

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE

Oddelenie informatizácie a riadenia procesov

Vytvorenie bloku LCZA v e – learningovom systéme MOODLE

Dňa: 18.05.2006

Miesto: Bratislava

Vedúci projektu: Ing. Tomáš Hirmajer

Vypracoval: Peter Šlapanský

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ
TECHNOLÓGIE STU V BRATISLAVE**

Oddelenie informatizácie a riadenia procesov

Číslo: 6/2006

Vec: Zadanie semestrálneho projektu na ukončenie bakalárskeho štúdia.

1. Meno študenta /ky/: Peter Šlapanský

(u vydatých aj rodné)

2. Meno vedúceho projektu: Ing. Tomáš Hirmajer

3. Názov projektu:

/So všetkými podrobnosťami: formulácia úlohy, rozsahu, podmienok riešenia, harmonogramu riešenia a pod./

Vytvorenie bloku LCZA v e – learningovom systéme MOODLE

4. Termín odovzdania záverečnej práce projektu: 20. 05. 2006

5. Záverečná práca projektu sa odovzdáva v 2 zviazaných exemplároch vedúcemu projektu.



Dátum: 20. 02. 2006

**Doc. Dr. Ing. Miroslav Fikar
vedúci oddelenia**

Pod'akovanie

Ďakujem Ing. Tomášovi Hirmajerovi, Doc. Dr. Ing. Miroslavovi Fikarovi a Ing. Ľubošovi Čirkovi za cenné rady a pripomienky, ktoré umožnili vznik tohto projektu.

Obsah

Zoznam použitých skratiek.....	5
Úvod.....	6
1 Databázy.....	8
1.1 Stručný úvod k databázam.....	8
1.2 Základné databázové pojmy.....	8
1.3 Manipulácia s údajmi v databáze.....	9
1.3.1 Príklady.....	9
1.4 Ďalšie funkcie SQL.....	11
2 LAMP.....	12
2.1 LAMP? SAMP? WAMP?.....	12
2.2 PHP.....	12
2.3 Krátka ukážka PHP.....	13
3 Moodle.....	14
3.1 Systémy na správu obsahu.....	14
3.2 Čo je Moodle?.....	14
3.3 Požiadavky pre používanie Moodle.....	14
3.4 Ostatné systémy a Moodle.....	15
3.5 Štruktúra Moodle.....	15
3.5.1 Výhody systému Moodle:.....	16
3.5.2 Nevýhody systému Moodle.....	16
4 Modul Exercisemod.....	17
4.1 Prečo ďalší modul?.....	17
4.2 Požadované vlastnosti modulu.....	17
4.3 Riešenia.....	18
4.3.1 Riešenie úloh 1, 2, 3 a 6.....	18
4.3.2 Riešenie úlohy 5.....	18
4.3.3 Riešenie úlohy 7.....	19
4.3.4 Riešenie úlohy 8.....	19
5 Záver.....	20
6 Zoznam použitej literatúry a referencií.....	21
7 Prílohy.....	22
7.1 Obrázková príloha.....	22

Zoznam použitých skratiek

Skratka	Význam skratky [2]
http	HyperText Transfer Protocol
WWW	World Wide Web
URL	Uniform Resource Locator
TCP/IP	Transmission Control Protocol / Internet Protocol
HTML	HyperText Markup Language
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
SVG	Scalable Vector Graphics
jpeg	Joint Photographic Experts Group
png	Portable Network Graphics
gif	Graphics Interchange Format
CSS	Cascading Style Sheets
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations
RAM	Random Access Memory
SSI	Server Side Includes
CGI	Common Gateway Interface
PHP	PHP: Hypertext Preprocessor
ASP	Active Server Pages
SQL	Structured Query Language
DDL	Data Definition Language)
ANSI	American National Standards Institute
ISO	International Organization for Standardization
Moodle	Modular Object-Oriented Dynamic Learning Environment
CMS	Content Management System
	Course Management System
LMS	Learning Management System
VLE	Virtual Learning Environments
CMC	Education via Computer-mediated Communication
IIS	Internet Information Services
GPL	General Public License

Úvod

S Internetom, sieťou sietí, dnes prichádza do kontaktu pravdepodobne každý. Menej ľudí už vie niečo podrobnejšie, tak z hľadiska hardware, ako aj software. Žiadaný obsah je obvykle pre užívateľov distribuovaný prostredníctvom bezstavového protokolu http (Hypertext Transfer Protocol) a súpravou protokolov TCP/IP (Transmission Control Protocol / Internet Protocol), ktorého najpoužívanejšia v čase písania tejto práce je verzia 1.1. World Wide Web (WWW alebo jednoducho Web) je služba, globálny informačný priestor, ktorý obsahuje dokumenty na všetkých internetových serveroch používajúcich protokol http. URL (Uniform Resource Locator, prakticky adresa tvorená slovami, číslami a interpunkčnými znamienkami) teda nemusí obsahovať „www“, aby bola jeho súčasťou.

Ďalším, pre túto prácu dôležitým prvkom, je skutočnosť, že sa používa model klient-server (klient odošle požiadavku, server ju spracuje a odošle naspäť spracovaný obsah). Klient je napríklad webový prehliadač, ktorý obvykle spracováva údaje a súbory odosielané serverom, ktoré môžu byť rôzneho typu, samotný prehliadač obvykle spracováva len súbory:

- HTML (HyperText Markup Language), XHTML (Extensible HyperText Markup Language), XML (Extensible Markup Language), SVG (Scalable Vector Graphics) - statické dokumenty, okrem SVG, ktorý môže byť animovaný použitím JavaScriptu, resp. ECMAScriptu
- jpeg (Joint Photographic Experts Group), png (Portable Network Graphics), gif (Graphics Interchange Format) - grafické formáty, t. j. bežné obrázky a animácie
- CSS (Cascading Style Sheets), JavaScript (referenčná norma je nazývaná ECMAScript, implementácia v prehliadačoch sa obvykle nazýva JavaScript), XSL/XSLT (Extensible Stylesheet Language / Extensible Stylesheet Language Transformations), ...

Všetky spomenuté textové dokumenty prehliadač interpretuje, v čom sú výhody (napr. jednoduchosť úprav) aj nevýhody (sú pomalšie ako binárne formáty).

Hoci spomínané dokumenty sú klientom prijímané ako statické dokumenty (t. j. obvykle bez možnosti priamej úpravy v prehliadači), veľmi často, najmä v prípade rozsiahlejších projektov, sú generované dynamicky na strane servera podľa prijatých požiadaviek klienta. Na jednej strane to uľahčuje administráciu väčších projektov a umožňuje užívateľom lepšie interagovať s obsahom, na strane druhej to znamená väčšiu záťaž pre servery a z toho vyplývajúce náklady v prípade ich nedostatočného výkonu a jeho zvyšovania. Napriek tomu je veľmi časté, že viacero webových stránok na viacerých webových adresách je obsluhovaných jedným, resp. dvoma servermi. Je to možné vďaka vysokému výkonu dnešných serverov. Tento fakt umožňuje aj pomerne lacné umiestnenie a obsluhovanie webového obsahu.

Ako je tento obsah generovaný dynamicky? Vo väčšine prípadov je viacero možných riešení, z nich najčastejšie je použitie SSI (Server Side Includes) a predovšetkým skriptov. Menej časté je použitie špeciálnych, pre tento účel naprogramovaných a skompilovaných programov. Rozdiel medzi skriptom a programom je v zásade v tom, že skripty sa neprevádzajú do natívneho binárneho kódu len raz, ale sú to, na rozdiel od programov, textové súbory „za letu“ interpretované programom nazývaným interpreter. Z tohto faktu vyplýva, že skripty sú o niečo pomalšie ako programy, ale pre jednoduchosť sú nesmierne populárne a využívané.

Skripty sú najčastejšie používané z niekoľkých dôvodov. Je možné pri nich používať rôzne skriptovacie jazyky, je to veľmi flexibilné, ľahko a rýchlo upraviteľné riešenie, naviac často dostatočne výkonné. Pokiaľ treba výkon naviac, obvykle sa používajú CGI (Common Gateway Interface) skripty, kde sú najčastejšie používané jazyky Perl a Python, výnimočne C/C++ (to sú však programovacie jazyky, nie skriptovacie, aj keď aj tieto sa dajú interpretovať).

Pre bežné skripty sa obvykle používajú jazyky ako PHP (najbežnejší), v menšej miere Perl, Python, Ruby, ASP a iné.

V súčasnosti sú zdrojom údajov pre dynamické generovanie obsahu buď obyčajné textové súbory, XML dokumenty alebo databázy.

1 Databázy

1.1 Stručný úvod k databázam

Databázy sú jedným z najčastejšie používaných zdrojov údajov pre dynamické generovanie stránok pre Web. Dôvodov je viacero.

Jednou z hlavných výhod oproti obyčajným textovým súborom, resp. XML súborom je skutočnosť, že väčšinou sa obsah databázy načítava do operačnej pamäte (RAM) servera, pokiaľ je jej dostatočné množstvo. RAM je podstatne rýchlejšia ako pevné disky i diskové polia, predovšetkým v latencii a dátovej priepustnosti. Vďaka tomu môže byť načítanie údajov z databázy podstatne rýchlejšie ako obdobný proces z textových súborov. Táto výhoda sa čiastočne stiera v prípade, ak je server poddimenzovaný pre danú aplikáciu a musí využívať odkladací priestor na pevnom disku – swap.

1.2 Základné databázové pojmy

Rýchlosť však nie je jedinou výhodou databázy, najmä pokiaľ sa jedná o tzv. relačnú databázu. Táto využíva relačný model správy databáz, ktorý je založený na tom, že všetky údaje v tejto databáze sú reprezentované ako matematické n-ticové vzťahy.

Prakticky takéto databázy ukladajú údaje do tabuliek (nazývajú sa aj entity), medzi ktorými sú určité vzťahy. Tabuľka má stĺpce nazývané atribúty a údaje sa ukladajú do riadkov (t. j. riadky obsahujú samotné hodnoty všetkých atribútov). Riadky sa nazývajú aj záznamy. Atribúty majú svoje názvy, definované obvykle pri návrhu štruktúry samotnej databázy, ako aj ich typ, ktorý je napr. reťazec, číslo, dátum,

Správu údajov v databáze zabezpečuje relačný databázový systém.

Základným princípom relácií medzi jednotlivými tabuľkami danej databázy je priradenie jednoznačného identifikátora každému riadku danej tabuľky. Tento identifikátor sa nazýva primárny kľúč a je to hodnota alebo množina hodnôt. Môže to byť napríklad identifikačné číslo študenta alebo knihy.

Obmedzenia (constraints) sú pravidlá, ktoré obmedzujú tabuľky a ich atribúty tak, aby sa dalo vyhnúť prípadným problémom na základe istej podmienky.

Index je dátová štruktúra, ktorá sa používa pre rýchly prístup k riadkom tabuľky. Obvykle sa zostavuje nad primárnym kľúčom tabuľky a potom môžeme pomocou neho rýchlo vyhľadať konkrétny riadok tabuľky. Indexy sa tiež zostavujú a definujú pre iné atribúty, ktoré sa často používajú v databázových otázkach (dotazoch).

Napriek tomu, že dotaz je nespisovný čechizmus, je zaužívaný aj u nás v slovnom spojení databázový dotaz. Z toho dôvodu ho budem aj používať v texte tejto práce.

Normalizovaná databáza je správne navrhnutá databáza, vytvorená z modelu entít a vzťahov.

1.3 Manipulácia s údajmi v databáze

Na komunikáciu s databázou a manipuláciu s údajmi v databáze sa vo väčšine prípadov používa jazyk SQL (Structured Query Language), čo je množina príkazov na správu databáz, tabuliek a údajov v nich. Existujú aj alternatívy, tie ale neboli pri tomto projekte využité.

SQL je ANSI (American National Standards Institute) a ISO (International Organization for Standardization) štandardom, stáva sa však, že v rámci daného databázového systému obsahuje pre daný systém vlastné rozšírenia tohto jazyka. Poslednou revíziou jazyka SQL je SQL:2003, nie je však voľne prístupná, ale môže byť zakúpená od organizácií ANSI a ISO, avšak za pomerne veľký poplatok.

Pre správu databázy existuje podmnožina jazyka SQL zvaná DDL (Data Definition Language).

Najrozšírenejšie databázové systémy pre webové aplikácie sú MySQL, PostgreSQL, Firebird, MSSQL a zriedkavo, pre veľmi veľké systémy, aj databázy Oracle Database od Oracle Corporation, ktorá je firmou, ktorá ako prvá prišla s komerčnou relačnou databázou. Niekedy sa, samozrejme, používajú aj iné databázy.

1.3.1 Príklady

Pokiaľ sú príkazy vykonávané z interpretera danej databázy cez CLI (Command Line Interface, príkazový riadok), na koniec riadku sa obvykle vkladá bodkočiarka.

Vytvorenie databázy:

Požije sa príkaz:

```
CREATE DATABASE nazov_databazy
```

kde `nazov_databazy` sa nahradí požadovaným názvom databázy.

Použitie databázy:

Databázovému systému je potrebné oznámiť, s ktorou jeho databázou chceme pracovať. Vtedy použijeme:

```
use nazov_databazy
```

kde sa opäť `nazov_databazy` sa nahradí názvom požadovanej databázy.

Vytvorenie tabuľky:

Jedna z tabuliek pre modul Exercisemod v LMS Moodle.

```
CREATE TABLE `moodle_exercisemod` (  
  `id` int(10) unsigned NOT NULL auto_increment,  
  `course` int(10) unsigned NOT NULL default '0',  
  `name` varchar(255) NOT NULL default '',  
  `nelements` tinyint(3) unsigned NOT NULL default '1',  
  `phase` tinyint(3) unsigned NOT NULL default '0',  
  `gradingstrategy` tinyint(3) unsigned NOT NULL default '1',  
  `usemaximum` tinyint(3) unsigned NOT NULL default '0',  
  `assessmentcomps` tinyint(3) unsigned NOT NULL default '2',  
  `anonymous` tinyint(3) unsigned NOT NULL default '1',  
  `maxbytes` int(10) unsigned NOT NULL default '100000',  
  `deadline` int(10) unsigned NOT NULL default '0',  
  `timemodified` int(10) unsigned NOT NULL default '0',  
  `grade` tinyint(3) NOT NULL default '0',  
  `gradinggrade` tinyint(3) NOT NULL default '0',  
  `showleaguetable` tinyint(3) unsigned NOT NULL default '0',  
  `usepassword` tinyint(3) unsigned NOT NULL default '0',  
  `password` varchar(32) NOT NULL default '',  
  PRIMARY KEY (`id`),  
  KEY `course` (`course`)  
) COMMENT='Defines exercisemod'
```

V tejto tabuľke sú definované atribúty, ich názvy, typy, maximálna veľkosť, predvolené hodnoty, obmedzenie určujúce, či daný atribút je alebo nie je povinný. Ako primárny kľúč je definovaný identifikátor `id`, ktorý je celým číslom, maximálne desať miestnym a pri každom vložení riadku sa automaticky zväčší o hodnotu 1.

Zmena tabuľky:

```
ALTER TABLE `moodle_exercisemod` zmena
```

kde `zmena` predstavuje kód pre samotnú zmenu.

Vkladanie a odstraňovanie údajov:

Vloženie údajov:

```
INSERT INTO `moodle_exercisemod` VALUES (hodnoty)
```

kde hodnoty sú hodnoty pre každý atribút danej tabuľky.

Odstránenie údajov:

```
DELETE FROM `moodle_exercisemod` WHERE id = 1
```

Tento príkaz odstráni z tabuľky `moodle\_exercisemod` všetky riadky, v ktorých je hodnota atribútu `id` rovná číslu 1.

Aktualizácia údajov:

```
UPDATE `moodle_exercisemod` SET phase = 2 WHERE id = 1
```

kde aktualizujeme riadok s atribútom `id` rovným jednej, pričom meníme hodnotu atribútu `phase` na 2.

1.4 Ďalšie funkcie SQL

Medzi funkcie databázových systémov patria aj:

- dotazy so spájaním tabuliek - technika, pri ktorej sa kombinujú riadky dvoch alebo viacerých tabuliek na základe spájacej podmienky, zadanej v klauzule `WHERE` [1], napr.:

```
SELECT atribut1, atribut2  
FROM tabulka1, tabulka2  
WHERE tabulka1.id = tabulka2.id  
ORDER BY atribut1;
```

kde `ORDER BY` zoraďuje nájdené údaje podľa `atribut1`

- matematické funkcie (napr. `AVG()` pre priemer, `SUM()` pre súčet, `COUNT()` pre počet riadkov v množine a iné)
- časové funkcie (pre vkladanie časových údajov)
- aritmetické operátory a ďalšie funkcie

2 LAMP

2.1 LAMP? SAMP? WAMP?

Ako ste si mohli všimnúť, tri slová v nadpise majú isté znaky spoločné. Jedná sa totiž o skratky „balíkov“.

AMP je obvykle skratka pre kombináciu Apache, MySQL a PHP (alebo Perl, resp. Python).

Apache je jeden z najrozšírenejších webových serverov, má niekoľko významných výhod, tou najdôležitejšou je obvykle jeho cena. Je totiž zadarmo aj so svojimi zdrojovými kódmi. Navyše sa jedná o pomerne robustný server, ktorý sa pomerne jednoducho konfiguruje a je veľmi dobre rozšíriteľný prostredníctvom množstva modulov, z ktorých väčšina je pod rovnakou licenciou ako Apache.

MySQL je systém pre správu databáz používajúci SQL. Jeho licencia je duálna, buď GNU GPL (General Public License), teda ako free software, alebo je dostupný pod klasickou komerčnou licenciou pre prípady, v ktorých použitie GNU GPL nie je možné, napr. kvôli nekompatibilitě licencií. Jedná sa o jeden z najrozšírenejších systémov vďaka licencií, postačujúcemu výkonu a možnostiam.

PHP je rekurzívna skratka pre „PHP: Hypertext Preprocessor“ [3]. Je to skriptovací jazyk na strane servera pre dynamické generovanie webových stránok. Jeho syntax je založená na jazykoch C, Java a Perl. Navyše obsahuje preň špecifické funkcie. Prostredníctvom rozšírení je možné tento jazyk využívať v spojení s databázami a aj na generovanie iného obsahu, ako čistého textu.

Počiatkové písmená v skratkách LAMP, SAMP a WAMP značia operačný systém, v tomto prípade Linux, Solaris, Windows. Neznamená to, samozrejme, že na iných systémoch AMP nebeží, môže, napríklad na variantoch systému BSD, a iných systémoch. Najčastejšie sa AMP používa s operačnými systémami Linux, FreeBSD (resp. OpenBSD alebo NetBSD) a Solaris. Systém Microsoft Windows obsahuje, resp. je možné doň doinštalovať IIS – Internet Information Services, čo je skupina služieb pre Internet pre servery používajúce Microsoft Windows. Je to druhý najpopulárnejší web server po serveri Apache.

2.2 PHP

Jazyk PHP sa stal známym a používaným pomerne rýchlo. Spočiatku najmä medzi amatérmi a nadšencami, neskôr, keď sa stal vyspelejším, prenikol aj medzi profesionálov.

Jeho silnými stránkami sú

- značná jednoduchosť

- veľké množstvo funkcií, ktoré jazyky nižšej úrovne, ale často ani iné skriptovacie jazyky, neponúkajú
- slušná rýchlosť interpretera
- pomerne veľké množstvo rozšírení
- nie je potrebné sa starať o uvoľňovanie pamäte
- typovanie premenných pri ich vytváraní
- dobrá flexibilita pri práci s poliami a objektmi

Jazyk bol navrhnutý pre dynamickú tvorbu webového obsahu, čomu nasvedčujú aj niektoré vstavané funkcie. Že je na to naozaj vhodný a používaný je badateľné už len z ponuky web hostingových služieb (prenajímajú priestor na svojich serveroch a iné), ktoré obsahujú podporu PHP prakticky vždy.

Jeho hlavnou nevýhodou je často nerovnaká syntax podobných funkcií, takže je nevyhnutné siahnuť po (vynikajúcom) manuále pre PHP pomerne často.

2.3 Krátka ukážka PHP

Ako prvé je potrebné uviesť, že na to, aby bol kód PHP interpretovaný ako taký, je potrebné mať správne nakonfigurovaný webserver a v súboroch samotných je potrebné interpreteru uviesť, ktoré časti súboru má interpretovať. Akonáhle má interpreter vymedzenú oblasť pôsobenia, začína pracovať.

Príklad:

Jednoduchý výpis svetoznámeho „Hallo World!“:

```
<?php
$string = 'Hallo World!';
echo $string;
?>
```

To, čo uvidíme v prehliadači bude čistý, neformátovaný text: Hallo World!, ktorý sme vypísali z premennej `$string`.

Nie je to síce veľa, ale príjemne jednoducho demonštruje jednoduchosť jazyka. Čo budeme potrebovať sú premenné, polia a objekty a operátory.

Vzhľadom na obmedzený priestor tejto prezentácie nechám ďalšie vysvetľovanie na literatúre a Webe.

3 Moodle

3.1 Systémy na správu obsahu

Systém na správu obsahu (Content Management System, CMS) je systém umožňujúci jeho administrátorom, resp. oprávneným užívateľom, spravovať obsah daného webového sídla bez znalostí (X)HTML a programovania, resp. tvorby skriptov.

Takéto systémy šetria čas a peniaze potrebné na platenie programátorov a dizajnérov, aby robili zásahy priamo v kóde systému.

Tieto systémy sú často upravované na špeciálne účely alebo sú pre ne priamo písané. V prípade riešenia úpravou všeobecného CMS sa tak zvykne robiť vytvorením modulu (ak daný systém moduly podporuje).

Pokiaľ sa jedná o systémy priamo písané na tento účel, obvykle dostávajú iný všeobecný názov ako Content Management System. Bývajú nazývané Learning Management System (LMS), Course Management System (CMS), Virtual Learning Environments (VLE), education via computer-mediated communication (CMC) alebo jednoducho Online Education (e-learning).

3.2 Čo je Moodle?

Moodle je systém priamo určený pre výučbu cez Internet, ktorý pôvodne vytvoril Martin Dougiamas na základe niektorých jeho štúdií, ktoré uskutočnil. Pôvodne to bol hobby projekt, ktorý prerástol do jeho PhD práce a momentálne sa mu venuje na plný úväzok.

Moodle však už nie je len projektom jediného človeka, na jeho vývoji sa podieľa viacčlenný tím. Jeho aktuálne zloženie môžete nájsť na stránkach systému Moodle.

Okrem iného je Moodle skratka pre „Modular Object-Oriented Dynamic Learning Environment“ [4], aj keď pôvodne znamenalo M „Martin“. Ako naznačuje sám názov, intenzívne sú využívané aj objektovo orientované vlastnosti jazyka PHP.

3.3 Požiadavky pre používanie Moodle

Moodle nekladie veľké nároky na software servera, vyžaduje:

- Webový server podporujúci PHP (obvykle Apache alebo IIS)
- PHP
- Databázový server (priamo podporované a odporúčané sú MySQL a PostgreSQL)

Vývojová platforma Moodle je LAMP, z čoho vyplýva, že najlepšie testovaný by mal byť práve na tejto platforme. Pokiaľ je však prítomná kombinácia AMP, rozdiely by mali byť minimálne.

Pre optimálny beh je rozumné mať na serveri nainštalované tieto rozšírenia PHP:

- Knižnica GD a FreeType 2 (pre Unixové a Linuxové) pre dynamickú tvorbu grafov, ktoré vytvára stránka so záznamami.
- mbstring pre zaobchádzanie s multi-bytovými reťazcami (pre Moodle 1.6 je odporúčané aj iconv)
- mysql pre použitie MySQL databáz
- pgsql pre použitie Postgre databáz
- zlib pre manipuláciu so zip súbormi
- ďalšie moduly, ktoré môžu byť potrebné pre voliteľnú funkčnosť Moodle

3.4 Ostatné systémy a Moodle

Moodle, samozrejme, nie je jediný systém tohto typu, ale je jedným z mála kvalitných systémov tohto typu pod licenciou GNU GPL.

Ostatné systémy s otvoreným zdrojovým kódom sú napr. ATutor, Dokeos, Interact, Site@School a iné. V krátkosti zhrnieme len hlavné výhody z vlastných skúseností s Moodle. Z technického hľadiska sa Moodle dá považovať za veľmi dobre premyslený a fungujúci systém s pomerne elegantnou a zároveň aj relatívne jednoduchou architektúrou.

3.5 Štruktúra Moodle

- config.php – obsahuje základné nastavenia. Tento súbor nie je dodávaný s Moodle – administrátor Moodle ho vytvára.
- install.php – skript, ktorý administrátor spustí na vytvorenie config.php
- version.php – definuje súčasnú verziu Moodle
- index.php – prvá stránka sídla
- admin/ - kód pre administráciu celého servera
- auth/ - zásuvné moduly pre autentifikáciu užívateľov
- blocks/ - zásuvné moduly pre malé bočné bloky na mnohých stránkach
- calendar/ - kód pre správu a zobrazovanie kalendárov
- course/ - kód pre správu a zobrazovanie kurzov
- doc/ - pomocná dokumentácia pre Moodle
- files/ - kód pre správu a zobrazovanie súborov prenesených na server
- lang/ - texty v rôznych jazykoch, jeden adresár pre jeden jazyk
- lib/ - knižnice kódu jadra Moodle

- login/ - kód pre prihlasovanie sa na účet a vytváranie účtov
- mod/ - sú tu všetky hlavné moduly kurzov Moodle
- pix/ - všeobecná grafika stránky
- theme/ - témy pre úpravu vzhľadu stránky
- user/ - kód pre správu a zobrazovanie užívateľov

3.5.1 Výhody systému Moodle:

Z technického hľadiska:

- vysoká modulárnosť
- premyslená a dobre fungujúca architektúra
- ľahká upraviteľnosť vzhľadu
- prijateľná upraviteľnosť funkčnosti prostredníctvom modulov
- jednoduchá inštalácia modulov

Z hľadiska pohodlia užívateľa:

- jednoduchá a rýchla registrácia
- rozličné množstvo vzdelávacích aktivít
- prehľadnosť
- užívateľská prívetivosť
- možnosti spätnej väzby
- jednoduché odovzdávanie zadaní
- jednoduché a rýchle známkovanie
- lokalizácia jazykových súborov

3.5.2 Nevýhody systému Moodle

Z technického hľadiska:

- Prakticky jediná nevýhoda, badateľná počas práce so systémom je jeho náročnosť na výkon servera pri väčšom množstve užívateľov, aj keď aj v tejto oblasti sa radí Moodle minimálne medzi lepšími prímermi.

Z hľadiska pohodlia užívateľa neboli zaznamenané vážnejšie nedostatky ani problémy.

4 Modul Exercisemod

4.1 Prečo d'alší modul?

Počas používania systému Moodle na oddelení informatizácie a riadenia procesov sa ukázalo, že momentálne existujúce moduly pre Moodle, ktoré sú oficiálne podporované a stiahnuteľné zo stránok Moodle, nevyhovujú plne všetkým požiadavkám oddelenia. Preto bola zadaná úloha vytvoriť alebo upraviť existujúce moduly, ktoré boli najbližšie požadovaným vlastnostiam oddelenia. Ukázalo sa, že existujú dva moduly, ktoré mali blízko k požadovanej funkcionalite. Sú to moduly:

- Assignment (Zadanie)
- Exercise (Cvičenie)

Počas testovania a skúmania týchto modulov sa prišlo k záveru, že najbližšie požadovaným vlastnostiam má modul Exercise, preto modul Exercisemod vznikol jeho úpravou, ako napovedá aj jeho názov.

4.2 Požadované vlastnosti modulu

Požiadaviek bolo niekoľko, medzi hlavné patria:

1. Možnosť dávkového uploadu súborov na server, podľa možnosti v jedinom skomprimovanom archíve pre jedno cvičenie
2. Automatické rozbalenie daného archívu
3. Automatické priradenie jednotlivých variantov úloh do daného cvičenia
4. Náhodné pridelenie jedného z variantov Cvičenia jednému zo študentov prihlásených do kurzu (vyriešené v module Exercise)
5. Možnosť okamžite zobrazíť odoslané riešenie v zozname odovzdaných riešení, ako aj hodnotenie odovzdaného riešenia učiteľom študentovi
6. Možnosť učiteľov zhladať aj vzorové riešenie pri hodnotení daného riešenia
7. Preklad 25 súborov pomocníka v module Exercise (z ktorých takmer všetky sú prenositeľné do Exercisemod)
8. Možnosť „rozbiť“ hodnotenie daného riešenia do niekoľkých častí (čiastočne riešené v module Exercise)

4.3 Riešenia

Riešenie daných úloh vyžadovalo vytvorenie dvoch nových skriptov pre jednoduchšiu údržbu a úpravu viacerých skriptov v rámci celého modulu.

4.3.1 Riešenie úloh 1, 2, 3 a 6

Pre vyriešenie týchto úloh bol vytvorený skript `zipstuff.php`, ktorý vykonáva spracovanie zip archívu presunutého na server prostredníctvom webového formulára v tomto module, a skript `showres.php`, ktorý generuje HTML kód pre zobrazenie riešenia pri hodnotení riešenia.

Spracovanie tohto archívu spočíva v jeho rozbalení do dočasného adresára, ktorý sa podľa potreby automaticky vytvorí, a následnej „simulácii uploadu“. Ide o využitie funkcií z jadra modulu a z jadra Moodle na presun jednotlivých súborov na ich miesto a následnú registráciu ako variantov daného cvičenia.

Zároveň s presunom daného variantu cvičenia sa presúva aj súbor so vzorovým riešením, ktoré má rovnaký názov ako daný variant cvičenie, ale navyše obsahuje predponu „res_“, do adresára `resolutions` v dátovom adresári `exercisemod` v rámci adresára daného cvičenia.

Na základe zhodného mena súboru zadania a riešenia je potom na stránke s hodnotením vygenerovaný odkaz na patričný súbor so vzorovým riešením, ktoré môže vyučujúci porovnať s odovzdaným riešením študenta.

Navyše bol kvôli prehľadnosti upravený aj skript zabezpečujúci výpis dostupných cvičení a bol upravený a pridaný kód pre zobrazenie celkového počtu zadaní, resp. vyriešených zadaní.

Pre správne fungovanie skriptov `zipstuff.php` a `showres.php` je nevyhnutné dodržať dve zásady:

- V zip archíve so zadaniami a riešeniami nesmú byť adresáre ani nadbytočné súbory
- Súbor s riešením má predponu „res_“ (bez úvodzoviek) a inak úplne rovnaký názov ako má daný variant cvičenia

4.3.2 Riešenie úlohy 5

Riešenie tejto úlohy bolo najjednoduchšie zo všetkých zadaných úloh a to práve vďaka dobre napísaným skriptom modulu `Exercise`.

Riešenie spočívalo jednoducho v priradení nulovej hodnoty premennej, ktorá mala obsahovať čas, počas ktorého je možné odoslané texty a hodnotenia v danom module upravovať a počas ktorého ešte nie je zobrazená iným užívateľom.

4.3.3 Riešenie úlohy 7

Riešenie tejto úlohy bolo naopak časovo pomerne náročné, predovšetkým preto, že bola snaha o dodržanie zavedeného názvoslovia, ktoré použil predchádzajúci prekladateľ pri preklade častí rozhrania modulu Exercise.

Celkovo pozostávala úloha z preloženia 25 súborov pomocníka do slovenčiny z angličtiny.

4.3.4 Riešenie úlohy 8

Posledná úloha, ktorá má umožniť hodnotiť ľubovoľný počet elementov zadania.

Pre tento účel bola vytvorená nová bodovacia stratégia, ktorá bola nazvaná aditívna bodovacia stratégia. Táto je príbuznou stratégiou nahromadenej, ale na rozdiel od nej nakladá inak s váhami pridelenými jednotlivým elementom zadania. Pokiaľ sú totiž váhy pri aditívnej stratégii pri všetkých prvkoch rovnaké, dôjde ku sčítaniu bodov za jednotlivé elementy. Nahromadená stratégia namiesto toho spraví z týchto bodov priemer. Je to dané použitým spôsobom výpočtu bodovania.

Naviac je pri stratégii aditívnej možné stanoviť ľubovoľný dvojciferný počet bodov pre daný element.

5 Záver

Cieľom tejto práce bolo vytvoriť modul pre e-learningový systém Moodle, čo bolo splnené.

Tento modul by mal byť v ďalšom semestri používaný oddelením informatizácie a riadenia procesov pri výučbe Laboratórnych cvičení zo základov automatizácie (LCZA), dúfam však, že to nebude jeho jediný prístav. Uvažuje sa aj o nasadení modulu v rámci predmetu Optimalizácia na tomto oddelení.

Informácie o skorších verziách projektu boli zverejnené na fórach na moodle.org. Predpokladá sa, že finálna verzia, predstavená ako samostatný modul, vyvolá väčšiu odozvu.

Proces tvorby tohto modulu bol pomerne zdĺhavý, v súčasnosti sa na ňom vykonávajú len drobné úpravy a vylepšenia. Navyše sa autor tohto modulu naučil pracovať s cudzím kódom podstatne efektívnejšie, predovšetkým s objektovo orientovaným kódom.

V budúcnosti, pokiaľ bude modul nasadený do „ostrej“ prevádzky a pokiaľ oň neprejavia záujem iní užívatelia systému Moodle, prejde pod správu oddelenia informatizácie a riadenia procesov. V prípade, že bude o modul širší záujem, je pravdepodobné, že budem vo vývoji alebo aspoň údržbe modulu pokračovať, pokiaľ bude širší záujem pretrvávať.

Navyše je možné, že pokiaľ budú mať implementované úpravy dostatočný ohlas, budú časti týchto úprav pridané do pôvodného modulu Exercise.

Odkaz na stiahnutie kompletného modulu nájdete v zozname použitej literatúry a referencií.

V časti Príloha na konci tejto práce sa nachádza obrázková príloha s ukázkami systému a modulu.

6 Zoznam použitej literatúry a referencií

- [1] PHP a MySQL vytváříme webové databázové aplikace, Hugh E. Williams & David Lane, Computer Press 2002, 143 00, Praha 4, <http://www.cpress.cz/>, str. 102 – 118
- [2] <http://en.wikipedia.org/>, Copyright © Wikimedia Foundation, Inc.
- [3] <http://www.php.net/>, Copyright © The PHP Group
- [4] <http://www.moodle.org/>, Copyright © Moodle Pty Ltd.

Ďalšie referencie:

- Časti kódu použité z: <http://slapocms.slapo.net/>, Copyright © Peter Šlapanský
- Stránky autora Moodle: <http://dougiamas.com/>, Copyright © Martin Dougiamas
- Adresa pre stiahnutie modulu: <http://www.slapo.net/>, sekcia FChPT

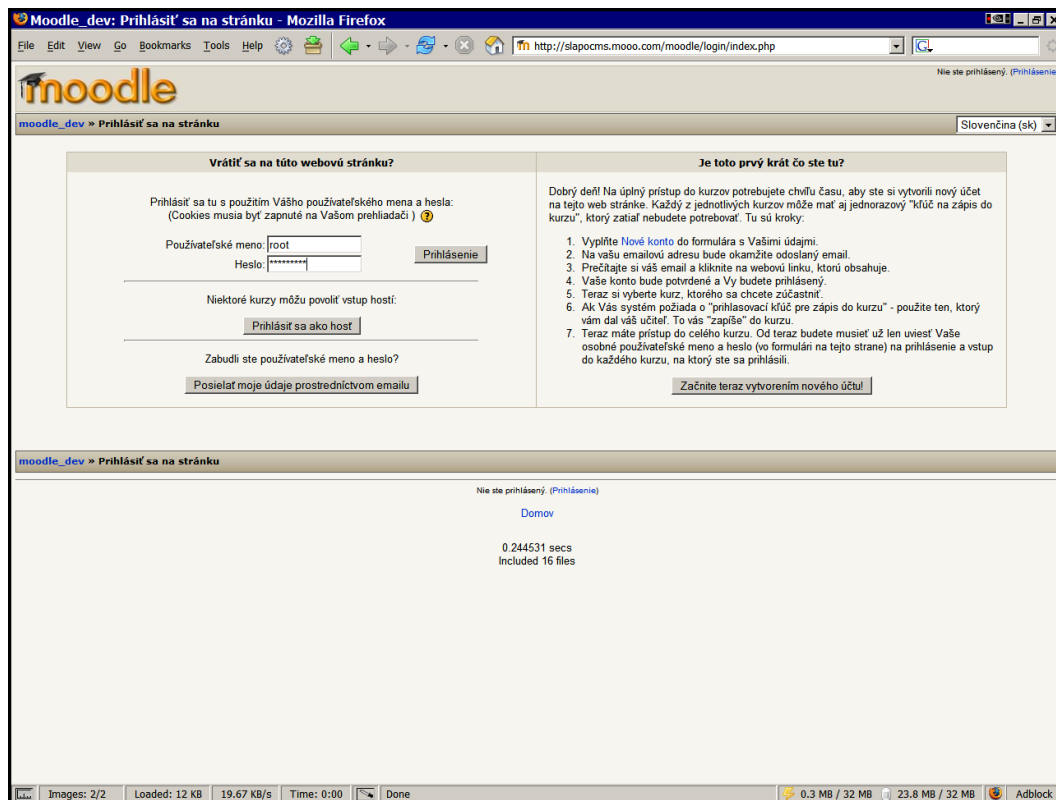
7 Prílohy

Pre zdrojový kód modulu navštívte jeho stránky, na ktorú je uvedený odkaz v zozname použitej literatúry a referencií.

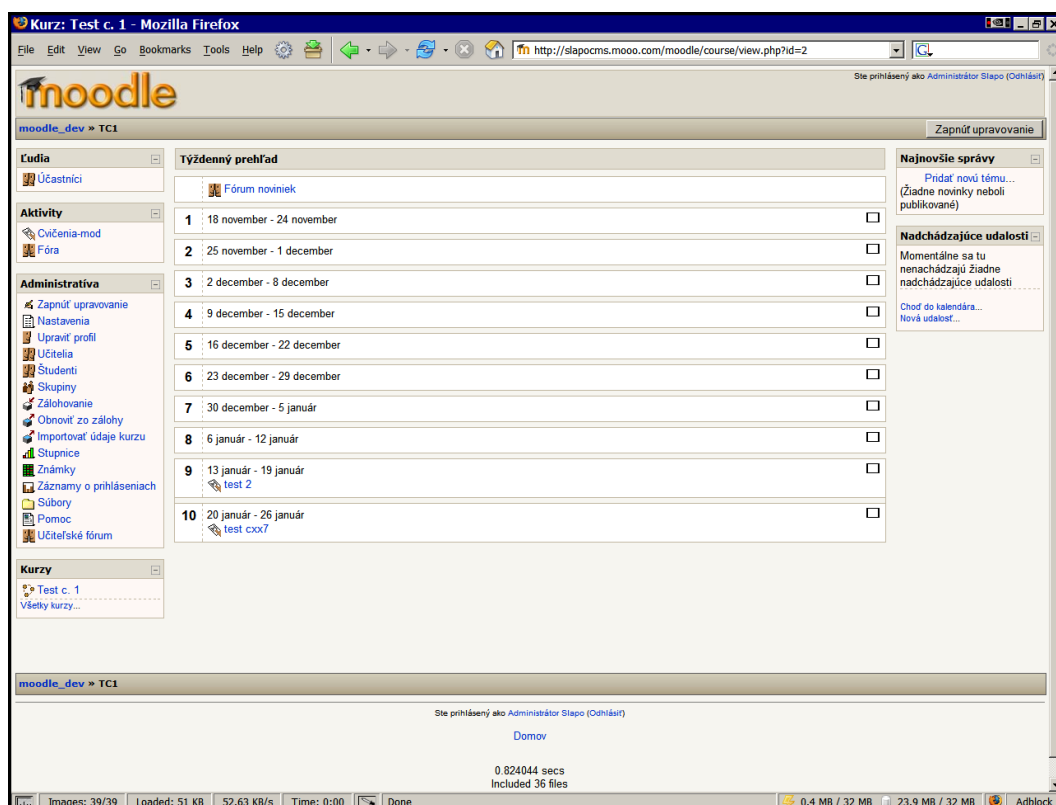
7.1 Obrázková príloha



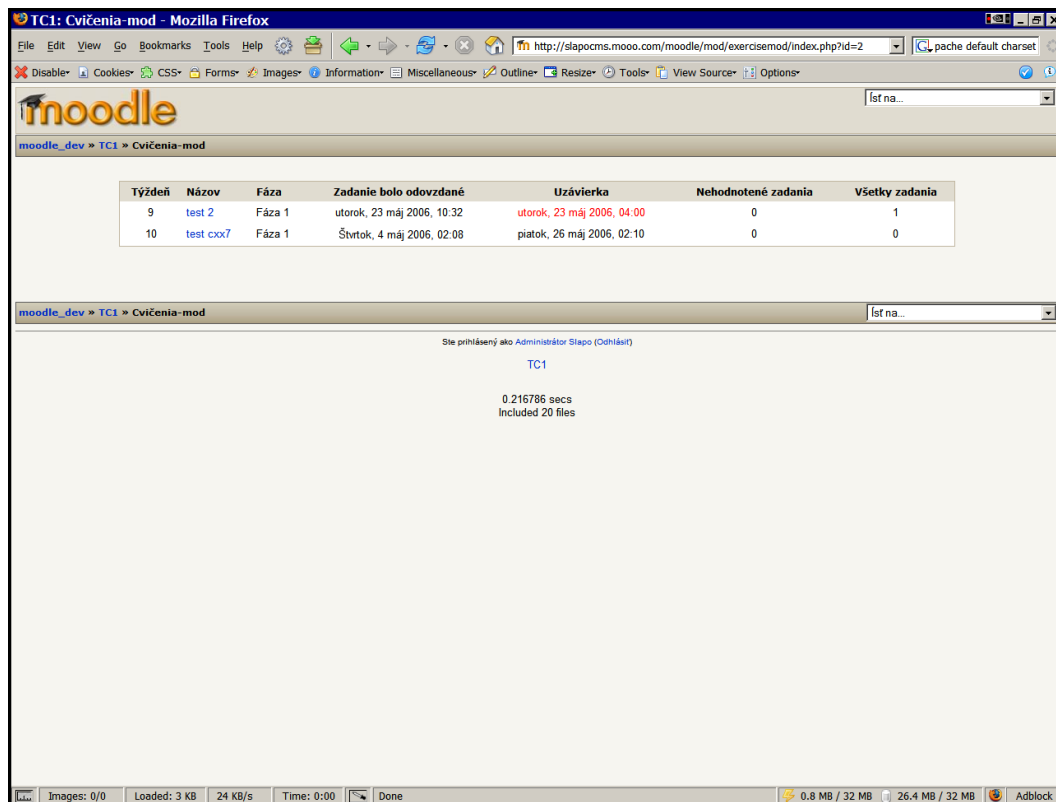
Obrázok 1: Ukážka prvej strany v Moodle



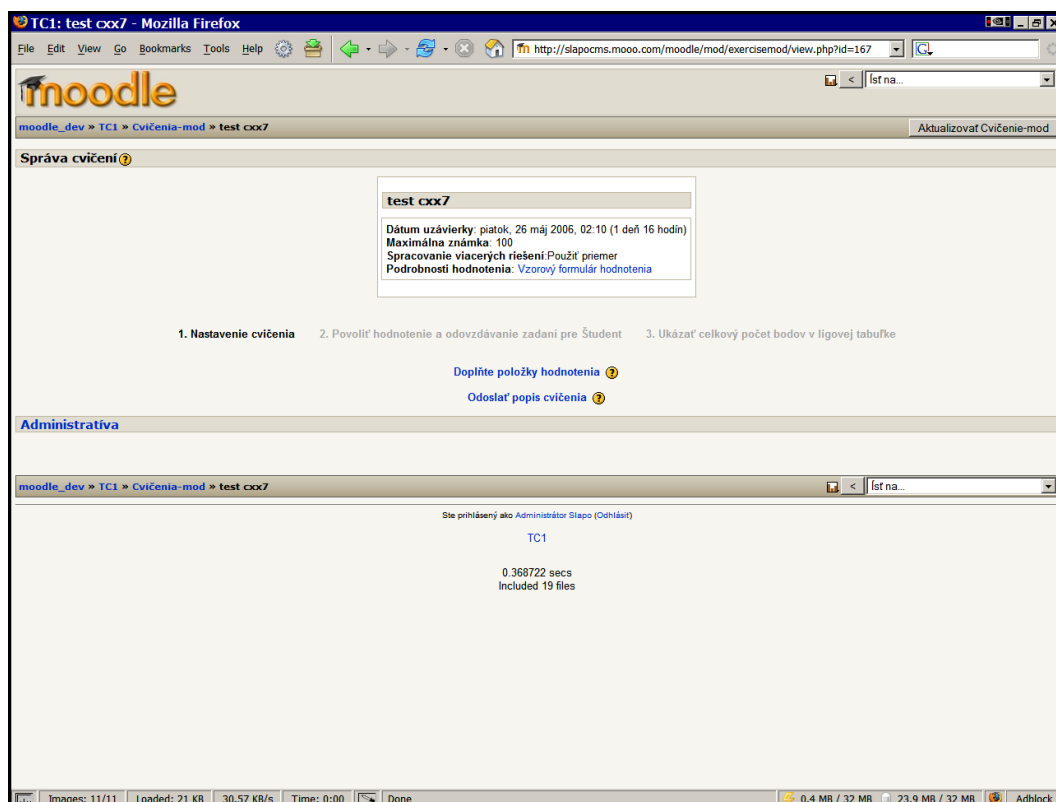
Obrázok 2: Prihlasovanie sa do systému



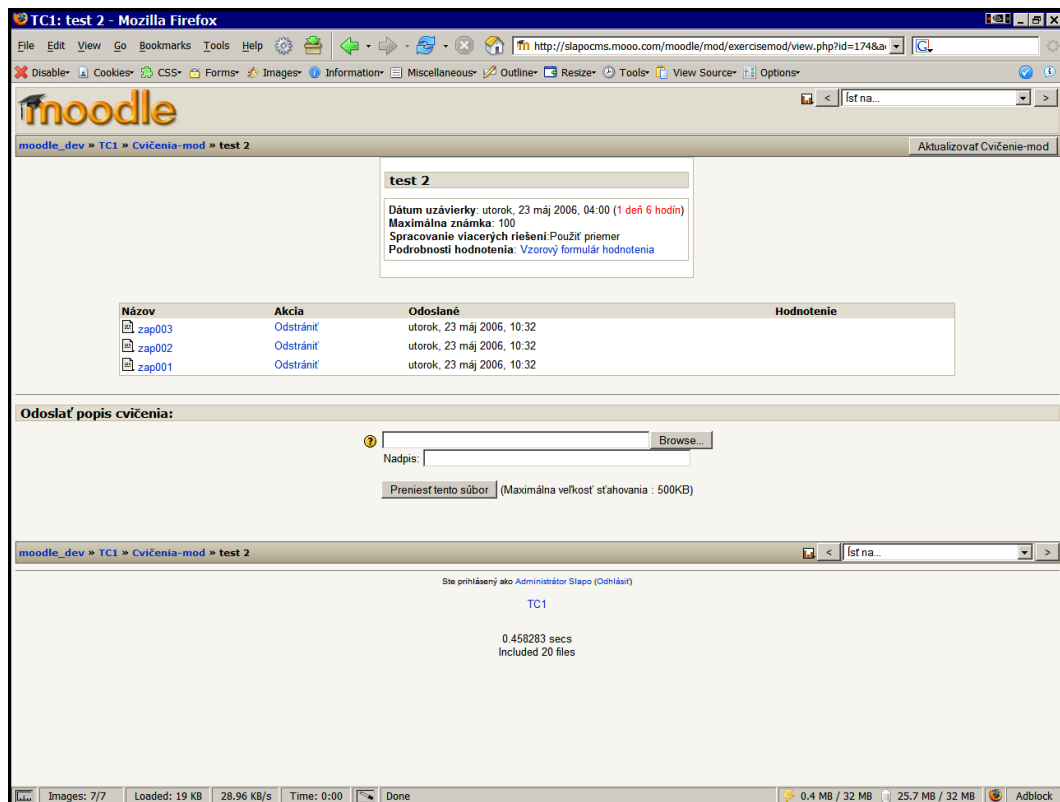
Obrázok 3: hlavná strana kurzu



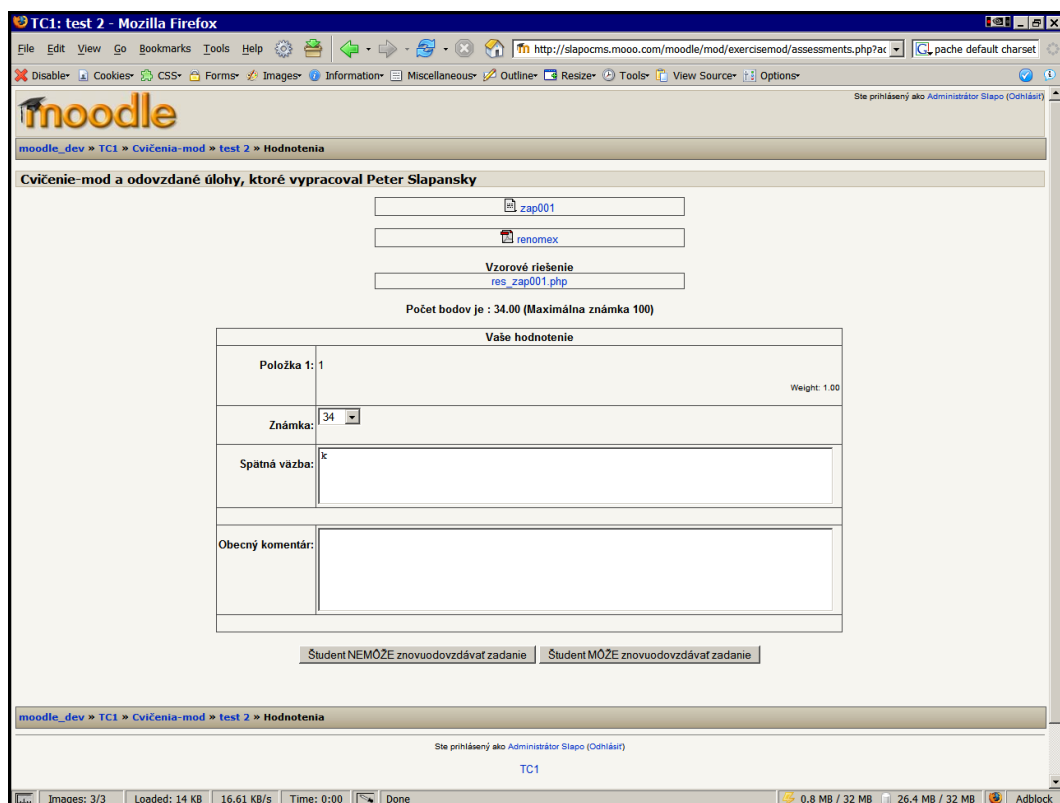
Obrázok 4: Zoznam cvičení v danom kurze



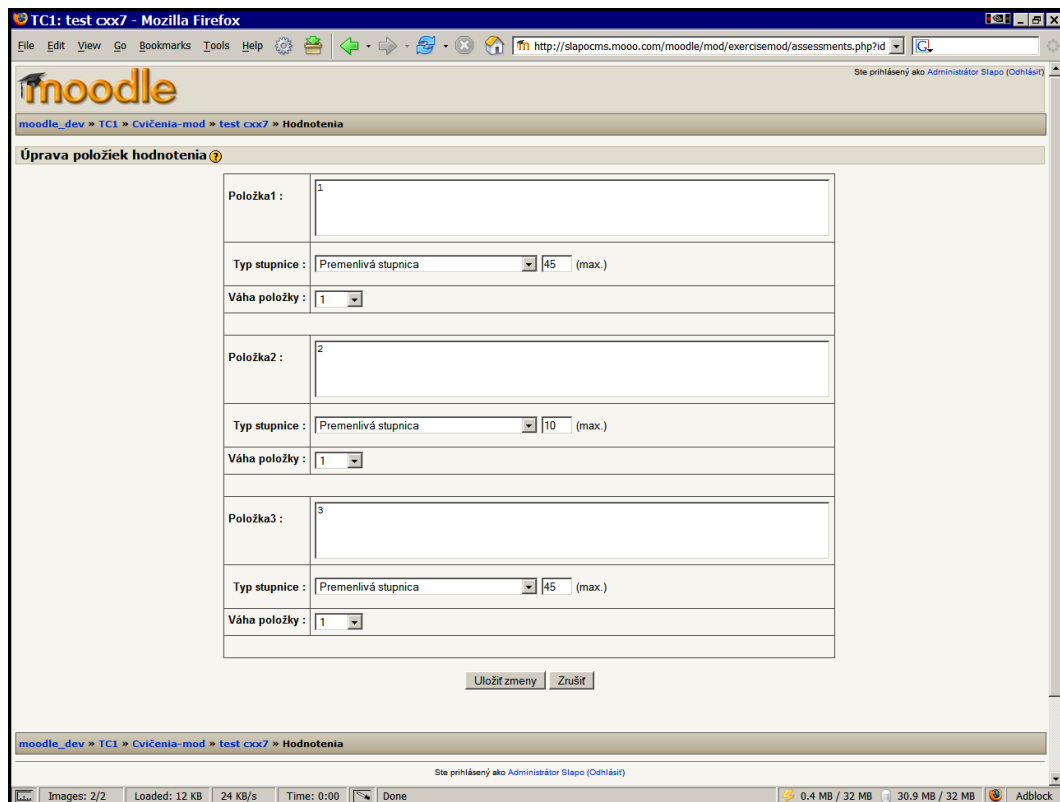
Obrázok 5: Samotné cvičenie.



Obrázok 6: Formulár pre upload súborov so zoznamom súborov zadání cvičenia na serveri



Obrázok 7: Hodnotenie riešenia študenta



Obrázok 8: Úprava položiek hodnotenia študenta