

Modul systému Moodle pro snadnou archivaci protokolů a úkolů

Milan Medlík

Bakalářská práce
2008



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav aplikované informatiky

akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Milan MEDLÍK**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Téma práce: **Modul systému Moodle pro snadnou archivaci protokolů a úkolů**

Zásady pro vypracování:

1. Vytvořte rešerši na téma možností rozšíření e-learningového systému Moodle o další uživatelské moduly dostupné jak pro učitele, tak i studenty.
2. Vytvořte modul pro systém Moodle, který by umožňoval zálohování protokolů a dalších úkolů odevzdávaných studenty v jednotlivých kurzech.
3. Modul bude mít oddělené uživatelské rozhraní pro učitele i studenty.
4. Modul vygeneruje HTML soubor(y) uložený(é) v kořenovém adresáři archivu obsahující informace o názvech a náplni jednotlivých úkolů a umožňující procházení archivovaných dokumentů.

Výstupem bakalářské práce bude:

1. Textový dokument obsahující rešerši viz. bod 1 a programová dokumentace modulu
2. Zdrojové kódy modulu

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Welling, L., Thomson, L.: PHP a MySQL – rozvoj webových aplikací, SoftPress, 2002.
2. Kosek, J.: PHP tvorba interaktivních internetových aplikací Podrobný průvodce. 1. vyd. Praha: Grada, 1998 492 s. ISBN: 80-7169-373-1.
3. Schlossnagle, G.: Pokročilé programování v PHP 5. 1. vyd., Brno: Zoner Press, 2004. 640 s. ISBN: 80-86815-14-5.
4. Internetové zdroje:
 - a. Moodle [online]. Dostupná z URL: <http://moodle.org/>.
 - b. W3C – WORLD WIDE WEB Consortium [online]. Dostupná z URL: <http://www.w3.org/>.
 - c. php [online]. Dostupná z URL: <http://www.php.net/>.
 - d. The Apache Software Foundation [online]. Dostupná z URL: <http://www.apache.org/>.
 - e. MySQL [online]. Dostupná z URL: <http://mysql.com/>.
 - f. Softintegration [online]. Dostupná z URL: <http://www.softintegration.com/>.

Vedoucí bakalářské práce:

Ing. Michal Bližňák

Ústav aplikované informatiky

Datum zadání bakalářské práce:

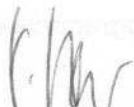
20. února 2008


Termín odevzdání bakalářské práce:

5. května 2008

Ve Zlíně dne 20. února 2008




prof. Ing. Vladimír Vašek, CSc.
děkan


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

ABSTRAKT

Cílem mé bakalářské práce je seznámit se s možnostmi rozšíření systému Moodle a vytvoření modulu do toho systému. Modul má umožňovat inteligentně archivovat protokoly odevzdávané v rámci laboratorních cvičení a seminářů různých kurzů. Je naprogramován ve skriptovacím jazyce PHP a napojen na databázi pomocí jazyku SQL. Pro jednodušší orientaci mezi archivovanými soubory je vygenerováno uživatelské rozhraní, které je realizováno propojením hypertextového jazyku HTML s jazykem JavaScript do jednoho dokumentu, který archivované soubory zobrazí v tabulce, jež se dá filtrovat podle názvu úkolu či podle uživatele, nebo jejich kombinací. Je také možné seřadit tabulku podle určitého sloupce.

Klíčová slova:

Moodle, modul, archivace, protokoly, PHP, HTML, JavaScript, SQL

ABSTRACT

The aim of my thesis is to introduce the possibilities of extending the system Moodle and creating the module for this system. The module is to provide intelligent archiving of protocols uploaded in laboratory practical trainings and seminars of different courses. It is programmed in the script language PHP and connected to the database via the language SQL. For a simpler orientation in the archive-files the user-interface is generated. This is realised by connecting the hypertext language HTML with the language JavaScript into one document, which shows the archived-files in a table that can be filtered by the name of the assignments or by the user, or their combination. It is also possible to sort the table by a specific column.

Keywords:

Moodle, module, archive, protocol, PHP, HTML, JavaScript, SQL

Tímto si dovoluji poděkovat vedoucímu bakalářské práce ing. Michalu Bližňákovi za odborné vedení během této bakalářské práce a především za jeho cenný čas při poskytování rad a informací, bez kterých by tato práce nemohla na odborné půdě obstát.

OBSAH

| | |
|---|-----------|
| ÚVOD | 8 |
| I TEORETICKÁ ČÁST | 9 |
| 1 MOODLE OBECNĚ | 10 |
| 2 ROZŠÍŘENÍ MOODLE | 11 |
| 2.1 PLUGIN BLOCK | 11 |
| 2.1.1 Tvorba jednoduchého bloku | 11 |
| 2.1.2 Funkce init | 11 |
| 2.1.3 Funkce get_content..... | 12 |
| 2.1.4 Funkce instance_allow_multiple | 12 |
| 2.1.5 Celý příklad | 13 |
| 3 HTML | 14 |
| 3.1 KONCEPCE | 14 |
| 3.2 DRUHY ZNAČEK..... | 14 |
| 3.2.1 Strukturální značky..... | 14 |
| 3.2.2 Popisné (sémantické) značky..... | 14 |
| 3.2.3 Stylistické značky | 15 |
| 3.3 PŘEDEPSANÁ STRUKTURA:..... | 15 |
| 3.4 BUDOUCNOST HTML | 15 |
| 4 PHP | 16 |
| 4.1 KONCEPCE | 16 |
| 4.2 SKRIPT ARCHIVE.PHP | 16 |
| 5 JAVACRIPT | 17 |
| II PRAKTICKÁ ČÁST | 18 |
| 6 POPIS | 19 |
| 6.1 APLIKAČNÍ ČÁST | 19 |
| 6.2 ŠABLONY | 19 |
| 6.3 LOKALIZAČNÍ ČÁST..... | 20 |
| 7 POPIS ČÁSTI KÓDU A NÁVAZNOST NA MOODLE | 21 |
| 7.1 ASSIGNMENTS_LIB.PHP | 21 |
| 7.1.1 Parametry funkce getSqlQuery | 21 |
| 7.1.2 Kód funkce getSqlQuery | 22 |
| 7.1.3 Funkce createIndexHtml..... | 23 |
| 7.1.4 Funkce createZIP | 23 |
| 7.2 BLOCK_ASSIGNMENTS_DOWNLOAD.PHP | 24 |
| 7.2.1 Funkce get_content..... | 24 |
| 7.3 PUSH_ZIP.PHP..... | 25 |
| 8 POPIS VÝSTUPNÍHO SOUBORU INDEX.HTML | 26 |

| | | |
|-----------|---|-----------|
| 8.1 | ULOŽENÍ DAT..... | 26 |
| 8.2 | ZOBRAZENÍ DAT..... | 26 |
| 8.3 | FILTROVÁNÍ DAT..... | 27 |
| 8.4 | ŘAZENÍ DAT..... | 28 |
| 9 | UŽIVATELSKÁ PŘÍRUČKA..... | 29 |
| 9.1 | UŽIVATELSKÁ PŘÍRUČKA K BLOKU | 29 |
| 9.1.1 | Rozvržení komponent..... | 30 |
| 9.1.2 | 1 - výběr úkolu..... | 30 |
| 9.1.3 | 2 – výběr uživatele..... | 31 |
| 9.1.4 | 3 - Výběr časového rozmezí | 32 |
| 9.1.5 | 4 – Stažení souboru .zip..... | 33 |
| 9.2 | UŽIVATELSKÁ PŘÍRUČKA V INDEX.HTML | 33 |
| 9.2.1 | Možnosti filtrování | 34 |
| | ZÁVĚR..... | 35 |
| | SEZNAM POUŽITÉ LITERATURY | 36 |
| 10 | INTERNETOVÉ ZDROJE..... | 37 |
| | SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK | 38 |
| | SEZNAM OBRÁZKŮ..... | 39 |
| | SEZNAM PŘÍLOH..... | 41 |

ÚVOD

System Moodle je používán na spoustě vzdělávacích zařízení od vysokých škol přes školy vyšší až po školy střední. Osvědčil se i na UTB Zlín, Fakultě Aplikované Informatiky. Studenti zde odevzdávají své protokoly v rámci laboratorních cvičení a seminářů různých kurzů. Jelikož Fakulta Aplikované Informatiky pod sebou sdružuje poměrně velké množství studentů, je obsah databáze odevzdaných prací poměrně velký a neustále se zvětšuje. Je nutné, aby i tyto položky byly čas od času zálohovány. System Moodle jako takový nepodporuje přímou archivaci protokolů a úkolů, ale poskytuje prostředky k jeho rozšíření. Tato bakalářská práce má za úkol usnadnění celého procesu archivace, jak na straně učitelů tak na straně studentů prostřednictvím rozšiřovacího modulu pro tento systém.

I. TEORETICKÁ ČÁST

1 MOODLE OBECNĚ

Moodle [čti můdl] je softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu. Jedná se o neustále se vyvíjející projekt, navržený na základě sociálně konstruktivistického přístupu k vzdělávání.

Moodle je poskytován zdarma jako Open Source software spadající pod obecnou veřejnou licenci GNU. To v zásadě znamená, že je chráněn autorskými právy, ale poskytuje přitom uživatelům značnou svobodu. Moodle můžete kopírovat, používat i upravovat, pokud souhlasíte s tím, že: budete tento zdroj poskytovat ostatním; nebudete měnit ani odstraňovat původní údaje o licencích a autorských právech, a uplatníte stejné licenční podmínky i u jakýchkoliv odvozených produktů. Pro podrobnější informace nahlédněte do plného znění licence. S případnými dotazy se laskavě obraťte přímo na držitele autorských práv.

Moodle lze použít na jakémkoliv počítači s fungujícím PHP. Podporuje řadu typů databází, především PostgreSQL a MySQL.

Slovo Moodle bylo původně akronymem pro Modular Object-Oriented Dynamic Learning Environment (Modulární objektově orientované dynamické prostředí pro výuku); tato informace může být zajímavá především pro programátory a teoretické pedagogy. Lze ho také považovat za sloveso, které popisuje proces líného bloumání od jednoho k druhému, dělání věcí podle svého, hravost, která často vede k pochopení problému a podporuje tvořivost. V tomto smyslu se vztahuje jak k samotnému zrodu Moodlu, tak k přístupu studenta či učitele k výuce v on-line kurzech. [4]

Od roku 2000 je systém Moodle používán také na UTB ve Zlíně na serveru www.vyuka.fai.utb.cz, kde slouží k podpoře výuky na Fakultě Aplikované Informatiky - učitelé mají možnost si zde zakládat kurzy a plnit je elektronickými materiály, ke kterým pak studenti mají v rámci výuky přístup. Systém Moodle, instalovaný na tomto serveru, studentům i učitelům také pomáhá s organizací výuky - zjednodušuje odevzdávání úkolů, umožňuje automatické rozesílání emailů s Novinkami nebo zprávami z diskusních fór, poskytuje funkce pro hodnocení (bodování) úkolů v předmětech, psaní a automatické ohodnocení testů, ...

2 ROZŠÍŘENÍ MOODLE

Jak bylo avizováno výše, písmeno M v názvu modle znamená Modulární. Z toho plyne, že funkcionality systému Moodle se dá dále rozšiřovat. Hlavní cesta, jak se dá Moodle rozšířit, je užití API pro mnoho typů pluginů. Např.: Activity modules, Admin reports, Assignment types, Authentication plugins, Blocks, Course formats, ... [5]

2.1 Plugin Block

Při tvorbě bakalářské práce jsem využil jen jednoho API, a to pro tvorbu bloků. Jedná se o API, které vytvoří malé rozhraní po stranách systému Moodle; pro příklad v něm můžeme vidět kalendář nadcházejících aktivit, nebo poslední provedené akce v kurzu, atd.

2.1.1 Tvorba jednoduchého bloku

Vytvoření vlastního jednoduchého bloku není těžké. Začneme vytvořením adresáře obsahujícího název bloku v Moodle adresáři „Blocks“ (např. Blocks/simplehtml), v něm vytvoříme soubor skládající se ze slova „block“ a názvu bloku odděleného podtržítkem (např. block_simplehtml.php) . Ve skriptorech souboru vytvoříme třídu stejného jména jako soubor, která je potomkem třídy block_base.

```
<?php
class block_simplehtml extends block_base
{
}
?>
```

2.1.2 Funkce init

Funkcí *init* řekneme systému moodle nadpisek celého bloku a jeho verzi. Nadpisek jsme získali funkcí *get_string*, která spadá do lokalizační části systému moodle a doplní text podle nastaveného jazyka. Verze modulu je realizována ve formátu RRRRMMDD00. Verze modulu je ukládána do databáze a v případě, že systém moodle detekuje novější verzi, je o tom administrátor systému informován.

```
function init()
{
    $this->title = get_string('simplehtml', 'block_simplehtml');
    $this->version = 2004111200;
}
```

2.1.3 Funkce `get_content`

Obsah (content) bloku vrací funkce `get_content`. Na prvních řádcích je testováno, jestli je kontent již vytvořen, tedy není `NULL` vrať již vytvořený kontent. V případě, že kontent není vytvořen, vytvoří jej jako standardní třídu (`stdClass`). Nejprve je nastaven `text`, který reprezentuje tělo bloku. Nastavením proměnné `footer` nastavíme patičku bloku. Tělo i patička bloku mohou být plnohodnotný HTML text včetně jeho tagů, javascriptu a v neposlední řadě kaskádových stylů (css).

```
function get_content()
{
    if ($this->content !== NULL)
    {
        return $this->content;
    }

    $this->content = new stdClass;
    $this->content->text=
        '< h1>Tělo SimpleHTML bloku</h1>'
        . '<p>Zde může být text, odkaz nebo formulář, ...</p>';
    $this->content->footer = 'Patička tady...';

    return $this->content;
}
```

2.1.4 Funkce `instance_allow_multiple`

Návratová hodnota funkce `instance_allow_multiple` říká systému moodle, jestli tento blok může být zobrazen vícekrát. V tomto příkladě je nastavena návratová hodnota na `true`, takže tento modul může mít více instancí.

```
function instance_allow_multiple()
{
    return true;
}
```

2.1.5 Celý příklad

Takto by mohl vypadat jednoduchý blok do systému moodle.

```
<?php
class block_simplehtml extends block_base
{
    function init()
    {
        $this->title = get_string('simplehtml', 'block_simplehtml');
        $this->version = 2004111200;
    }

    function get_content()
    {
        if ($this->content != NULL)
        {
            return $this->content;
        }

        $this->content = new stdClass;
        $this->content->text=
            '< h1>Tělo SimpleHTML bloku</h1>'
            . '<p>Zde může být text, odkaz nebo formulář, ...</p>';
        $this->content->footer = 'Patička tady...';

        return $this->content;
    }

    function instance_allow_multiple()
    {
        return true;
    }
}
?>
```

3 HTML

3.1 Koncepce

Přestože systém moodle je postaven na skriptovém jazyku PHP, je nutné se alespoň okrajově zmínit o jazyku HTML, který je v tomto případě výstupem a je zobrazen pomocí internetových prohlížečů (např. Opera, Mozilla, IE, ...). Tento jazyk je také použit při tvorbě šablon, jednak pro samotný blok a jednak pro výstupní index.html. Jazyk HTML je charakterizován množinou značek a jejich atributů definovaných pro danou verzi. Mezi značky se uzavírají části textu dokumentu, a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky (< a >). Část dokumentu tvořená otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. element (prvek) dokumentu. Například je otevírací značka pro zvýraznění textu a Červená Karkulka je element obsahující zvýrazněný text. Součástí obsahu elementu mohou být další vnořené elementy. Tagy mohou obsahovat atributy, které popisují jejich vlastnosti nebo nesou jinou informaci. Příkladem může být odkaz (tag a), jehož atribut href říká, kam se uživatel po kliknutí na něj dostane (v tomto příkladu na stránku <http://example.com>):

```
<a href="http://example.com">text odkazu</a>
```

3.2 Druhy značek

Značky lze z hlediska významu rozdělit na tři základní skupiny:

3.2.1 Strukturální značky

rozvrhují strukturu dokumentu, příkladem jsou odstavce (<p>), nadpisy (<h1>, <h2>). Dodávají dokumentu formu.

3.2.2 Popisné (sémantické) značky

popisují povahu obsahu elementu, příkladem je nadpis (<title>) nebo adresa (<address>). Současný trend je orientován právě na sémantické značky, které usnadňují automatizované zpracovávání dokumentů a vyhledávání informací v záplavě dokumentů na webu. Vyvrcholením této snahy je v současné době jazyk XML.

3.2.3 Stylistické značky

určují vzhled elementu při zobrazení, typickým příkladem je značka pro tučné písmo (). Od tohoto druhu značek se postupně upouští, trendem je používání kaskádových stylů, které vzhled popisují odděleně od obsahu dokumentu. Mezi důvody pro neužívání těchto značek patří především to, že tyto značky jsou orientovány na prohlížení na obrazovce počítače, příliš se však nepočítá s používáním dokumentu jiným způsobem – alternativní prohlížeče pro postižené (čtečky pro slepce), v mobilních zařízeních a podobně. Kaskádové styly umožňují definovat rozdílné zobrazení pro různá zařízení.

3.3 Předepsaná struktura:

Deklarace DTD - Je povinná až ve verzi 4.01, je uvedena direktivou <!DOCTYPE.

Kořenový element - Element html (značky <html> a </html>) reprezentuje celý dokument. Kořenový element je povinný, ale otevírací a ukončovací značka samotná povinná není (pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle kontextu).

Hlavička elementu - Obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení. Hlavička je uzavřena mezi značky <head> a </head>. Element head je opět povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

Tělo dokumentu - Obsahuje vlastní text dokumentu. Vymezuje se značkami <body> a </body>. Element body je povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

3.4 Budoucnost HTML

Vývoj jazyka HTML byl původně ukončen verzí 4.01. Dle W3C měl další vývoj psaní dokumentů na web patřit jazyku XHTML – následníkovi HTML, využívajícímu univerzální jazyk XML. Některým lidem se však vývoj okolo XHTML nezamlouval. Část z nich včetně některých tvůrců webových prohlížečů, jako například Mozilla Foundation, Opera Software či Apple, založila iniciativu WHATWG, jejímž cílem bylo vytvořit novou verzi HTML, která se posléze začala označovat jako „HTML 5“. [10]

4 PHP

4.1 Koncepce

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocesor“, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML či WML, což je velmi výhodné pro tvorbu webových aplikací. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti. Syntaxe jazyka kombinuje hned několik programovacích jazyků (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez úprav na mnoha různých operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP, ...)

PHP se stalo velmi oblíbeným především díky jednoduchosti použití a tomu, že kombinuje vlastnosti více programovacích jazyků a nechává tak vývojáři částečnou svobodu v syntaxi. V kombinaci s databázovým serverem (především s MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Díky velmi častému nasazení na serverech se vžila zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP nebo Perl. S verzí PHP 5 se výrazně zlepšil přístup k objektově orientovanému programování podobný Javě. [11]

4.2 Skript `archive.php`

Jde o malou knihovnu, kterou napsal Devin Doucette pod licencí GNU GPL pro tvorbu a extrakci archivu ve formátech `.tar` `.gzip` `.bzip2` `.zip`. Podporuje relativní cesty, komentáře a rekurzi skrz adresář. Ukládání archivu přímo na disk dovoluje uživateli přímé stahování, nebo umožňuje vrátit obsah souboru v řetězci či jako pole. Nepotřebuje žádný externí program pro spuštění, pouze PHP musí být zkompileováno s parametrem `-with-zlib` pro `gzip` a `-with-bzip2` pro `bzip2`. Podporuje také vytváření samorozbalovacích archivů. Získat ho lze na adrese <http://phpclasses.web4u.cz/browse/package/945.html>

5 JAVACRIPT

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

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

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl v červenci 1997 standardizován asociací ECMA (European Computer Manufacturers Association) a v srpnu 1998 ISO (International Organization for Standardisation). Standardizovaná verze JavaScriptu je pojmenována jako ECMAScript a z ní byly odvozeny i další implementace, jako je například ActionScript.

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

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

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

II. PRAKTICKÁ ČÁST

6 POPIS

Vytvořený modul pro e-Learningový systém Moodle je postaven na API plugin blok. Jeho aplikační část je zpracována ve skriptovacím jazyce PHP. Jeho uživatelská část je zpracována v jazyce HTML a JavaScript. Pro nový modul do systému Moodle bylo potřeba zvolit vhodné jméno. Nazval jsem jej tedy „assignments downloader“. Z americké angličtiny to můžeme přeložit jako stahovač úkolů, protože v systému Moodle se protokolům, které studenti odevzdají v rámci laboratorních cvičení a seminářů kurzů říká jednotně „assignments“. Modul jsem rozdělil do několika zdrojových souborů, které by se daly dále rozčlenit do tří základních skupin:

6.1 Aplikační část

Aplikační část se stará o obsluhu systému Moodle. Jedná se o tři skriptové soubory v jazyce PHP *assignments_lib.php*, *block_assignment_download.php* a *pushzip.php*. Soubor *assignments_lib.php* je mnou napsaná knihovna pro získání informací o úkolech, jako jsou jejich názvy nebo aktuálně zapsaní členové v kurzu, dostupné soubory atd. Je to jádro celého modulu, více bude rozepsáno v Popisu části kódu. Jeho funkce jsou volány uvnitř skriptu *block_assignment_download.php*, jehož úkolem je zobrazit blok a také uvnitř souboru *pushzip.php* - tento PHP skript je volán hypertextovým odkazem a na základě vstupních parametrů, mezi které může patřit id kurzu, id úkolů, id uživatele, atd. vybere ze systému Moodle uživatelem vložené soubory, jako jsou protokoly ze seminářů a laboratorních cvičení, vloží je do zip archívu a nabídne jej ke stažení.

6.2 Šablony

Tato část se skládá ze dvou souborů se šablonami v jazyce HTML. Soubor *temlate_block.html* určuje vzhled a rozmístění komponent ve formuláři, jako jsou popisky, komboboxy, a checkboxy v bloku, který se bude zobrazovat v systému Moodle jako blok. Soubor *temlate_index.html* určuje vzhled a rozmístění prvků ve výstupním indexu. Obsahuje také funkce v jazyce javascript pro filtrování jednotlivých úkolů a uživatelů. Více v části Generování výstupu.

6.3 Lokalizační část

Jedná se o soubory, které jsou načítány automaticky Moodle podle zvoleného jazyka v Moodle. Jsou umístěny v adresáři *lang/locale/*, kde „locale“ představuje jazyk (např. *en_utf8* nebo *cz_utf8*). V jednotlivých adresářích se nachází soubor *block_assignments_download.php*, jehož obsahem je rozšíření asociovatelného pole s názvem *string*, tedy pole, jehož položky mají klíč a hodnotu. Každá hodnota se dá vyvolat přes svůj klíč. Tyto hodnoty jsou získávány pomocí funkce *get_string()*, která je zmiňována v Teoretické části – tvorba jednoduchého bloku. Zde je část souboru pro lokalizaci do anglického jazyka, který je jako výchozí v případě, že pro zvolený jazyk lokalizace chybí.

```
<?php
$string['downloader'] = 'Assignment Download';
$string['assignments_text_desc'] = 'Select assignment: ';
$string['assignments_text_all'] = 'All assignments in this course';
// ...
```

7 POPIS ČÁSTI KÓDU A NÁVAZNOST NA MOODLE

V této části jsou vysvětleny důležité části kódu a jeho návaznost na systém Moodle, která je z části realizována přes funkce, a z části prací s databází a v poslední řadě procházením souborového systému pro získávání jednotlivých souborů, které jsou odevzdávány jako protokoly v rámci laboratorních cvičení a seminářů různých kurzů.

7.1 assignments_lib.php

Jak bylo zmíněno výše, tento PHP skript je stěžejním skriptem celého Moodle modulu. Nachází se v adresáři lib/ . Jeho hlavní funkcí je dohledávání úkolů v databázi, jež obstarává funkce *getSqlQuery* pomocí propojení tabulek, které je znázorněno v příloze [P1]. Tato funkce je volána dalšími třemi funkcemi *getAssignmentRecords* – získá všechny údaje o úkolech. Tato data jsou použita při generování výstupního indexu, *getAssignmentNameRecords* – získá všechny názvy úkolu v daném kurzu, *getUsersRecords* – získá všechny uživatele, kteří odevzdali nějaký úkol v daném kurzu.

7.1.1 Parametry funkce getSqlQuery

sql_select – které sloupce z sql tabulky požadujeme, na příklad jméno úkolu

courseid – id požadovaného kurzu

assignmentid – id požadovaného úkolu, výchozí hodnota je -1, která znamená, že budou načteny všechny úkoly v aktuálním kurzu

userid – id požadovaného uživatele, výchozí hodnota je -1 , která znamená, že budou načtení všichni uživatelé, kteří nějaký úkol nahrávali.

date_start – určuje nejstarší datum, které mají mít nahrávané úkoly

date_end – určuje nejmladší datum, které mají mít nahrávané úkoly

7.1.2 Kód funkce getSqlQuery

```
function getSqlQuery( $sql_select,
                    $courseid,
                    $assignmentid = '-1',
                    $userid = '-1',
                    $date_start = '',
                    $date_end = '')
{
    global $CFG;

    $sql_from = " FROM {$CFG->prefix}assignment_submissions asub"
                . " INNER JOIN {$CFG->prefix}assignment a"
                . " ON a.id = asub.assignment"
                . " INNER JOIN {$CFG->prefix}user u"
                . " ON asub.userid = u.id";
    $sql_where = " WHERE a.course = $courseid AND asub.numfiles > 0";

    // set assignment id
    if( $assignmentid != '-1')
        $sql_where .= " AND a.id = $assignmentid";

    // set user id
    if( $userid != '-1')
        $sql_where .= " AND asub.userid = $userid";

    // date_start or date_end hadn't set
    if( $date_start == '' && $date_end == '' )
    {
        $context = get_context_instance(CONTEXT_COURSE, $courseid);

        if( $context == false)
            return $context;

        //var_dump($context);
        $sql_from .= " INNER JOIN {$CFG->prefix}role_assignments ar"
                    . " ON asub.userid = ar.userid";
        $sql_where .= " AND ar.contextid = $context->id";
    }
    // date selected
    else
    {
        if( $date_start != '')
            $sql_where .= " AND asub.timemodified > $date_start";
        if( $date_end != '')
            $sql_where .= " AND asub.timemodified < $date_end";
    }

    $sql_sortby = "";

    // debug...
    // print_r($sql_select . $sql_from . $sql_where . $sql_sortby);

    return "SELECT " .
           $sql_select .
           $sql_from .
           $sql_where .
           $sql_sortby;
}
```

7.1.3 Funkce createIndexHtml

Tato funkce, jak plyne z anglického názvu, generuje `index.html`, který se přiřazuje k archivu. Její hlavní část spočívá v načtení šablony `template_index.html` a přidání úkolů a uživatelů z databáze. Tato data dostává tato funkce jako její parametry. Načtení šablony:

```
// open and read teplate
$fd = fopen ($file_name, "r");
$template = fread($fd, filesize($file_name));
fclose($fd);
```

Vložení dat do souboru `index.html` je realizováno přes PHP funkci `str_replace`, která nahradí specifická slova uvnitř šablony za požadovaná data. Tato specifická (klíčová) slova jsou realizována jako HTML poznámka, takže v případě jejich nenahrazení se `index.html` v internetovém prohlížeči zobrazí zcela korektně.

```
// set arrays to template
$template = str_replace("/*HERE-DATA*/", "$html_text", $template);
$template = str_replace("<!--HERE-NAME-->",
                        $COURSE->fullname,
                        $template);
$template = str_replace("<!--HERE-ASSIGNMENT_FILTER-->",
                        $html_assignment_names,
                        $template);
$template = str_replace("<!--HERE-USER_FILTER-->",
                        $html_user_names,
                        $template);
```

7.1.4 Funkce createZIP

Funkce `createZip` volá všechny předešlé jmenované funkce. Její vstupní parametry jsou stejné jako u funkce `getSqlQuery` a jejím výstupem je struktura s archivem typu ZIP. Vytvoření samotného archivu vypadá následovně:

```
// new zip file
$zipfile = new zip_file($filename);
$compression = isset ($configdata->compression) ?
    $configdata->compression : block_assignments_default_compression();

$maxsize = isset ($configdata->maxsize) ?
    $configdata->maxsize: block_assignments_default_maxsize();

// zip file option
$zipfile->set_options(array (
    'basedir' => "${CFG->dataroot}${courseid}/${CFG->moddata}/",
    'followlinks' => 1,
    'inmemory' => 1,
    'level' => $compression,
    'recurse' => 1,
    'maxsize' => $maxsize*1024*1024
));
```

Do tohoto archívu jsou postupně přidávány soubory, které byly do modulu nahrány studenty jako odevzdané protokoly v rámci laboratorních cvičení a seminářů různých kurzů a samotný *index.html*, který byl vytvořen funkcí *createIndexHTML*.

```
foreach($assignments as $assignment)
{
    // al_debug($assignment, "record");
    // Add files to Zip
    if (isset($assignment["filedesc"]))
    {
        foreach($assignment["filedesc"] as $filedesc)
            $zipfile->add_files($filedesc["path"]);
    }
}
// add index
$zipfile->add_files("index.html");
```

7.2 block_assignments_download.php

Tento skriptový soubor je volán systémem Moodle a zobrazuje blok, pokud je administrátorem nebo učitelem s rozšířenou funkcí v kurzu aktivován. Obsahuje funkci *init*, která dává systému Moodle základní informace o modulu, jako je jeho verze a název titulku. Ale hlavně funkci *get_content*.

7.2.1 Funkce get_content

Ve funkci *get_content* je načtena šablona ze souboru *template_blog.html* lokalizuje ji. Dále doplní informace o kurzu, jako názvy úkolů. Pokud je přihlášen správce nebo učitel působící v kurzu, jsou doplněni všichni uživatelé, kteří odevzdali svůj úkol. Pokud je přihlášen student, je možné vybrat jen sám sebe. Lokalizace a doplnění informací o kurzu je opět uskutečněno nahrazováním řetězců v šabloně pomocí funkce *str_replace*.

```
// set localize texts
$template = str_replace($this->tag_assignments_desc,
                        $this->text_assignments_desc, $template);
$template = str_replace($this->tag_assignments_all,
                        $this->text_assignments_all, $template);
$template = str_replace($this->tag_users_desc,
                        $this->text_users_desc, $template);
$template = str_replace($this->tag_select_interval,
                        $this->text_select_interval, $template);
// show user in the course
// see to setUserToBlock
$template = str_replace($this->tag_users_here,
                        $this->setUserToBlock(), $template);
$template = str_replace($this->tag_assignments_here,
                        $this->getAssignmentsToBlock(), $template);
$template = str_replace($this->tag_action, $CFG->wwwroot .
                        $assignments_block . "push_zip.php", $template);
$template = str_replace($this->tag_courseid, $COURSE->id, $template);
```


Funkce vrací *content*, který si systém Moodle zpracuje, a zobrazí ho v uživatelském rozhraní. Do patičky je přiřazen hypertextový odkaz na stáhnutí ZIP souboru, který obstarává skript *pushzip.php*

```
// show content
$this->content = new stdClass;
$this->content->text = $template;
$this->content->footer =
    "<a href=\"javascript:document.assignments_form.submit();\">"
    . " language=\"JavaScript\">"
    . "$this->text_download_file" . '</a>';
```

7.3 push_zip.php

Tomuto velice malému skriptu jsou předána data z formuláře zobrazeného v uživatelském rozhraní systému Moodle. Nejprve se zkontroluje, zda je možné pokračovat:

```
// Securitycheck
if (!$courseid)
{
    error("no courseid!");
    exit ();
}
```

Na základě získaných dat z formuláře je vygenerován ZIP soubor popsanou metodou *createZip* (ze souboru *assignments_lib.php* popsaného výše), a takto vytvořený archiv je nabídnut ke stažení.

```
$zipfile = createZip($courseid, $assignments, $userid, $date_start, $date_end);
$zipfile->download_file();
```

8 POPIS VÝSTUPNÍHO SOUBORU INDEX.HTML

Soubor *index.html* je součástí ZIP archivu se zálohou úkolů, vygenerovaný modulem *assignments_download*. Tento soubor slouží k lepší orientaci v archivu. Mezi jeho přednostní funkce patří to, že slouží k filtrování dat podle názvu úkolu, podle jména uživatele, nebo jejich libovolnou kombinací. Filtrování je realizováno jazykem JavaScript integrovaným do HTML souboru. Jeho vzhled je ovlivněn kaskádovými styly CSS taktéž integrovanými do HTML souboru. Nejsou tudíž přímými potomky vzhledu systému Moodle.

8.1 Uložení dat

Daty se myslí informace o úkolech, obsahují id úkolu a id uživatele, tyto však jsou skryty. Sklouží pouze k filtrování, dále obsahují název úkolu, jméno a příjmení uživatele, jméno souboru s odkazem na něj a datum vložení souboru. Tato data jsou uložena jako pole v JavaScriptu :

```
// JavaScript Document
var afile1 = new Array('1', '2', 'ukol 2', 'admin', 'Správa', 'Uživatel', '<a href="assignment/1/2/DSC00281.JPG">DSC00281.JPG</a><br />', '26.03.2008');
var afile2 = new Array('1', '3', 'ukol 2', 'miuan', 'milan', 'medlik', '<a href="assignment/1/3/DSC00279.JPG">DSC00279.JPG</a><br />', '31.03.2008');
var afile3 = new Array('1', '6', 'ukol 2', 'student2', 'Jarek', 'Novak', '<a href="assignment/1/6/eclipse.ini">eclipse.ini</a><br />', '14.04.2005');
var afile4 = new Array('1', '5', 'ukol 2', 'student1', 'mirek', 'navratil', '<a href="assignment/1/5/DSC00284.JPG">DSC00284.JPG</a><br />', '15.04.2008');
var data = [afile1 ,afile2 ,afile3 ,afile4];
```

8.2 Zobrazení dat

Zobrazení dat je realizováno přes dvě JavaScriptová pole, v jednom jsou obsažena všechna data a je pojmenováno *data* (je zmíněno v předchozí části) a pole jménem *attachments*, ve kterém jsou jen vyfiltrovaná data z pole *data* a tato jsou zobrazena pomocí funkce *show_attachments*, která projde celé pole *attachments* a vytvoří z něj text obsahující HTML tabulku.

```
for( i = 0; i != attachments.length; i++)
{
    if( i%2 != 0)
        html_text += '<tr>';
    else
        html_text += '<tr style="background-color:PaleGreen;">';
        // put cells
        for( n = 2; n < attachments[i].length; n++)
            html_text += cell(attachments[i][n]);

        html_text += '</tr>';
}
```

Tento text s tabulkou je pak vložen pomocí JavaScriptové funkce *innerHTML* do dokumentu do div-bloku, který má id *attachment_div*

```
html_text = show_attachments();  
  
var element = document.getElementById('attachment_div');  
  
if(element != null)  
    element.innerHTML = html_text;
```

8.3 Filtrování dat

Jak bylo výše zmíněno, ve skriptu jsou dvě pole – jedno obsahující všechna data (*data*) a druhé obsahující pouze data, která se mají zobrazit (*attachments*), tedy data již vyfiltrovaná. Filtrování dat provádí funkce *set_filter*, jejíž vstupní parametr je sloupec, podle kterého se má filtrovat (používán je první a druhý sloupec, který obsahuje id úkolu, respektive id uživatele). Další parametr je hodnota, kterou musí řádek na požadovaném sloupci obsahovat, aby byl zařazen do pole zobrazení. Poslední parametr *element_id* slouží pouze k informační hodnotě, který filtr je aktuálně zvolen. Jelikož jsou v dokumentu obsaženy dva filtry, je nutno porovnávat, zda řádek splňuje podmínky obou filtrů.

```
// for all data  
for( n = 0; n != data.length ; n++)  
{  
    // set add  
    var add = true;  
  
    // for all filter  
    for(var f = 0; f != ga_show_filter.length; f++)  
        // filter on index f is set to false  
        // or filter is set the value [1]  
        // in filter column [0]  
        if( ga_show_filter[f][1] != false  
            && data[n][ga_show_filter[f][0]] != ga_show_filter[f][1])  
            // filter disable add  
            add = false;  
  
    if(add)  
        // attachments  
        attachments.push(data[n].slice());  
}
```

8.4 Řazení dat

Vyfiltrovaná data se dají dále řadit podle uživatelem zadaných sloupců, jako je jméno, příjmení, soubor, datum odevzdání, či jméno úkolu. Řazení dat je realizováno funkcí *sort_attachments*. Její vstupní parametry jsou číslo sloupce, podle kterého se má řadit, a typ – jestli sestupně, nebo vzestupně. Tato funkce provádí řazení metodou bubble-sort.

```
// loop if changed row
while(changed)
{
    // set default to row wasn't changed
    changed = 0;
    for(n = 0; n < attachments.length -1; n++)
    {
        // compare string in column
        if((sort_type
            && attachments[n][column+2] > attachments[n+1][column+2]) ||
            (!sort_type
            && attachments[n][column+2] < attachments[n+1][column+2]))
        {
            // save row to temp
            var row = attachments[n];
            // n+1 row insert to upper
            attachments[n] = attachments[n+1];
            // n row set to down
            attachments[n+1] = row;
            // set changed row
            changed = 1;
        }
    }
}
```

9 UŽIVATELSKÁ PŘÍRUČKA

9.1 Uživatelská příručka k bloku

Modul pro zálohování odevzdaných úkolů se nachází na jedné ze stran klientské oblasti systému Moodle mezi ostatními bloky.

The screenshot shows the Moodle course interface for 'Ukazkovy kurz'. The 'Assignment Download' block is highlighted with a red circle. It contains a 'Select assignment:' dropdown menu set to 'All assignments in this course', a 'Select user:' dropdown set to 'All users in this course', and two 'Select year interval' sections, each with a year dropdown (2008) and two numeric dropdowns (1). A 'download zip file' button is at the bottom. The 'Topic outline' shows a list of tasks: 'ukol 1' through 'ukol 5'. The right sidebar contains 'Latest News', 'Upcoming Events', and 'Recent Activity' sections.

1. Ukázka umístění bloku

V tomto případě jej můžeme najít vlevo nahoře, ale jeho pozice může být libovolná, a záleží na správci kurzu, kam jej nechá umístit. Pomocí pozičních tlačítek.

This close-up shows the 'Assignment Download' block in edit mode. A red circle highlights the top toolbar, which includes icons for moving the block (four arrows), deleting it (an 'X'), and a refresh icon. Below the toolbar are the same selection options as in the previous screenshot: 'Select assignment:', 'Select user:', 'Select year interval', and the 'download zip file' button.

2. Poziční tlačítka v editačním režimu

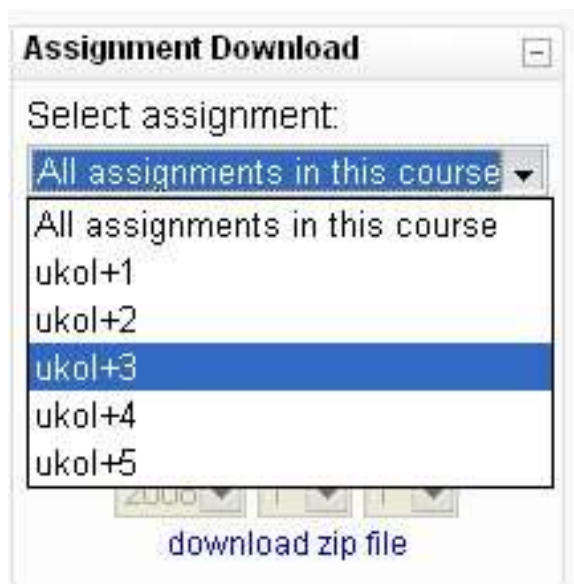
9.1.1 Rozvržení komponent



3. Popis komponent v bloku

9.1.2 1 - výběr úkolu

V tomto comboboxu si může uživatel či učitel zvolit všechny odevzdané protokoly, nebo jen protokoly z konkrétního úkolu.



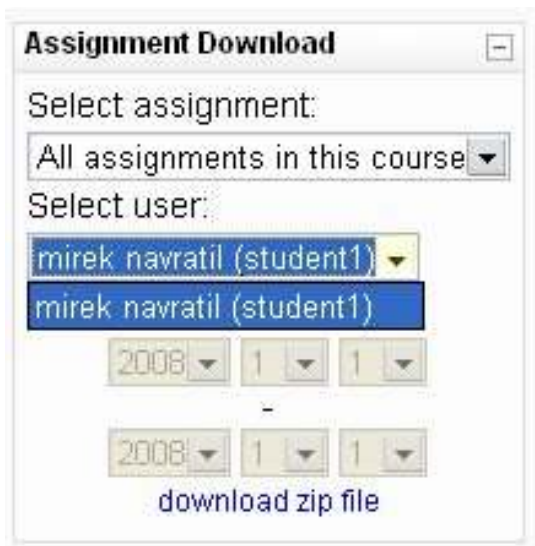
4. Výběr úkolů

9.1.3 2 – výběr uživatele

Tato komponenta umožňuje, ale pouze učitelům v daném kurzu, výběr buď všech uživatelů, kteří jsou v kurzu zapsáni, nebo jen jednoho požadovaného uživatele. Student má v tomto výběru pouze sám sebe, čímž je rozlišena funkcionality modulu pro učitele a studenty.



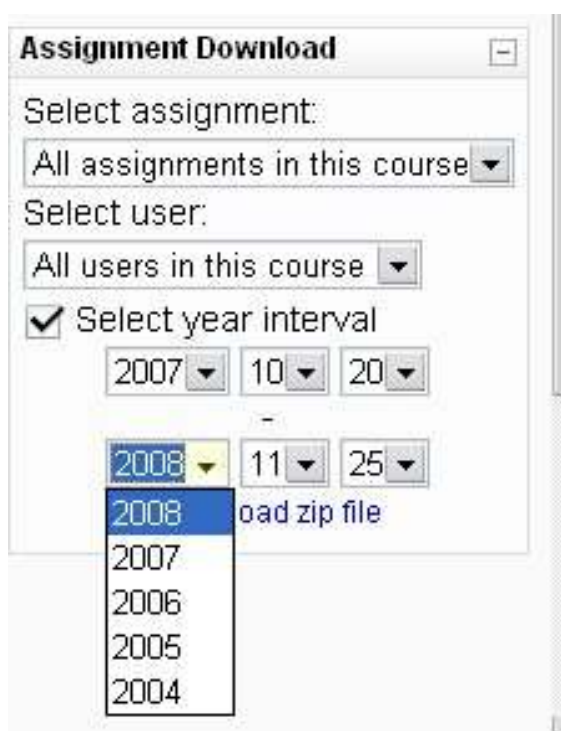
5. Ukázka výběru uživatelů pro učitele



6. Ukázka výběru uživatelů pro studenta

9.1.4 3 - Výběr časového rozmezí

Důležitou komponentou je také časové rozmezí, které ve výsledku vybere jen ty protokoly a úkoly, které byly nahrány do systému v tomto rozmezí. Pokud není aktivováno, jsou vybrány všechny protokoly a úkoly aktuálně zapsaných uživatelů v kurzu. Komponenta se aktivuje zaškrtnutím checkboxu, po němž se aktivují comboboxy s výběrem datum. První (hořejší) datum určuje nejmladší nahrané soubory, druhé datum určuje nejstarší přípustné datum nahraného souboru.

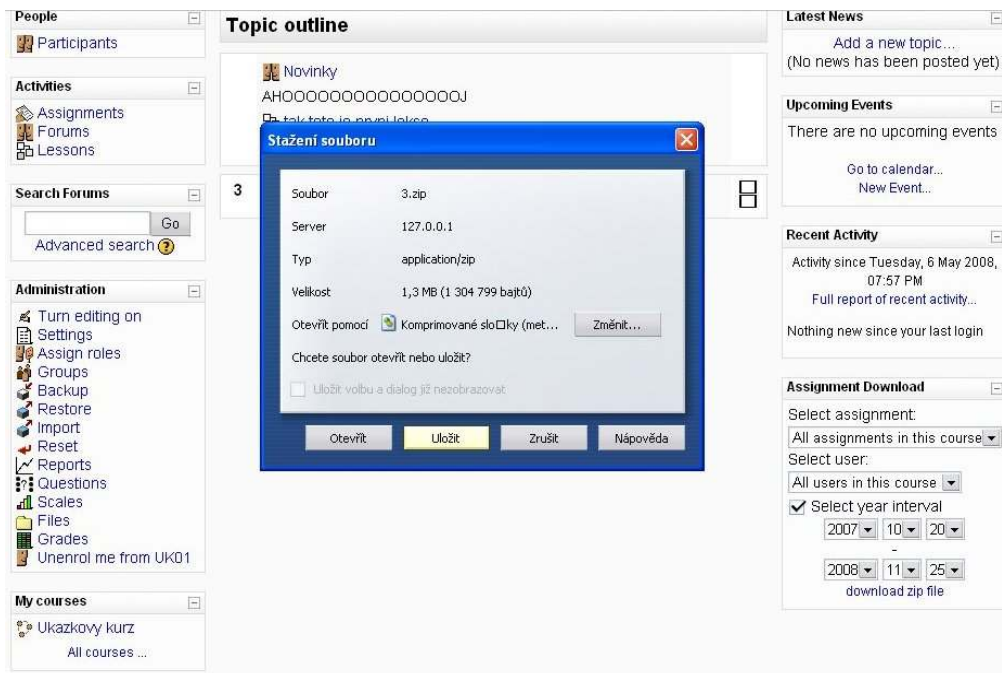


7. Výběr intervalu

Na tomto příkladu budou vybrány všechny protokoly a úkoly všech uživatelů, které byly nahrány od 20.10.2007 do 25.11.2008

9.1.5 4 – Stažení souboru .zip

Poslední komponenta je hypertextový odkaz, který stáhne zip soubor s úkoly a protokoly podle zadaných požadavků – „z kterých úkolů“, „kterých uživatelů“, „v jakém časovém rozmezí“.



8. Stažení výsledného .zip souboru

9.2 Uživatelská příručka v index.html

Ve výsledném zip souboru se nachází navigační dokument index.html, po jehož otevření se nám zobrazí

Vypis z kurzu BC 2008

| Assignments | Members | Firstname | Lastname | File | Date upload |
|-------------|----------|-----------|----------|-----------------------------------|-------------|
| ukol 2 | admin | Správa | Uživatel | DSC00281.JPG | 26.03.2008 |
| ukol 2 | miuan | milan | medlik | DSC00279.JPG | 31.03.2008 |
| ukol 2 | student1 | mirek | navratil | d.html | 06.05.2008 |
| ukol 2 | ucitel | karel | mladek | deprecatedlib.php | 16.04.2008 |
| ukol 4 | admin | Správa | Uživatel | DSC00283.JPG | 26.03.2008 |
| ukol 4 | miuan | milan | medlik | DSC00283.JPG | 31.03.2008 |
| ukol 4 | student1 | mirek | navratil | DSC00283.JPG | 07.04.2008 |

9. Ukázka navigačního dokumentu.

Tento dokument se skládá ze tří komponent, filtrování úkolů, filtrování uživatelů a nakon- nec samotného zobrazení zálohovaných protokolů a úkolů s informacemi o jménu a pří- jmení odevzdávajícího s odkazem na soubor a datum odevzdání.

9.2.1 Možnosti filtrování

Filtrovat lze, jak bylo řečeno výše, jak podle úkolů tak podle uživatelů. Tyto filtry lze kombinovat jak demonstrují následující obrázky.

Assignments filter

No filter

ukol 2

ukol 4

Adresa: javascript:set_filter(0,1,'filter_n_1')

Users filter

No filter

milan medlik (miuan)

karel mladek (ucitel)

mirek navratil (student1)

Správa Uživatel (admin)

| Assignments | Members | Firstname | Lastname | File | Date upload |
|-------------|----------|-----------|----------|-----------------------------------|-------------|
| ukol 2 | admin | Správa | Uživatel | DSC00281.JPG | 26.03.2008 |
| ukol 2 | miuan | milan | medlik | DSC00279.JPG | 31.03.2008 |
| ukol 2 | student1 | mirek | navratil | d.html | 06.05.2008 |
| ukol 2 | ucitel | karel | mladek | deprecatedlib.php | 16.04.2008 |

10. Ukázka filtrování v navigačním dokumentu.

Assignments filter

No filter

ukol 2

ukol 4

| Assignments | Members | Firstname | Lastname | File | Date upload |
|-------------|----------|-----------|----------|------------------------------|-------------|
| ukol 2 | student1 | mirek | navratil | d.html | 06.05.2008 |
| ukol 4 | student1 | mirek | navratil | DSC00283.JPG | 07.04.2008 |

Users filter

No filter

milan medlik (miuan)

karel mladek (ucitel)

mirek navratil (student1)

Správa Uživatel (admin)

11. Ukázka filtrování v navigačním dokumentu

Assignments filter

No filter

ukol 2

ukol 4

| Assignments | Members | Firstname | Lastname | File | Date upload |
|-------------|---------|-----------|----------|------------------------------|-------------|
| ukol 2 | miuan | milan | medlik | DSC00279.JPG | 31.03.2008 |

Users filter

No filter

milan medlik (miuan)

karel mladek (ucitel)

mirek navratil (student1)

Správa Uživatel (admin)

12. Ukázka filtrování v navigačním dokumentu

ZÁVĚR

Úkolem mé bakalářské práce bylo zjistit metody rozšíření Moodle. Možností se nabízí hned několik: Activity modules, Admin reports, Assignment types, Authentication plugins, Blocks, Course formats, atd., více v kapitole 2. Ale pro můj účel se nabízí pouze jediný vhodný modul, a to je modul typu Blocks, který dokáže zobrazit různá data a informace v uživatelské oblasti. Modul je naprogramován tak jako samotný Moodle v jazyku PHP a přes funkci *get_content* obsluženou Moodle je vložen do uživatelské oblasti. Funkcí *get_record_sed* je napojen na databázi Moodle, odkud jsou čerpány informace o odevzdaných úkolech, pro konkrétní kurz, jméno úkolu (assignments), jméno uživatele, a pokud je to požadováno, tak i datum nahrání souboru do systému. Samotné soubory nejsou uloženy v databázi, ale na diskovém oddílu, Modul si k nim vytvoří cestu opět na základě informací z databáze. Tyto soubory jsou shromážděny do jednoho archivu a nabídnuty ke stažení. Součástí archivu je i dokument *index.html*, který slouží k lepší navigaci mezi obsaženými soubory v tomto archivu. Tento dokument pomocí jazyku JavaScript umožňuje filtrování souborů, podle jména úkolů a podle uživatelů tyto filtry se dají libovolně kombinovat a mají za úkol ještě více zvýšit orientaci mezi soubory. Soubory jsou zobrazeny v tabulce s informacemi, jako je jméno a příjmení uživatele, datum odevzdání a v neposlední řadě odkaz na soubor. Tato tabulka se dá dále seřadit podle požadovaného sloupce. Při řešení této bakalářské práce jsem narazil na spoustu dalších vylepšení, jako je lokalizace do více jazyků, zlepšení vzhledu navigačního dokumentu tak, aby se přiblížil více podobě Moodle, a mnoho dalších, avšak s z důvodu časové tísně nebylo možné se věnovat jejich realizaci a veškerá priorita směřovala na funkční stránku modulu. Modul bude uveřejněn také na www.moodle.org, kde se nachází další více či méně specifické moduly a kde bude dál žít svým životem a může být o tato vylepšení obohacen třeba jinými programátory, administrátory či zapálenými stoupenci komunity kolem Moodle, nebo v případě velkého zájmu přímo svým autorem.

SEZNAM POUŽITÉ LITERATURY

- [1] Welling, L., Thomson, L.: PHP a MySQL – rozvoj webových aplikací, SoftPress, 2002
- [2] Kosek, J.: PHP tvorba interaktivních internetových aplikací Podrobný průvodce. 1. vyd. Praha: Grada, 1998 492 s. ISBN: 80-7169-373-1
- [3] Schlossnagle, G.: Pokročilé programování v PHP 5. 1. vyd., Brno: Zoner Press, 2004. 640 s. ISBN: 80-86815-14-5

10 INTERNETOVÉ ZDROJE

[4] **Moodle** [online]. Dostupná z URL: <http://moodle.org/>.

[5] **Moodle Development Documentation** [online]. Dostupná z URL:
http://docs.moodle.org/en/Development:Developer_documentation

[6] **php** [online]. Dostupná z URL: <http://www.php.net/>

[7] **The Apache Software Foundation** [online]. Dostupná z URL: <http://www.apache.org/>

[8] **MySQL** [online]. Dostupná z URL: <http://mysql.com/>

[9] **Softintegration** [online]. Dostupná z URL: <http://www.softintegration.com/>

[10] **Wikipedie HTML** [online]. Dostupná z URL:
http://cs.wikipedia.org/wiki/HyperText_Markup_Language

[11] **Wikipedie PHP** [online]. Dostupna z URL: <http://cs.wikipedia.org/wiki/Php>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

Tabulka 1 Seznam použitých zkratek

| | |
|------|---------------------------|
| PHP | Hypertext Preprocessor |
| SQL | Structured Query Language |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |

SEZNAM OBRÁZKŮ

| | |
|--|----|
| 1. Ukázka umístění bloku..... | 29 |
| 2. Poziční tlačítka v editačním režimu | 29 |
| 3. Popis komponent v bloku..... | 30 |
| 4. Výběr úkolů..... | 30 |
| 5. Ukázka výběru uživatelů pro učitele..... | 31 |
| 6. Ukázka výběru uživatelů pro studenta | 31 |
| 7. Výběr intervalu..... | 32 |
| 8. Stažení výsledného .zip souboru | 33 |
| 9. Ukázka navigačního dokumentu. | 33 |
| 10. Ukázka filtrování v navigačním dokumentu. | 34 |
| 11. Ukázka filtrování v navigačním dokumentu | 34 |
| 12. Ukázka filtrování v navigačním dokumentu | 34 |

SEZNAM TABULEK

| | |
|---|----|
| Tabulka 1 Seznam použitých zkratek..... | 38 |
|---|----|

SEZNAM PŘÍLOH

Příloha P1 Propojení tabulek při dohledávání úkolů

PŘÍLOHA P I: PROPOJENÍ TABULEK PŘI DOHLEDÁVÁNÍ ÚKOLŮ

