všech existujících cest webových stránek
rights	pole obsahující všechna ID systémových oken, ke kterým má uživatel povolen přístup

Tabulka 3.1 – přehled proměnných systému v poli \$_SESSION



klíč \$_SESSION ['window']['ID_okna']	popis proměnné
icon-sorting	pořadí zástupce okna na hlavním panelu
show	okno je otevřeno (0/1)
min_width	minimální a výchozí šířka okna [px]
min_height	minimální a výchozí výška okna [px]
width	aktuální šířka okna [px]
height	aktuální výška okna [px]
title	název okna
visible	okno je maximalizované (0/1)
resize	je možné měnit velikost okna (0/1)
top	souřadnice umístění okna na ploše – odsazení od horního okraje [px]
left	souřadnice umístění okna na ploše – odsazení od levého okraje [px]
zindex	překrývání oken – určuje pořadí zobrazení oken na ploše (atribut CSS)
tab	pole určené pro proměnné záložek, které okno obsahuje
table	pole určené pro proměnné tabulek

form	pole určené pro proměnné formulářů
------	------------------------------------







Tabulka 3.2 – přehled atributů systémových oken ukládaných do \$_SESSION

K trvalému uložení dat slouží databáze MySQL. K jádru patří následujících 5 tabulek:

- *icons* – obsahuje seznam a pořadí zástupců všech uživatelů (kapitola 3.1.1)
- *languages* – seznam všech dostupných jazykových lokalizací systému
- *modules* – seznam dostupných modulů (obrázek 3.10)
- *system* – konfigurace systému
- *windows* – seznam všech systémových oken, které je možné v systému otevřít a jejich konfigurace (obrázek 3.9)

Jméno	Typ	Další vlastnosti	Komentář
 Primární index	id	unique	
 module	id_module		
 id	int(10) unsigned	auto_increment	ID okna
 width	int(5) unsigned		minimalni a vychozi sirka okna [px]
 height	int(5) unsigned		minimalni a vychozi vyska okna [px]
 resize	int(1)		povolena zmena velikosti okna 0/1
 max_width	int(5) unsigned		maximalni sirka okna [px]
 max_height	int(5) unsigned		maximalni vyska okna [px]
 sico	int(1)		vzdy zobrazovat zastupce na hlavnim panelu 0/1
 rights_logout	int(1)		prava - povolen pristup vsem odhlasenym uzivatelum 0/1
 rights_login	int(1)		prava - povolen pristup vsem prihlasenym uzivatelum 0/1
 mainbar	int(1)		povolit vytvorit zastupce na hlavnim panelu 0/1
 title_cz	varchar(50)		nazev okna v cestine
 title_en	varchar(50)		nazev okna v anglictine
 ico	varchar(50)		nazev ikony ktera se pouzije pro zastupce
 module	varchar(50)		prikaz urcujici obsah okna
 id_module	int(5) unsigned		ID modulu, ke kteremu okno patri

Obrázek 3.9 – databázová tabulka *windows*

Jméno	Typ	Další vlastnosti	Komentář
 Primární index	id	unique	
 id	int(5) unsigned	auto_increment	ID modulu
 title_cz	varchar(50)		jmeno modulu
 title_en	varchar(50)		
 version	varchar(50)		verze
 module	varchar(50)		ID2

Obrázek 3.10 – databázová tabulka *modules*

3.2.3 Zpracování stránky

Inernetový prohlížeč komunikuje se serverem prostřednictvím dvou skriptů, které jsou umístěny v domovském adresáři:

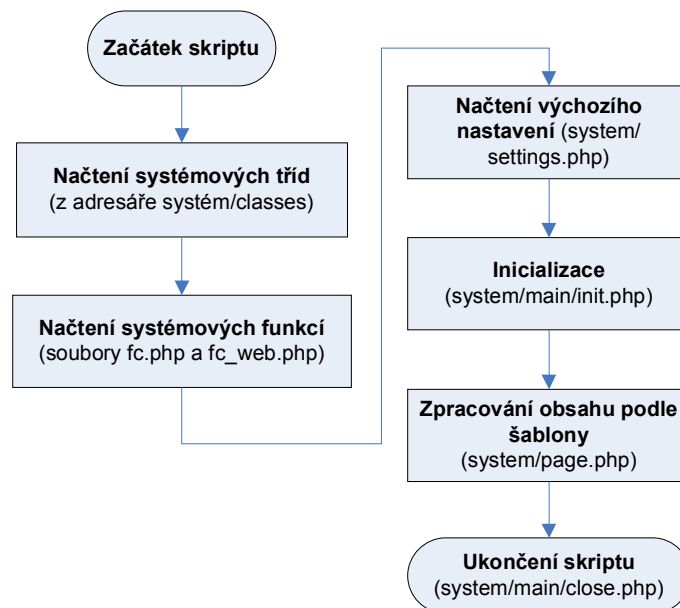
- `index.php` – na tento skript jsou přesměrovány (souborem `.htaccess`) všechny stránky internetové prezentace, generuje celé stránky
- `window_ajax.php` – slouží výhradně ke komunikaci s AJAXem

3.2.3.1 Generování celé stránky (*index.php*)

Vývojový diagram průběhu zpracování celé stránky je znázorněn na obrázku 3.11. Po načtení systémových tříd, funkcí a výchozího nastavení dojde k inicializaci. Součástí inicializace jsou tyto procesy:

- vytvoření relace, připojení k databázi
- načtení jazykového souboru – lokalizace systému
- načtení funkcí modulů (soubory modulů `*.fc.php`)
- načtení nastavení systému z databáze
- autorizace uživatele a přiřazení práv
- vytvoření navigačních cest
- pokud byl odeslán formulář, načíst post soubor (soubor končící `*.post.php`) příslušného modulu

Po inicializaci, podle šablony uložené v souboru `system/page.php`, dojde ke generování obsahu stránky. Následně je ukončeno připojení k databázi s stránka odeslána k uživateli.



Obrázek 3.11 – vývojový diagram zpracování celé stránky

3.2.3.2 Načítání obsahu prostřednictvím AJAXu (*window_ajax.php*)

K dynamickému načítání obsahu stránky je určena funkce `wAJAX()` uložena v souboru `window.js`. Podle požadavků uživatele odesílá na server dotazy, které zpracovává skript `window_ajax.php`. Dotazy jsou několika typů:

- uložení proměnné na server (do `$_SESSION['window']` nebo `$_SESSION['board']` – změna velikosti okna, posunutí apod.), příklad minimalizování okna s ID 3 (vlastnost `visible` je nastavena na nulu):

```
wAJAX('set','window.3.visible:0');
```

První parametr určuje typ dotazu, v druhém parametru je předán příkaz: název proměnné a její hodnota. Jednotlivé klíče se oddělují tečkami, hodnota dvojtečkou. Předání více proměnných v jednom dotazu je možné spojením více příkazů řetězcem: „%%“.

- obnova hlavního panel (aktualizuje zástupce na hlavním panelu) – provede se příkazem:

```
wAJAX('wRefresh','desktop');
```

- načtení obsahu okna nebo jeho části. Použito především při otevírání oken a dynamické změně obsahu (záložky a tabulky). Následující příkaz provede načtení obsahu a otevření okna s ID 5:

```
wAJAX ("wOpen" , 5) ;
```

- zavření okna

Příklad zavření okna s ID 5:

```
wAJAX ("wClose" , 5) ;
```

Funkce `wAJAX()` dovoluje i komplexnější volání, kdy je možné uložit proměnné na server a po uložení obnovit určitý obsah.

3.3 MODULY

Moduly rozšiřují základní funkce systému. Rozdělit je můžeme na systémové (systém bez nich není schopen provozu) a doplňkové (mohou být odstraněny). Systémové moduly jsou:

- Systém
- Správa uživatelů
- Správa účtu

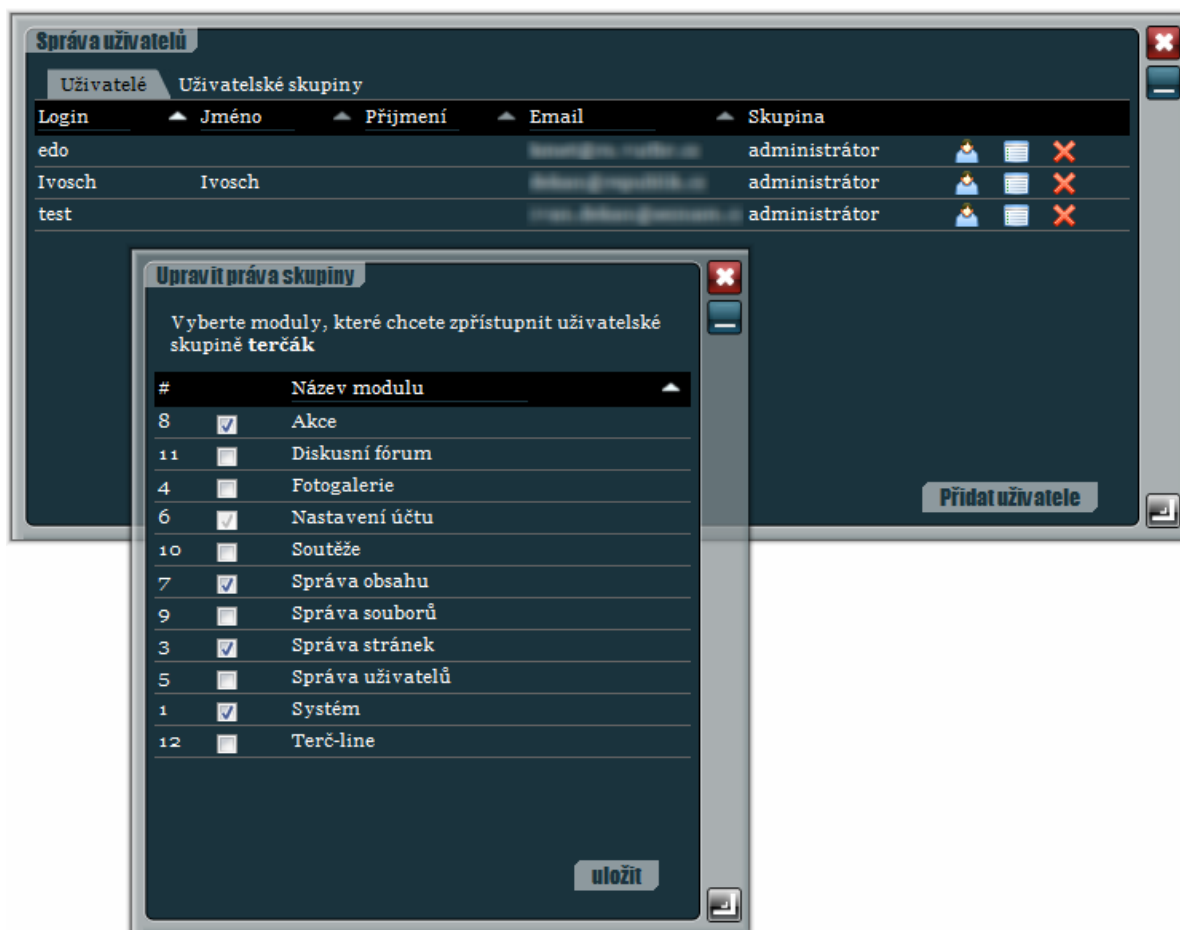
Každý modul je zapsán v tabulce *modules* (obrázek 3.10) a má uloženy potřebné záznamy v tabulce *windows*. Zároveň vlastní identifikátor - například „system“ u modulu *Systém*. Základ tohoto modulu představuje třída *SYSTEM*, jejíž název je odvozený od identifikátoru a je zapsán velkými písmeny. Třída je uložena v souboru *system.php* a vlastním adresáři *modules/system*. Další částí modulu je post skript (*system.post.php*), který se zpracovává (načítá) po odeslání jakéhokoliv formuláře modulu *SYSTEM*. Posledním souborem je *system.fc.php*. Zde se umísťují rozšiřující funkce modulu pro všeobecné použití. Soubory **.post.php* a **.fc.php* nejsou povinné a modul je nemusí obsahovat.

3.3.1 Systém

Jde o malý modul zajišťující přihlášení a odhlášení uživatelů. Dále obsahuje jedno systémové okno informující o doplňcích a stavu systémových proměnných uložených v `$_SESSION`. K modulu *Systém* patří databázová tabulka *system*, která má pouze dva sloupce (*key* a *value*) k uložení názvu proměnné a její hodnoty. V současném stavu obsahuje pouze tzv. *key* systému určený ke kódování hesel.

3.3.2 Správa uživatelů

Modulu dovoluje spravovat uživatelské účty. Na první záložce lze vytvářet nové, upravovat a odstraňovat stávající. Podmínkou je, aby existoval alespoň jeden administrátorský účet. Na druhé se nachází správa uživatelských skupin. Uživatelská



Obrázek 3.12 – správa uživatelských účtů a úprava práv uživatelské skupiny

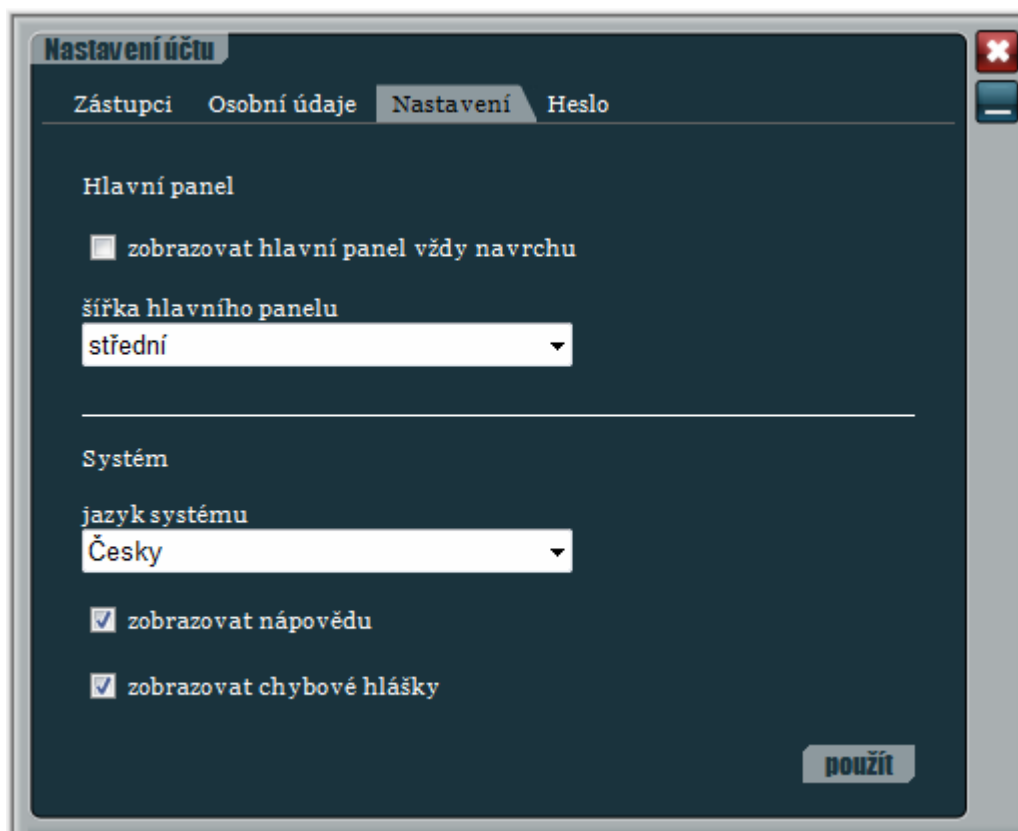
skupina „administrátor“ je stálá a nelze ji odstranit. Uživatelům této skupiny zpřístupňuje všechny moduly. Práva můžeme přiřadit až dalším uživatelským skupinám.

Data modulu jsou uložena celkem ve třech tabulkách:

- *users* – informace o uživateli a jeho účtu
- *groups* – uživatelské skupiny
- *rights* – práva uživatelských skupin

3.3.3 Správa účtu

Každému uživateli je dovoleno ve Správě účtu upravovat osobní informace, měnit si heslo a umisťovat zástupce na hlavní panel. Modul obsahuje i nastavení systému jako je například jazyk rozhraní (obrázek 3.13).



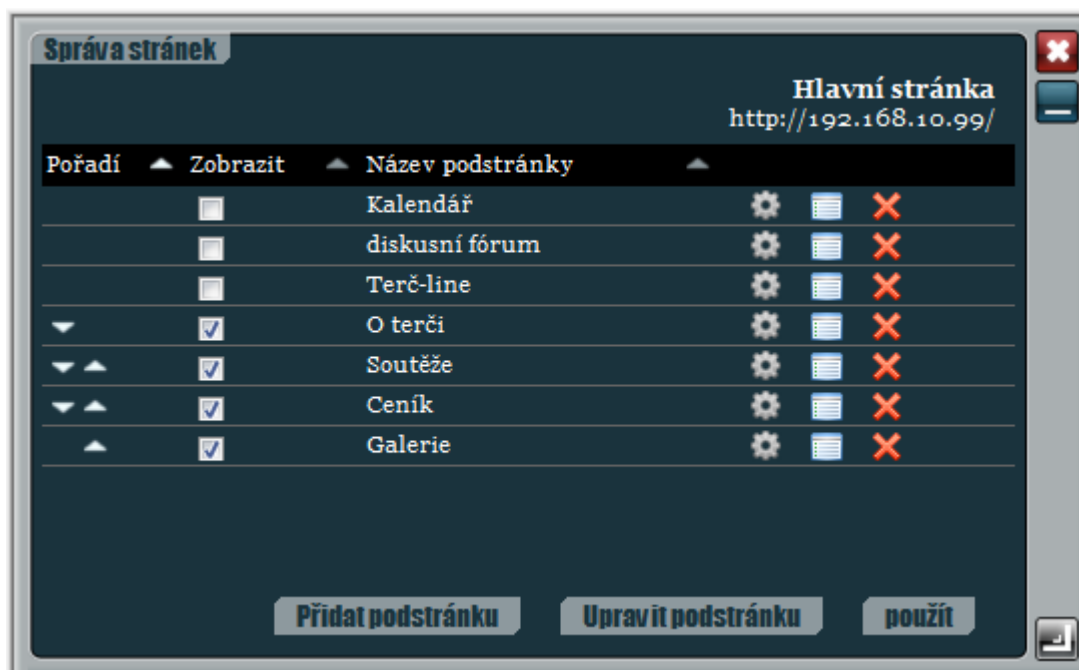
Obrázek 3.13 – nastavení uživatelského účtu

3.3.4 Diskusní fórum

Modul diskusního fóra neobsahuje žádná systémová okna. Jeho hlavní část je obsažena ve funkčním souboru *forum.fc.php*. Jedná se o funkci *forum()*, která vloží do internetové stránky celé diskusní fórum. Témata diskusního fóra jsou ukládána do databázové tabulky *forum* a jednotlivé příspěvky do *forum_data*. K vkládání příspěvků je využito wysiwyg editoru TinyMCE v omezené konfiguraci. Vkládání je u nepřihlášených uživatelů chráněno antispamovou ochranou.

3.3.5 Správa stránek

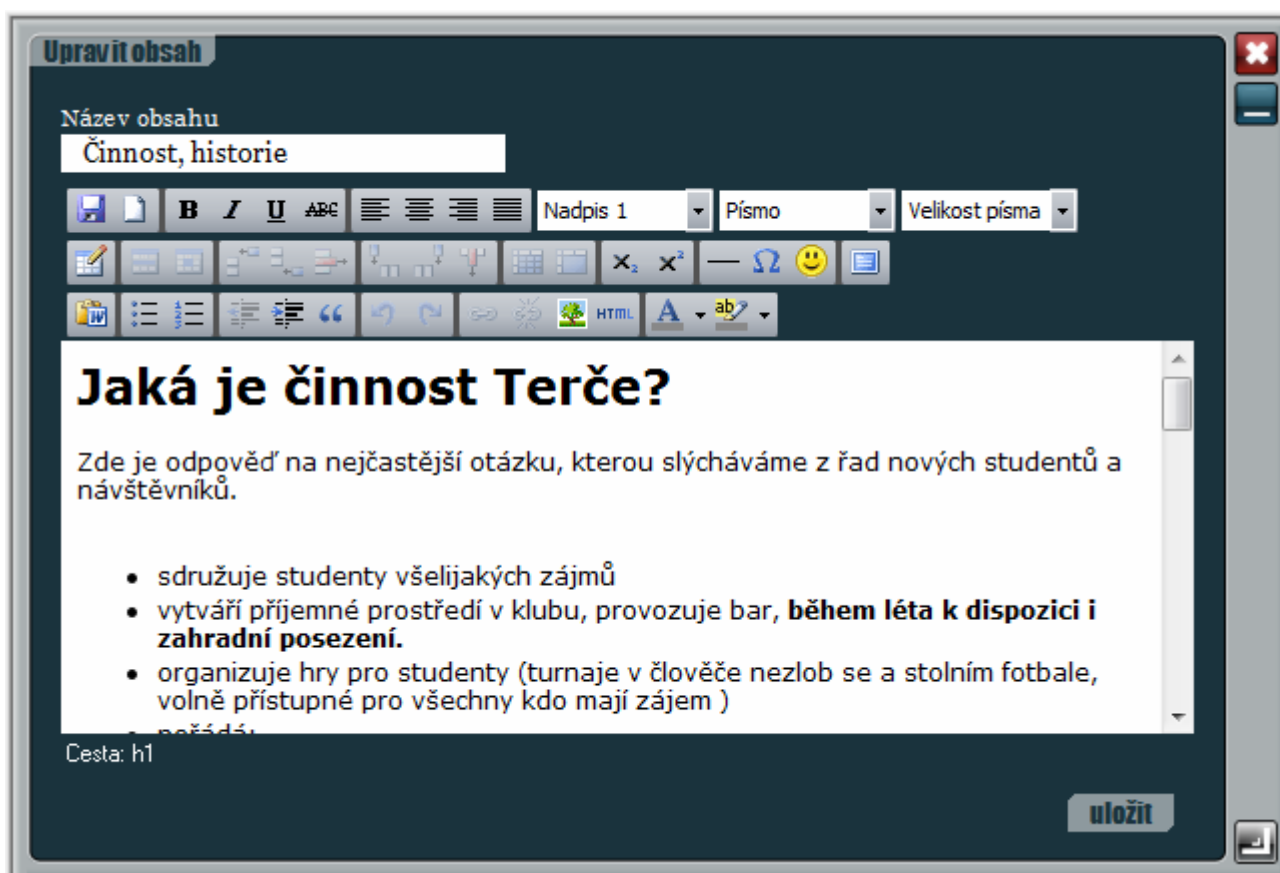
K úpravě struktury stránek slouží modul *Správa stránek*. Dokáže vytvářet nové stránky a podstránky, měnit jejich viditelnost a pořadí a odstraňovat je. Zároveň zde najdeme editor zdrojového kódu stránek. Struktura stránek se ukládá do tabulky *content*. Úprava struktury webových stránek je ukázaná na obrázku 3.14.



Obrázek 3.14 – úprava struktury webových stránek

3.3.6 Správa obsahu

Tento jednoduchý modul má za úkol usnadnit úpravu samotného obsahu webových stránek. Vytvořit, editovat a smazat je možné libovolný počet obsahů. Úprava je prováděna v editoru TinyMCE s bohatým výběrem funkcí (obrázek 3.15). Vložení už vytvořeného obsahu do stránky se provádí v editoru zdrojového kódu stránky (modul *Správa stránek*). Ukládání obsahů probíhá do tabulky *content2*.

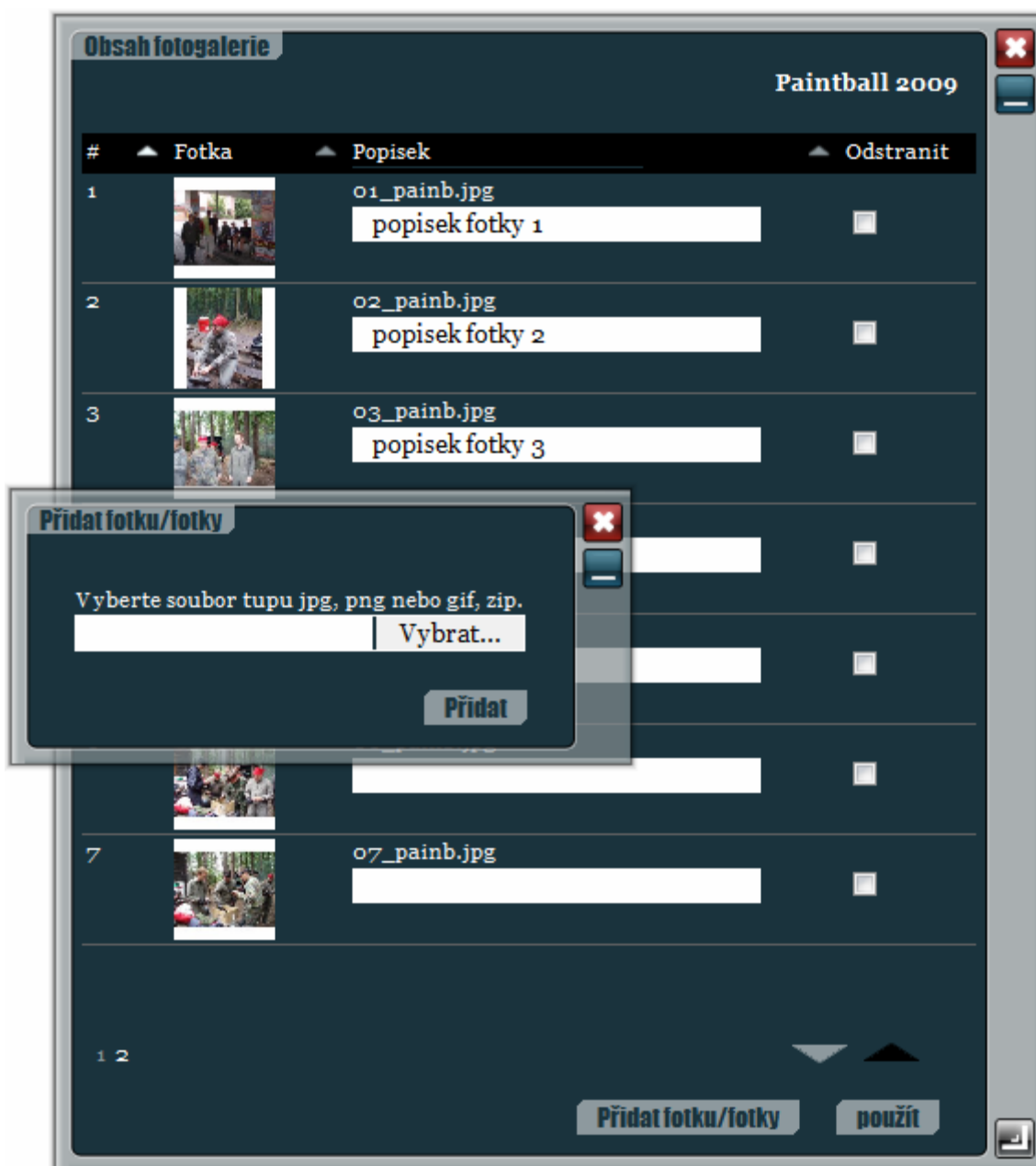


Obrázek 3.15 – úprava obsahu

3.3.7 Fotogalerie

Modul *Fotogalerie* dovoluje správu a prezentaci fotek na webových stránkách. Ke každé nové fotogalerii je přiřazeno ID a datum. Podle datumu lze fotogalerii při procházení třídit. Vkládání nových fotek do galerie se provádí formulářem. Povolené typy souboru jsou png, jpg, gif a nebo zip obsahující fotky.

Fotky jsou na serveru ukládány do *files/photogallery/ID_galerie*. Do složky *files/photogallery* je nutné mít povolený zápis. Seznam fotogalerií je uložen v tabulce *photogallery*, informace o fotografiích v tabulce *photogallery_data*.



Obrázek 3.16 – úprava obsahu fotogalerie

3.3.8 Novinky

Modul novinky formou krátkých příspěvků informuje o aktuálním dění. Zpravidla bývají umístěny na titulní stránce webových prezentací (na vzorových webových stránkách VŠK Terč tuto funkci plní modul *Akce*).

3.3.9 Soutěže

Nad rámec zadání byl pro Vysokoškolský klub Terč naprogramován modul *Soutěže*. Přes uživatelské rozhraní systému iSys dovoluje zakládat nové soutěže, zapisovat do nich nové hráče a upravovat jejich výsledky v jednotlivých kolech. Běžný návštěvník stránek má pak výsledky soutěží přehledně zobrazeny v tabulce.

3.3.10 Anketa

Modul dovoluje návštěvníkům stránek hlasovat v anketách. Z každé IP adresy je povoleno jedno hlasování na každou anketu.

3.3.11 Vytvoření nového modulu

Na závěr této kapitoly uvedu návod k vytvoření nového modulu. Jak bylo už zmíněno, moduly jsou zapsány v tabulce *modules*. Je zde tedy nutné vložit nový řádek. Důležitými parametry jsou při tom ID modulu (sloupec *id*) a označení modulu ID2 (sloupec *module*). Podle označení ID2 vytvoříme nový adresář ve složce *system/modules*. Se stejným názvem vytvoříme i nový soubor *php*. Bude-li mít například náš nový modul označení *test*, soubor *test.php* bude umístěn v adresáři *system/modules/test*. V souboru vytvoříme třídu opět s označením ID2 (názvy tříd jsou vždy velkými písmeny, například *TEST*) jako rozšíření třídy *WINDOW*. Podle vzoru níže vložíme do třídy *constructor* a metodu *main*.

```
class TEST extends WINDOW {  
  
    function __construct($id) {  
        parent::__construct($id);  
    }  
}
```

```
function main() {  
    ...  
}  
}
```

Nyní už stačí přidat nový řádek do tabulky *windows*. Sloupce a vlastnosti systémového okna byly popsány v kapitole 3.2.2. Doplním pouze informace ke sloupcům *id_module* a *module*. Jak už název napovídá, sloupec *id_module* obsahuje ID modulu, ke kterému se váže. Sloupec *module* obsahuje příkaz určující obsah okna. Příkaz může obsahovat až 3 části oddělené dvojtečkou:

jmeno_modulu:metoda:a,b

První část je povinná, jde o označení modulu – ID2. Jestliže nejsou další části určeny, volá se metoda *main()* zvoleného modulu (třídy). Druhou částí je jméno metody a třetím jsou parametry volané metody (oddělují se tečkami).

Tímto je hotov základ nového modulu. Následně můžeme modul rozšířit o post skript **.post.php* a funkcí soubor **.fc.php*.

3.4 INSTALACE SYSTÉMU

Instalace systému je snadná a zkušený uživatel jej dokáže zprovoznit na novém serveru v několika minutách. Postup instalace je následující:

1. soubor *isys.zip* (součást přílohy) rozbalíme a obsah překopírujeme na server
2. do MySQL databáze prezentace importujeme soubor *isys.sql*, taktéž uložený v *isys.zip*.
3. Správně upravíme informace v konfiguračním souboru *system/settings.php*.

3.5 VZOROVÁ APLIKACE

Nasazení systému v praxi prezentuje internetová stránka vysokoškolského klubu Terč. Obsahově se jedná o standardní webovou prezentaci s dvouúrovňovým menu (obrázek 3.17). Využití zde nachází všechny moduly. Pro klub byly dále doprogramovány moduly *Soutěže* a *Terč-line*. Modul *Soutěže* umožňuje snadný zápis výsledků soutěží a jejich prezentaci, *Terč-line* slouží k vnitřním potřebám klubu.

Webová prezentace bude minimálně do konce státní závěrečné zkoušky dostupná na internetové adrese <http://new.vskterc.cz/> s vytvořeným administrátorským účtem:

přihlašovací jméno: test







heslo: test1



Vysokoškolský KLUB

Pondělí - Čtvrtek
15:00 - 01:00
Pátek - Neděle
19:00 - 01:00



-  [úvod](#)
-  [akce v terči](#)
-  [inormace pro kapely](#)
-  [činnost, historie](#)
-  [členové](#)
-  [pro zájemce o členství](#)



Vítejte mezi námi

Co jsme?

Studentský klub organizující zábavu (pro ostatní nejen studenty na kolejích) a provozující bar. (Více informací najdete na stránce činnost, historie)

Jak nás najít?

Adresa je uvedena v záhlaví stránky.

Možnost 1: šalinou(= tramvají) 12 nebo 13 na zastávku skácelova a pak autobusem č.53 na zastávku Kolejní, odkud se vydáte k bloku A06 a nebo se doptáte.

Možnost 2: šalinou 12/13 na konečnou (tech. park) a pak 3 minuty pěšky doleva nahoru na koleje. Najdete nás v budově, kde sídlí CESA (Centrum sportovních aktivit) naproti bloku A04(bývalý blok K3).

Kontakt?

Nejspolehlivější: je osobní kontakt na baru v ordinálních hodinách.

telefon: bar +420541149599, (kancelář +420541149598)

Na telefony není příliš spolehlutí, jelikož obvykle není na blízku ani operátor telefonu ani účetní email: vsk.terc@seznam.cz - mail se vybírá většinou jednou denně, přesto doporučuji důležitá oznámení poslat den předem.

pro kapely: podrobné informace a kontakty na klubového kulturistu jsou v infu pro kapely webmaster (Duncan): i.duncan.cz@gmail.com

Kdy máme otevřeno?

pondělí - čtvrtek







15:00 - 01:00

pátek - neděle

19:00 - 01:00

Případné změny jsou uvedeny na stránce aktualit .

Co u nás najdete?

-  příjemné prostředí, uvnitř klubu (zákaz kouření), nebo na zahrádce (kouření povoleno)
-  Zahrádka je otevřena během teplých letních měsíců a krásného počasí, zahrádka funguje do 22:00.
-  báječné lidi
-  skvělé pivo, zde bych chtěl pochválit vinikající péči o čepované pivo, jelikož jsme díky skvělému chlazení, údržbě v ýčepního zařízení a především lásce s jakou se o čepované pivo náš kolektiv stará získali certifikaci od Plzeňského prazdroje který zaručuje za pomoci stále kontroly vysokou kvalitu piva.
-  možnost zahrát si stolní fotbal (2 stoly), kulečník, billiard (na přelomu let 2008/2009 pořízen nový stůl), dále také společenské hry jako šachy, karty, člobrdo, pexeso atd.
-  turnaj v človeče nezlob se pořádaný většinou každý první čtvrtek v měsíci a nebo ligu ve stolním fotbálku který se koná každý první dvouciferný čtvrtek v měsíci. viz rozpis

A nakonec dobrá rada pro vás:

Choďte včas, hospoda není holubník.



přihlášení



Obrázek 3.17 – webová prezentace VŠK Terč

4. DALŠÍ MOŽNOSTI VÝVOJE SYSTÉMU A JEHO ZNÁMÉ NEDOSTATKY

Přestože systém v současném stavu obsahuje mnoho funkcí a doplňků, jedná se stále o první verzi a bylo by v budoucnu vhodné některé části kódu upravit a rozšířit systém o další funkce/moduly.

Několik návrhů na úpravu nebo rozšíření systému:

- Jádru systému nepůsobí celistvým dojmem. Přepsáním do vlastní třídy by se velká část kódu zpřehlednila.
- Správce souborů - modul pro vytváření, procházení a mazání souboru a složek.
- Logování činnosti přihlášených uživatelů.
- Instalace a deinstalace modulů – systém by si sám zkontroloval přítomnost všech modulů v adresáři *system/modules* a nabídl možnost jejich instalace a deinstalace. Třídy modulů by tak mohly obsahovat například metody `install()` a `uninstall()`.
- Vícejazyčnost internetové prezentace.

5. ZÁVĚR

Cílem diplomové práce bylo vypracovat systém pro správu webového obsahu. V první kapitole jsme se seznámili s technologiemi, s kterými systém pracuje. Jejich výhodou vidím především v široké podpoře na trhu (PHP, MySQL) – web hostingů. A jelikož se jedná o volně šířitelné technologie, jsou nízké i provozní náklady.

V druhé kapitole je popsán samotný systém. Při jeho vývoji byl kladen důraz na přehledné uživatelské rozhraní a zvyklosti uživatelů. Proto si systém iSys bere příklad z uživatelských rozhraní a funkcí dnes používaných operačních systémů PC. V tom vidím odlišnost a jedinečnost oproti ostatním systémům z oblasti správy obsahu. Odezvu systému na reakce uživatele a dynamiku systému zajišťují technologie AJAXu.

Systém iSys sestává z jádra (kapitola 3.2), podpůrných tříd a funkcí a modulů. Jádro zajišťuje inicializaci systému a správu prostředků. K potřebám jádra a uživatelského rozhraní bylo naprogramováno několik tříd. Důležitou roli při vykreslování systémových oken hrají třídy *TAB* (záložky), *TABLE* (tabulky), *FORM* a *CHECK* (formuláře), které významný dílem ulehčují tvorbu nových modulů. Moduly využívají systémové prostředky a svými funkcemi rozšiřují funkce celého systému. Nedělitelnou částí systému jsou moduly *Systém*, *Správa uživatelů* a *Správa účtu*. Dle zadání byly naprogramovány moduly *Novinky*, *Anketa*, *Fotogalerie* a *Knihá návštěv*. Všechny moduly spolu s návodem jak vytvořit modul nový jsou popsány v kapitole 3.3. Perspektiva systému je v jeho modularizaci.

Přestože bych z dnešního pohledu zapsal některé části kódu jiným způsobem, považuji systém iSys ve své první verzi za vydařený. Za dobu testování byly odstraněny všechny chyby a systém je spolehlivý. Kladnou vlastností je i plná podpora internetových prohlížečů Internet Explorer 7 a 8, Mozilla Firefox a Opera. Zápornou vlastností je jeho závislost na skriptovacím jazyku JavaScript.

Jestliže si uživatel JavaScript zakáže ve svém prohlížeči, systém nebude správně pracovat.

6. SEZNAM LITERATURY

- [1] Gilmore J.W.: Velká kniha PHP 5 a MySQL - kompendium znalostí pro začátečníky i profesionály. Brno, Zoner press, 2005, ISBN 80-86815-20-X.
- [2] Resig J.: JavaScript a Ajax - moderní programování webových aplikací. Brno, Computer Press, 2007, ISBN 978-80-251-1824-5.
- [3] Staniček P.: CSS Kaskádové styly - kompletní průvodce. Brno, Computer Press, 2003, ISBN 80-7226-872-4.
- [4] Cloud Computing, URL <http://cloud.abakowiki.cz>
- [5] EyeOS, URL <http://eyeos.org>
- [6] Bernard B.: Srovnání PHP a ASP.NET, Vysoká škola ekonomická v Praze, 2004, URL: <http://www.borber.com/files/Bernard-PHP-vs-ASP.NET.pdf>
- [7] Lacko L.: PHP 5 a MySQL 5 – Hotová řešení. Brno, Computer Press, 2007, ISBN 978-80-251-1695-1
- [8] MySQL, <http://www.mysql.com/>
- [9] Asleson A., Schutta N. T.,: AJAX – Vytváříme vysoce interaktivní webové aplikace. Brno, Computer Press, 2007, ISBN 80-251-1285-3
- [10] Lhoták J.: Pracujeme se session v PHP, URL:
<http://interval.cz/clanky/pracujeme-se-session-v-php/>

7. SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

AJAX (Asynchronous JavaScript and XML) – označuje soubor technologií k tvorbě interaktivních webových stránek

CGI – Common Gateway Interface – serverové rozhraní umožňující spouštět skripty

Cloud Computing – pojem označující distribuci aplikací a sdílení hardwarových prostředků pomocí sítě

CMS (content management system) – systém pro správu (nejčastěji webového) obsahu

CSS (Cascading Style Sheets) – kaskádové styly - jazyk je určený pro formátování obsahu internetových stránek

DOM (Document Object Model) – objektově orientovaná reprezentace internetové stránky

IP adresa – adresa síťového rozhraní v počítačové síti

JavaScript – skriptovací jazyk implementovaný do většiny dnešních internetových prohlížečů

MySQL – multiplatformní databázový server

PHP (Hypertext Preprocessor) – skriptovací jazyk určený pro programování dynamických internetových stránek/aplikací, PHP skripty se zpracovávají na straně serveru

px (pixel) – jeden zobrazovací bod (např. na monitoru nebo v rastrovém obrázku)

TinyMCE – webových wysiwyg editor textu

URI (Uniform Resource Identifier) – jednotný identifikátor zdroje (typicky na internetu)

wysiwyg (What you see is what you get) editor – vizuální editor

XHTML – jazyk definující vazby mezi webovými dokumenty a jejich strukturu

8. SEZNAM PŘÍLOH

- CD obsahující:
 - zdrojové kódy systému spolu s vytvořeným ukázkovým webem VŠK Terč a uloženou databází (*isys_terc.zip*)
 - zdrojové kódy samostatného systému a uloženou databází (*isys.zip*)
 - elektronickou podobu tohoto textu ve formátech PDF a DOC