

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÁ APLIKACE PRO VÝUKU FORMÁLNÍCH JAZYKŮ
A PŘEKLADAČŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

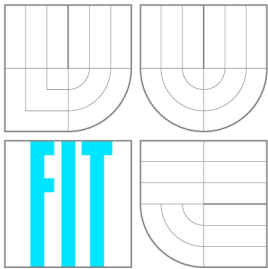
AUTOR PRÁCE
AUTHOR

FILIP BRYCHTA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

WEBOVÁ APLIKACE PRO VÝUKU FORMÁLNÍCH JAZYKŮ A PŘEKLADAČŮ

WEB APPLICATION FOR FORMAL LANGUAGE AND COMPILER TEACHING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

FILIP BRYCHTA

Ing. JIŘÍ TECHET

BRNO 2007

Zadání bakalářské práce

Řešitel: **Brychta Filip**
Obor: Informační technologie
Téma: **Webová aplikace pro výuku formálních jazyků a překladačů**
Kategorie: Databáze

Pokyny:

1. Seznamte se s požadavky kladenými na webovou aplikaci určenou pro výuku formálních jazyků a překladačů.
2. Seznamte se s jazykem XHTML, kaskádovými styly CSS, databázovým serverem MySQL, jazykem PHP a jeho SQL dialektem pro přístup k databázovému serveru, a dále potom s formátem RSS.
3. Proveďte návrh informačního systému s využitím prostředků UML. Věnujte pozornost různým rolím uživatelů (neautorizovaný uživatel, student, administrátor), jejich oprávněním a zabezpečenému přístupu k informačnímu systému. Systém musí umožňovat sledování změn na stránkách pomocí RSS kanálů.
4. Implementujte navržený systém s použitím výše uvedených nástrojů.
5. Diskutujte přednosti a nedostatky vašeho řešení a nastiňte další možný vývoj vašeho projektu.

Literatura:

- Welling, L., Thomsonová, L.: PHP a MySQL - rozvoj webových aplikací, Softpress 2003, ISBN 8086497607.

Při obhajobě semestrální části projektu je požadováno:

1. Seznamte se s požadavky kladenými na webovou aplikaci určenou pro výuku formálních jazyků a překladačů.
2. Seznamte se s jazykem XHTML, kaskádovými styly CSS, databázovým serverem MySQL, jazykem PHP a jeho SQL dialektem pro přístup k databázovému serveru, a dále potom s formátem RSS.
3. Proveďte návrh informačního systému s využitím prostředků UML. Věnujte pozornost různým rolím uživatelů (neautorizovaný uživatel, student, administrátor), jejich oprávněním a zabezpečenému přístupu k informačnímu systému. Systém musí umožňovat sledování změn na stránkách pomocí RSS kanálů.

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

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

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

Vedoucí: **Techet Jiří, Ing.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Filip Brychta**
Id studenta: 84446
Bytem: Deštné v Orlických horách 59, 517 91 Deštné v Orlických.h.
Narozen: 30. 07. 1984, Opočno
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Webová aplikace pro výuku formálních jazyků a překladačů
Vedoucí/školitel VŠKP: Tchet Jiří, Ing.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užit, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnožení.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.


Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel



Autor

Abstrakt

Tato práce se zabývá vývojem webové aplikace vhodné pro výuku formálních jazyků a překladačů. Aplikace je implementována z části jako redakční systém a z části jako datový sklad. Hlavním cílem je usnadnit přístup k materiálům týkajícím se této problematiky ze strany nejširší veřejnosti. Umožňuje snadné šíření informací jak v textové podobě (komentáře, diskuze, aktuality), tak i prostřednictvím uploadu/downloadu dokumentů.

Klíčová slova

informační systém, webová aplikace, databáze, redakční systém, PHP, MySql, CSS, XHTML, RSS, Atom

Abstract

This thesis deals with the development of a web application used for teaching of formal languages and compilers. The application is implemented partly as an content management system and partly as a data store. Its main purpose is to simplify the access to the related materials for the public. It enables a simple distribution of information both in the textual form (comments, discussions, news) and also by upload/download of documents.

Keywords

information system, web application, database, content management system, PHP, MySql, CSS, XHTML, RSS, Atom

Citace

Filip Brychta: Webová aplikace pro výuku formálních jazyků a překladačů, bakalářská práce, Brno, FIT VUT v Brně, 2007

Webová aplikace pro výuku formálních jazyků a překladačů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Techeta.

.....
Filip Brychta
13. května 2007

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu práce panu Ing. Jiřímu Techetovi za konzultace a užitečné rady. Dále Oldřichu Bittnerovi za pomoc při řešení grafického návrhu aplikace.

© Filip Brychta, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Základní pojmy	4
2.1	Informační systémy	4
2.1.1	Obecný systém	4
2.1.2	Redakční systémy	4
2.2	Prostředky využívané při tvorbě IS	5
2.2.1	Skriptovací jazyk PHP	5
2.2.2	MySql	6
2.2.3	Apache	6
2.2.4	phpMyAdmin	7
2.2.5	HTML	7
2.2.6	XHTML	8
2.2.7	CSS	9
2.2.8	JavaScript	9
3	Analýza a návrh	11
3.1	Prohlášení o cílech	11
3.2	Požadavky na systém	11
3.3	Model případu užití	13
3.3.1	Vztahy mezi uživateli	13
3.3.2	Typy uživatelů	13
3.4	ER Diagram	18
3.5	Návrh databáze	20
3.5.1	Popis jednotlivých tabulek	21
4	Implementace	27
4.1	Struktura kódu	27
4.2	Popis jednotlivých částí systému	27
4.2.1	Registrace	28
4.2.2	Přihlášení do systému	28
4.2.3	Zobrazení a vyhledávání dokumentů	28
4.2.4	Komentářový subsystém	29
4.2.5	Diskuze	29
4.2.6	Aktuality	29
4.2.7	Upload souborů	29
4.2.8	Testy	30
4.2.9	Administrace systému	30

4.2.10 Sledování novinek prostřednictvím RSS	30
4.2.11 Uživatelská nastavení	30
4.3 Použité prostředky	31
5 Závěr	32
Seznam použitých zdrojů	33
Seznam příloh	34
A Zdrojový kód pro vytvoření databázových tabulek	35
B Návod na instalaci	38
B.1 Vytvoření databáze	38
B.2 Vytvoření databázových tabulek	38
B.3 Vytvoření adresářové struktury na webovém serveru	38
C Uživatelská příručka	40
C.1 Registrace	40
C.2 Odebírání novinek	40
C.3 Vyhledávání	40
C.4 Upload souborů	42
C.5 Tvorba testů	42
C.6 Editace testů	43
C.7 Administrace systému	43
D Různé pohledy uživatelů	44
D.1 Pohled přihlášeného uživatele na dokumenty	45
D.2 Pohled přispěvatele na editaci svého testu	46
D.3 Pohled administrátora na uživatele	47

Kapitola 1

Úvod

Webové aplikace s prvky redakčního systému jsou dnes velmi rozšířené především díky své univerzálnosti. Profesionální redakční systémy lze použít téměř pro jakoukoliv webovou aplikaci informačního typu. Aplikace, které se věnuje tento dokument je kombinací redakčního systému a datového skladu. Je určena pro výuku formálních jazyků a překladačů, ačkoli by díky své univerzálnosti mohla být použita pro různá jiná témata. Cílem bylo vytvořit jednoduchou, pro uživatele přátelskou webovou aplikaci, která by umožňovala z uživatelského pohledu snadný přístup k materiálům týkajícím se problematiky formálních jazyků a překladačů, možnost tyto materiály komentovat a diskutovat o nich a možnost sledovat aktuální informace prostřednictvím kanálu RSS. Z pohledu přispěvatele (uživatel oprávněn pro vkládání dokumentů a aktualit) zajistit snadnou správu dokumentů (upload dokumentu na server, vkládání a editace informací o dokumentu) a snadné vkládání a editaci aktualit. Z pohledu administrátora by měla tato aplikace umožňovat snadnou správu uživatelských účtů a samotného obsahu (odstraňování nevhodných příspěvků). Aplikace je implementována v jazyku PHP a XHTML s využitím kaskádových stylů (CSS). Jako databázový server je použit systém MySQL.

Následující text se věnuje popisu jednotlivých kroků vývoje této aplikace. V první části je uvedena teorie nezbytná pro pochopení problému. Tato část vysvětluje základní pojmy týkající se informačních systémů a popisuje prostředky používané při vývoji a provozu webové aplikace. Další část obsahuje analýzu problému. Ta spočívá v získávání požadavků na vlastnosti systému od zadavatele a následné formální specifikaci těchto požadavků ve formě diagramu případu užití a jeho popisu. Následuje kapitola věnující se návrhu aplikace. V této části je uveden ER diagram a jeho popis, je zde popsán způsob transformace ER diagramu na databázové tabulky a tyto jednotlivé databázové tabulky jsou tu popsány. Kapitola implementace obsahuje popis způsobu řešení jednotlivých částí aplikace a použité prostředky při jejím vývoji. Poslední kapitola obsahuje celkové zhodnocení projektu a diskutuje možnosti jeho rozšíření.

Kapitola 2

Základní pojmy

V této kapitole budou popsány základní pojmy a prostředky využívané při tvorbě a provozu informačního systému.

2.1 Informační systémy

Informační systémy jsou systémy pro sběr, udržování, zpracovávání a poskytování informací a dat [10]. Data mohou být jakékoliv zaznamenané poznatky či fakta, které jsou schopné přenosu, uchovávání, interpretace či zpracování. Pojem informace pak může být definovaný jako data, ke kterým si člověk přiřadí význam. Informační systémy jsou využívány v mnoha oblastech. Běžné jsou různé ekonomické informační systémy, informační systém banky, knihovny, restaurace a mnoho dalších. Příkladem ekonomického systému může být systém, který zajišťuje evidenci osob, výpočty mzdy, evidence majetku, evidence materiálu na skladě a manipulace s ním, podvojný účetnictví a další.

2.1.1 Obecný systém

Systém lze chápat jako množinu prvků a vazeb mezi nimi, které jsou účelově definovány na nějakém nosiči [1]. Nosičem v systému může být knihovna, úřad nebo škola. Další možná definice systému: systém je účelově definovaný soubor komponent, mezi kterými existují určité vztahy, a které splňují nějaký cíl. Obecný systém se skládá ze vstupní části, kudy do systému zdroje vstupují, výstupní části, kudy zdroje ze systému vystupují. V části mezi vstupem a výstupem probíhají transformace těchto zdrojů. V některých systémech může být aplikována zpětná vazba. Tato vazba přivádí výstupní zdroje zpět na vstup a tím je ovlivněno chování celého systému. Podle interakce s okolím můžeme systémy dělit na systémy otevřené a uzavřené. Otevřené systémy jsou v interakci s okolím, uzavřené v interakci s okolím nejsou. Interakce s okolím znamená, že je systém k okolí připojen tokem zdrojů. Informační systémy jsou otevřené a typicky mají zpětnou vazbu.

2.1.2 Redakční systémy

Výraz redakční systém se používá ve smyslu anglického Content management system (systém pro správu obsahu). V dnešní době tyto systémy zajišťují správu nejčastěji webového obsahu. Redakční systém představuje aplikaci, která spravuje informace různého charakteru a zároveň se stará i o jejich efektivní využití a zobrazení na některém z předpřipravených výstupů [2]. Výstupem takového redakčního systému jsou například dynamicky generované

HTML stránky, ke kterým uživatelé přistupují prostřednictvím webových prohlížečů. Tyto systémy umožňují snadné vkládání a modifikaci článků prostřednictvím WYSIWYG editoru, vkládání novinek, komentářů k článkům, nebo vytváření různých anket. Dále zajišťují jednoduchou správu uživatelů a jejich práv, sběr statistických údajů, možnost downloadu přístupných článků a souborů a mnoho dalších funkcí, které se mohou v různých redakčních systémech lišit. V dnešní době jsou dostupné jak komerční, tak i freeware aplikace. Jednodušší redakční systémy mohou být naprogramovány v JavaScriptu, řada aplikací používá skriptovací jazyk PHP v kombinaci s databázovým systémem MySQL. Současné redakční systémy jsou natolik vyspělé a do značné míry i univerzální, že je lze využít při řešení téměř všech typů informačních webů.

2.2 Prostředky využívané při tvorbě IS

Následuje popis prostředků, které jsou v dnešní době hojně využívány při tvorbě a provozu informačního systému s webovým rozhraním. Tyto prostředky byly použity při tvorbě webové aplikace, kterou popisuje druhá část této práce.

2.2.1 Skriptovací jazyk PHP

PHP (Hypertext Preprocessor, původně Personal Home Page). Tento jazyk byl vytvořen Rasmussem Lerdorfem v roce 1994. Po přidání jeho Form Interpreteru vzniklo PHP/FI (toto vydání je považováno za PHP verze 2). V roce 1997 dva Izraelští vývojáři, Zeev Suraski a Andi Gutmans, přepsali parser a zformovali tak základ PHP 3. Od této chvíle se začal používat název Hypertext Preprocessor. Oficiálně pro veřejnost byl PHP 3 zpřístupněn v červnu roku 1997. Poté začali s vývojem nového skriptovacího enginu Zend, který se stal v roce 2000 jádrem verze PHP 4. Rok 2004 přinesl PHP 5 pohaněný novým Zend Engine II. Tato verze s sebou přináší podporu objektového přístupu, lepší podporu MySQL, podporu SQLite, ošetření chybových zpráv použitím výjimek a další (viz [9]). Poslední stabilní verze je PHP 5.2.1, která vyšla 8. února 2007. Vývoj probíhá na bázi otevřeného kódu. V současné době je stále vyvíjen celosvětovou komunitou vývojářů.

Tento jazyk je určený především pro programování dynamických internetových stránek. Nejčastěji je začleněn přímo do struktury jazyka HTML či XHTML, což je velmi výhodné při tvorbě webových aplikací. Tento jazyk se stal velmi oblíbeným díky jeho jednoduchosti, přístupnosti a spolehlivosti. PHP je nezávislý na platformě, takže skripty fungují bez úprav na mnoha operačních systémech. Díky velmi častému spojení operačního systému Linux, webového serveru Apache, databázového systému MySQL a PHP se vžila zkratka pro toto spojení LAMP. V této kombinaci je velmi často využíván pro tvorbu webových aplikací.

PHP skripty jsou prováděny PHP procesorem na straně serveru. Způsob integrace PHP do webového serveru se může u různých serverů lišit. Například u HTTP serveru Apache je PHP integrován jako přídatný modul, který se dynamicky připojuje k serveru. Při požadavku klienta na nějaký dokument PHP procesor rozpozná typ dokumentu a pokud se jedná o PHP kód, je tento kód procesorem zpracován a výstup je odeslán do prohlížeče k uživateli. Pokud se jedná o běžný dokument (HTTP, obrázek), je odeslán přímo do prohlížeče. Model PHP lze přirovnat k ostatním skriptovacím jazykům prováděným na straně serveru, jako je ASP.NET od Microsoftu, JavaServer Pages od Sun Microsystems. Všechny tyto jazyky dynamicky generují obsah pro klienta z webového serveru.

2.2.2 MySQL

MySQL je relační databázový systém vytvořený švédskou firmou MySQL AB. Je dostupný jak pod bezplatnou licenci GPL, tak pod placenou komerční licenci. První verze se objevila v roce 1995, první verze pro operační systém Windows v roce 1998. V současnosti je ke stažení pro různé operační systémy. Poslední vydaná verze je MySQL 5.1, ke stažení je i testovací verze MySQL 5.2 Alpha (viz [8]).

Systém MySQL je hojně využíván ve webových aplikacích. Jeho popularita je úzce spjata s popularitou jazyku PHP, se kterým je velmi často kombinován. Podle MySQL AB je po celém světě přes 10 milionů instalací. Každá databáze v MySQL je tvořena jednou nebo více tabulkami. V jednotlivých řádcích tabulky jsou záznamy. Sloupce obsahují atributy. Komunikace s databází, práce s tabulkami a daty se provádí prostřednictvím dotazů. Dotazy vycházejí z deklarativního programovacího jazyka SQL. Podobně jako u ostatních SQL databází se jedná o dialekt tohoto jazyka s různými rozšířeními. V MySQL je možnost výběru z několika typů (storage engine) databázových tabulek. Základní typ MyISAM je nejpoužívanější, ale nepodporuje mnoho potřebných vlastností jako jsou transakce a cizí klíče. Dalším hojně používaným typem je InnoDB, tento typ už podporu pro transakce a cizí klíče obsahuje. Podrobný popis těchto i dalších typů lze nalézt v dokumentaci. MySQL bylo od počátku zaměřeno především na rychlost, některé důležité vlastnosti jako jsou uložené procedury, trigger a pohledy, byly doplněny až nedávno ve verzi 5.0. Online zálohování pro všechny storage engine, XML funkce, plánování událostí a další užitečné vlastnosti byly přidány ve verzi MySQL 5.1. Do budoucna (verze 5.2) se plánuje například paralelismus a nový storage engine nazvaný Falcon.

2.2.3 Apache

Apache HTTP server je multiplatformní, free/open source webový server. V současné době je to celosvětově nejpoužívanější webový server. Poslední měření z března 2007 uvádí, že Apache obsluhuje 58% všech webových stránek (viz [3]).

V době svého vzniku byl Apache jedinou možnou free/open source alternativou k webovému serveru od Netscape Communications Corporation (dnes známému jako Sun Java System Web Server). První verze Apache známá jako NCSA HTTPd byla vytvořena v National Center for Supercomputing Applications s hlavním programátorem Robertem McCoolem v roce 1993. V polovině následujícího roku Robert McCool opustil vývojářskou skupinu a další vývoj začal stagnovat, až byl úplně zastaven. Správci webových serverů, kteří NCSA HTTPd používali, začali psát své vlastní rozšíření. Brian Behlendorf a Cliff Skolnick založili emailovou konferenci pro sběr a koordinaci těchto úprav. První veřejná verze 0.6.2 byla vydána v roce 1995. Následovalo kompletní přepsání zdrojových kódů a založení Apache Group, která je dnes základem vývojářského týmu. Další vývoj se zaměřil na přenositelnost, lepší podporu ostatních operačních systémů a podporu IPv6. Poslední vydaná verze je Apache 2.2.4.

Apache je určen k obsluze jak statických, tak i dynamických webových stránek. Přidáváním různých modulů je možné rozšířit základní funkcionalitu serveru. Velmi častým rozšířením je skriptovací jazyk PHP. Konfigurace Apache se děje prostřednictvím konfiguračního souboru httpd.conf. Existuje i několik grafických rozhraní umožňujících více intuitivní konfiguraci serveru. Virtuální hostování dovoluje na jedné instalaci Apache hostit

více domén zároveň. Apache je použit i v různých balíčcích spolu s jiným softwarem. Tyto balíčky poskytují jednoduchou instalaci všech potřebných komponent pro provoz například webových aplikací. Hlavním konkurentem je v současné době Internet Information Services od Microsoftu.

2.2.4 phpMyAdmin

Je nástroj implementovaný v jazyce PHP umožňující jednoduchou správu obsahu databáze MySQL prostřednictvím webového rozhraní. V současné době je to jeden z nejpopulárnějších nástrojů pro správu databáze, který je dostupný už v 52 jazycích.

phpMyAdmin byl vyvíjen Tobiasem Ratschillerem od roku 1998. Když v roce 2000 opustil projekt z nedostatku času, phpMyAdmin už byl zaběhlým a oblíbeným nástrojem pro správu MySQL databází. Aby bylo možné pokračovat ve vývoji a koordinovat přibývajících patche, založila v roce 2001 skupina tří vývojářů The phpMyAdmin Project a převzala kontrolu nad dalším vývojem. Poslední stabilní verze je phpMyAdmin 2.10.0.2 vydaná 2. 3. 2007 (viz [4]).

2.2.5 HTML

HTML (Hypertext Markup Language) je značkovací jazyk z rodiny SGML (Standard Generalized Markup Language). *Hypertext* je běžný text, jehož prostřednictvím se můžeme dostat k bližším informacím o popisovaném tématu. *Markup Language* (značkovací jazyk) je jazyk, jehož zdrojový text obsahuje jak samotný text, tak i instrukce, jak tento text zpracovat. HTML je využíván především k tvorbě webových stránek.

Tim Berners-Lee navrhl v roce 1990 jazyk HTML a protokol HTTP pro jeho šíření po síti. Neexistovala žádná specifikace, byl to jen soubor nedokonale definovaných elementů. Postupem času s růstem internetu se mu věnovala čím dál větší pozornost. První verze obsahovala 22 elementů z nichž 14 zůstalo dodnes. V roce 1991 byl zprovozněn první web. První grafický prohlížeč byl vyvinut v roce 1993 a zaznamenal obrovský úspěch. Nastal obrovský rozmach internetu a bylo nutné pro HTML definovat standardy (viz [6]).

Standardy HTML:

- **HTML 2.0** - říjen 1995, první verze odpovídající syntaxi SGML.
- **HTML 3.2** - leden 1997, pro svou složitost nebyla tato specifikace nikdy přijata.
- **HTML 4.0** - prosinec 1997, přidány nové prvky pro tvorbu tabulek a formulářů. Nově standardizovány rámy. Některé elementy byly zavrženy.

Ve třech verzích:

- Strict - Nepovoluje zavržené elementy.
- Transitional - Umožňuje použít zavržené elementy.
- Frameset - Pro použití rámu.
- **HTML 4.01** - prosinec 1999, opravuje chyby předchozí verze a přidává některé nové značky. Je to poslední verze jazyka HTML, který už se dál nevyvíjí. Jeho nástupcem je jazyk XHTML. Stejně jako HTML 4.0 ve třech verzích. Strict, transitional a frameset.

Pro každou verzi existuje definice pravidel DTD, na kterou musí být od verze 4.01 uveden v HTML dokumentu odkaz.

Struktura HTML je tvořena značkami (tagy), mezi které se vkládá text. Tyto značky určují význam (sémantiku) textu mezi nimi. Značky mohou obsahovat atributy, které upřesňují vlastnosti elementu. Uvnitř elementu může být vnořen další element. Značky jsou ohraničeny úhlovými závorkami a mohou být párové i nepárové.

2.2.6 XHTML

XHTML je nástupcem jazyka HTML, jehož vývoj byl ukončen verzí 4.01. Narozdíl od svého předchůdce se jedná o dokument XML. Jedním z důvodů vzniku XHTML je možnost snadného parsování dobře formovaného dokumentu využitím obecných XML prostředků. HTML tuto možnost nenabízel.

Cílem první specifikace XHTML 1.0 bylo upravení jazyka HTML tak, aby vyhovoval podmínkám tvorby XML dokumentů a přitom byla zachována zpětná kompatibilita. Jsou definovány tři verze: Strict (stejně jako HTML 4.01 Strict, ale řídí se syntaktickými pravidly XML), Transitional (stejně jako HTML 4.01 Transitional, ale řídí se syntaktickými pravidly XML, podporuje vše co XHTML 1.0 Strict a navíc několik zastaralých elementů) a Frameset (stejně jako HTML 4.01 Frameset, ale řídí se syntaktickými pravidly XML).

Hlavní rozdíly oproti HTML jsou:

- Dokument musí být dobře formovaný (well formed) a veškeré elementy musí být uzavřeny.
- Názvy všech značek a atributů jsou case sensitive a musejí být definovány malými písmeny.
- Hodnoty atributů musejí být uzavřeny do uvozovek.

Specifikace XHTML 1.1 vychází z XHTML 1.0 Strict. Byla zavedena modularizace, což umožňuje vytvořit co nejvhodnější jazyk pro danou aplikaci. Jsou odstraněny všechny zavržené elementy a definice vzhledu je možná pouze prostřednictvím CSS. Tento typ dokumentu je nutno odesílat jako MIME type application/xhtml+xml.

XHTML 2.0 je stále ve vývojovém stádiu, tato specifikace už nebude zpětně kompatibilní.

Některé vlastnosti, které přinese specifikace XHTML 2.0:

- HTML formuláře budou nahrazeny XForms.
- HTML rámy budou nahrazeny XFrames.
- DOM Events budou nahrazeny XML Events, které používají XML Document Object Model.
- Každý element bude moci pracovat jako odkaz.

2.2.7 CSS

CSS (Cascading Style Sheets) je jazyk pro popis způsobu zobrazení dokumentů napsaných některým ze značkovacích jazyků. Nejčastěji je využíván pro stylování webových stránek napsaných v HTML nebo XHTML. Jazyk byl navržen a je udržován standardizační organizací W3C (World Wide Web Consortium).

Hlavním cílem CSS je oddělit popis vzhledu dokumentu od popisu jeho struktury a obsahu. Pod popisem vzhledu si můžeme představit například výběr barvy písma, velikosti písma, typ fontů, ale i layout stránky. Starší verze jazyka HTML obsahují velké množství elementů, které nepopisují pouze strukturu a obsah dokumentu, ale i vzhled dokumentu. V XHTML 1.1 jsou všechny tyto vlastnosti pro popis vzhledu odstraněny a jediný možný způsob pro popis vzhledu je prostřednictvím CSS.

Hlavní výhody CSS:

- **Rozsáhlejší možnosti pro popis vzhledu** - CSS nabízí rozsáhlejší možnosti popisu vzhledu dokumentu než samotné HTML.
- **Konzistentní vzhled** - K dodržení konzistentnosti vzhledu (například nadpisů, seznamů atd.) bylo v HTML nutné vzhled objektu definovat při každém jeho výskytu, což bylo pracné a velmi špatně udržovatelné. Při jakékoliv změně vzhledu bylo nutné najít všechny výskyty objektu a jeho vzhled upravit. Za použití CSS se vše výrazně zjednodušilo. Veškeré definice vzhledu se uloží do zvláštního souboru, který se připojuje k HTML. Předpisy v tomto souboru pak platí pro celý HTML soubor, takže například pro změnu barvy všech nadpisů stačí editovat pouze definici na jednom místě.
- **Oddělení popisu vzhledu od popisu struktury a obsahu**
- **Snadná změna vzhledu** - Pro změnu vzhledu webových stránek stačí editovat jeden soubor. Není nutné vyhledávat jednotlivé elementy v HTML dokumentu.
- **Kratší doba načítání stránky**

Hlavní nevýhodou CSS je stále špatná podpora webových prohlížečů, které se nedrží specifikace CSS. To způsobuje, že různé prohlížeče interpretují CSS různě, což znamená někdy i značné odlišnosti v konečném vzhledu webové stránky. Pro tvůrce webové stránky je obtížné dosáhnout stejného vzhledu v různých prohlížečích. Není prakticky možné dosáhnout podpory pro všechny prohlížeče.

2.2.8 JavaScript

JavaScript je objektově orientovaný, interpretovaný skriptovací jazyk. Jeho klientská verze (pracující s DOM - Document Object Model) je součástí většiny celosvětově rozšířených webových prohlížečů. V dnešní době je používán hlavně pro webové stránky, kde jím jsou obvykle ovládány různé interaktivní prvky. Je vkládán přímo do HTML kódu, nebo je ve zvláštním souboru, který je k HTML připojen. Program v JavaScriptu je obvykle vykonán prohlížečem až po stažení webové stránky z internetu. JavaScript je ovšem možné vykonávat i na straně serveru. První takovou implementací JavaScriptu byl LiveWire firmy Netscape z roku 1996. Syntaxe jazyka je podobná jazykům C, C++, Java. Slovo Java v názvu nemá nic společného s programovacím jazykem Java, jedná se spíše o marketingový tah. Verze používaná v Internet Exploreru je označována jako JScript.

JavaScript byl původně vytvořen Brendanem Eichem z Netscape pod jménem Mocha. Později přejmenován na LiveScript a dnešní JavaScript. Poprvé byl představen spolu s Netscape browser version 2.0B3 v roce 1995. V červenci 1997 byl JavaScript standardizován asociací ECMA. Tato standardizovaná verze byla pojmenována ECMAScript a jsou z ní odvozeny další implementace jako například ActionScript. Pro objektový model dokumentů DOM, který je součástí klientského JavaScriptu, publikovala organizace W3C první specifikaci v roce 1998. Poslední specifikace DOM3 je z roku 2004. Webové prohlížeče bohužel tuto specifikaci nedodržují. To způsobuje, že v různých prohlížečích se stejný program v JavaScriptu chová jinak. Toto se musí ošetřit větvením programu pro různé prohlížeče.

V dnešní době je JavaScript využíván jako součást Ajaxu. Ajax (Asynchronous JavaScript and XML) je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání [7]. Termín Ajax se poprvé objevil v článku Ajax: A New Approach to Web Applications od Jesse Jamese Garretta. Velká popularita a rozšíření Ajaxu nastalo spolu se spuštěním několika služeb využívajících Ajax společností Google.

Hlavní výhodou Ajaxu je možnost výměny pouze požadovaných dat se serverem. Například při akci uživatel jsou na server odeslána pouze data spojená s touto akcí a ze serveru zpět klientem načteny pouze změny související s touto akcí, přičemž zbylý obsah stránky zůstává nezměněn. To snižuje datový provoz mezi klientem a serverem. Příkladem využití Ajaxu může být automatické doplňování hledaného řetězce při vyhledávání. Ajax poskytuje pro uživatele mnohem přátelštější prostředí, ale vyžaduje použití moderních webových prohlížečů.

Kapitola 3

Analýza a návrh

V této části bude popsána analýza a návrh. V sekci prohlášení o cílech je uveden stručný popis představ zadavatele o tom, co by měl systém podpořit. V sekci požadavky na systém jsou tyto představy dále rozvinuty a upřesněny. Poté následuje model případu užití, jeho popis a popis jednotlivých uživatelů v systému a návrh databáze.

3.1 Prohlášení o cílech

Po konzultaci se zadavatelem by měla tato webová aplikace podpořit následující činnosti: ukládání souborů různých typů souvisejících s výukou, stahování a mazání těchto souborů, možnost přidání textových informací k jednotlivým souborům, editaci těchto informací, možnost nahrání aktualizované verze souboru, vyhledávání souborů, komentování jednotlivých souborů, možnost editace svých komentářů, přidávání a editaci svých novinek, jednoduché diskusní fórum, vytváření, editaci a spouštění testů, sledování změn na stránkách prostřednictvím RSS, registrace nových uživatelů, správu uživatelů, správu obsahu (mazání nevhodných příspěvků), autentizovat se jako uživatel systému. Webová aplikace by měla dvě části, část veřejnou, přístupnou pouze přihlášeným uživatelům a část neveřejnou přístupnou všem.

3.2 Požadavky na systém

Tato webová aplikace, jak je již podle názvu zřejmé, by měla sloužit pro výuku formálních jazyků a překladačů. Musí být volně přístupná pro nejširší veřejnost prostřednictvím internetu. Musí obsahovat dvě části, část veřejnou, kam budou mít přístup i nepřihlášení uživatelé a část neveřejnou, kam budou mít přístup pouze uživatelé po přihlášení. Přihlásit se mohou pouze uživatelé, kteří mají v systému vytvořený uživatelský účet. Tento uživatelský účet si může každý uživatel vytvořit prostřednictvím registrace. V systému musí být o uživateli udržovány následující informace: křestní jméno, příjmení, uživatelské jméno, email (nepovinně) a heslo. Registrace musí obsahovat nějaký prostředek pro zamezení vytváření účtů různými automaty.

Ve veřejné části systému musejí být přístupné veřejné dokumenty, které budou rozčleněny do následujících kategorií: přednášky, opory, záznamy a ostatní. V jednotlivých kategoriích budou vhodným způsobem zobrazeny všechny odpovídající dokumenty a následující informace o nich: jméno dokumentu, popis dokumentu, velikost a datum vložení. Uživatel bude mít možnost vybrání jednoho dokumentu a zobrazení detailních informací o něm.

Uživatel bude mít dále možnost tento dokument stáhnout. O dokumentu musí být v systému udržovány následující informace: jméno dokumentu, kdo dokument vložil, popis dokumentu (nepovinně), klíčová slova dokumentu (nepovinně), kdy byl dokument vložen, mime typ dokumentu, velikost dokumentu, jakého je dokument typu a zda-li je dokument veřejně přístupný, nebo není. Dále musí existovat možnost v těchto dokumentech vyhledávat. Ve veřejné části budou přístupné i veřejné testy. Uživatelé budou mít možnost tyto testy vyhledávat a spouštět. Všechny testy budou zobrazovány v jedné kategorii s následujícími informacemi o nich: jméno testu, popis testu, počet otázek a datum vytvoření testu. Uživatel bude mít dále možnost zobrazit detailní informace o testu, které budou navíc obsahovat informaci o tom, kdo test vytvořil a klíčová slova. U veřejných testů a dokumentů budou zobrazeny komentáře k nim patřící. Pro uživatele musejí být snadno dostupné z jakékoli části informace o nejnovějších komentářích, nejnovějších dokumentech a testech.

Po přihlášení bude mít uživatel přístup i do neveřejné části. Tato část bude obsahovat už všechny dokumenty a všechny testy a komentáře k nim patřící. Zobrazení dokumentu a testů bude stejné jako v neveřejné části. Dokumenty budou opět rozčleněny do kategorií: přednáška, opora, záznam a ostatní. Po zobrazení detailních informací o dokumentu nebo testu budou mít přihlášení uživatelé navíc možnost přidávat k jednotlivým dokumentům nebo testům komentáře. Tyto komentáře musí obsahovat následující informace: nadpis, obsah, datum vložení a kým byl komentář vložen. Na komentáře musí být možnost odpovídat. Výsledné uspořádání komentářů by mělo být takové, aby byla snadno rozpoznatelná struktura (jaká odpověď kam patří). Uživatel musí mít možnost jím vložené komentáře dále editovat.

Pro přihlášené uživatele musí být k dispozici jednoduché diskusní fórum. V této diskusi budou moci uživatelé přidávat nové příspěvky a odpovídat na příspěvky ostatních. Ze struktury příspěvků musí být zřejmé k jakému příspěvku odpověď patří. Příspěvky musí být vhodným způsobem děleny na stránky.

Uživatelé, kteří mají v systému vytvořen uživatelský účet, musejí mít možnost sledovat prostřednictvím kanálu RSS aktuality a informace o nových dokumentech.

V systému musí být možnost pro vytvoření uživatele, který bude mít práva pro přidávání aktualit, dokumentů a vytváření testů. Aktuality musejí obsahovat následující informace: nadpis, obsah, datum vložení a kým byla aktualita vložena. Ten, kdo aktualitu vložil, musí mít možnost pro její editaci. Tyto aktuality budou dostupné přihlášeným uživatelům přímo na stránkách a deset nejnovějších prostřednictvím RSS. Při uploadování dokumentu na server musejí být spolu s ním v systému uloženy i požadované informace o něm uvedené výše. Tyto informace musejí být vlastníkem souboru editovatelné. Vlastník souboru musí mít možnost nahrazení souboru aktualizovanou verzí a možnost odstranění souboru ze systému.

Při vytváření testů musejí být zadány požadované informace o něm popsané výše. Test může obsahovat libovolné množství otázek. Otázka musí obsahovat 2 až 8 odpovědí. Počet správných odpovědí může být 0 až N. Informace o testu i jeho znění musejí být autorem testu editovatelné. Editace testu bude spočívat v odebírání otázek, přidávání otázek, změně otázek, změně odpovědí, odebírání odpovědí, přidávání odpovědí a změně správnosti odpovědí. Po spuštění a odeslání testu musí být test vyhodnocen.

V systému musí být administrátor, který se bude starat o uživatelské účty a obsah aplikace. Správa uživatelských účtů bude spočívat v následujících možnostech: změna přístupových práv uživatele a zamezení nebo povolení přístupu uživatele do systému. Administrátor musí mít možnost odebrání nevhodných komentářů, příspěvků v diskusi, dokumentů a testů.

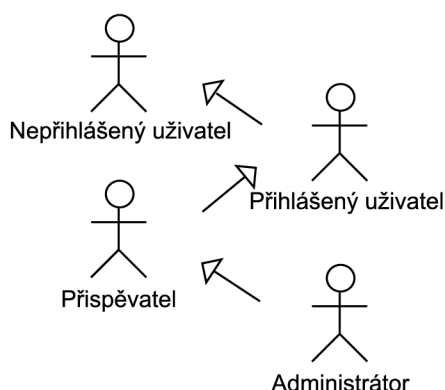
Systém jako celek by měl být uživatelsky přátelský, snadno a intuitivně ovladatelný.

3.3 Model případu užití

Hlavním důvodem tvorby modelu je vymezení hranic systému, nalezení účastníků a zobrazení přípustných akcí z pohledu jednotlivých účastníků. Je to první ucelená představa o tom, co by měl systém z pohledu jednotlivých účastníků zajišťovat. Model by měl být srozumitelný, aby bylo na první pohled zřejmé, k jakým akcím jsou jednotliví účastníci oprávněni. Z důvodu lepší čitelnosti je tento model rozdělen na části podle typu uživatele. K jednotlivým modelům je připojen i popis. V systému budou vystupovat čtyři typy uživatelů. Na nejnižší úrovni to bude nepřihlášený uživatel, který se po vytvoření uživatelského účtu a přihlášení stane přihlášeným uživatelem. Další úroveň je přispěvateľ, přispěvatelem se může stát libovolný uživatel s vytvořeným uživatelským účtem, kterému budou administrátorem změněna práva na přispěvatele. Poslední, nejvyšší úroveň je administrátor. Podrobný popis jednotlivých úrovní bude připojen k odpovídajícím modelům. Vztahy mezi uživateli jsou znázorněny na obrázku 3.1.

3.3.1 Vztahy mezi uživateli

Na obrázku 3.1 jsou znázorněny vztahy mezi jednotlivými uživateli. Pro zjednodušení modelu je zde využito specializace. Specializace spočívá v tom, že pokud má jeden uživatel přístup ke stejným akcím jako druhý, a navíc má přístup k nějakým dalším, označí se tento nadřazený uživatel jako následník druhého uživatele a v modelu se vyznačí pouze vztahy s akcemi pro něj specifickými. Předkem administrátora je přispěvateľ, jehož předkem je přihlášený uživatel a předkem přihlášeného uživatele je nepřihlášený uživatel. Každý následník může provádět akce svého předka.



Obrázek 3.1: Vztahy mezi uživateli

3.3.2 Typy uživatelů

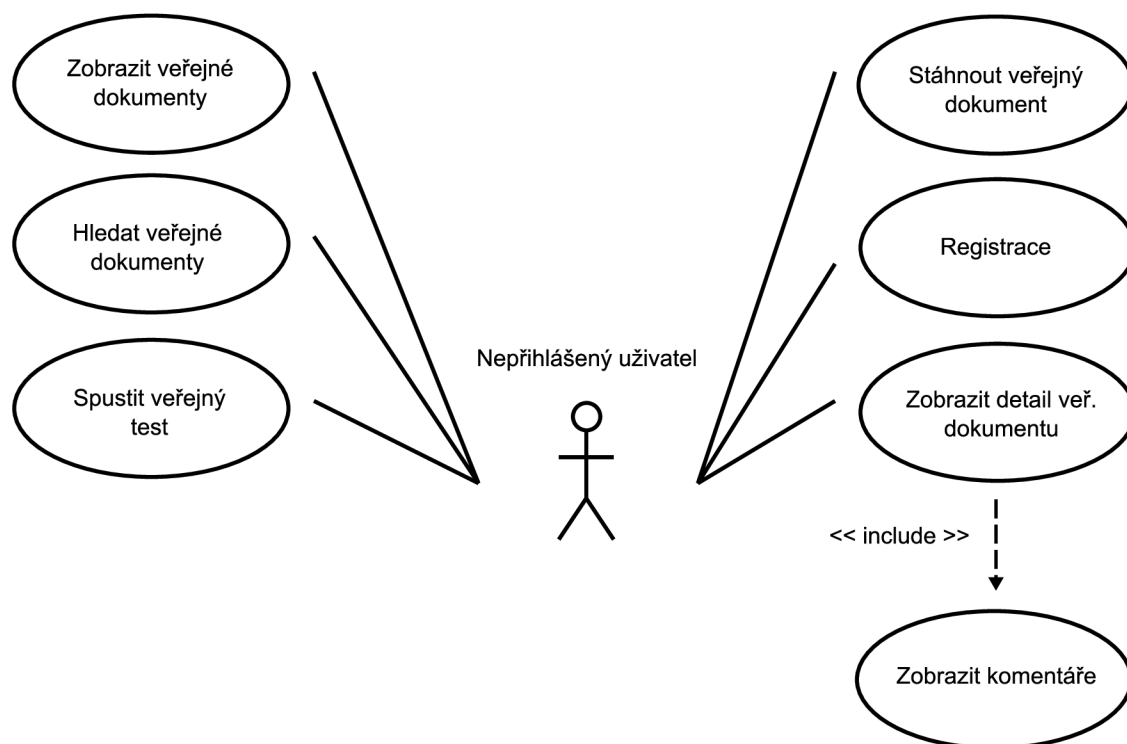
Následuje popis práv jednotlivých typů uživatelů.

Nepřihlášený uživatel

Nepřihlášeným uživatelem je každý uživatel před přihlášením. Uživatelé, kteří nejsou přihlášení mohou prohlížet pouze dokumenty a testy označené jako veřejné. Dokumenty jsou

rozděleny do následujících kategorií: přednášky, opory, záznamy a ostatní. Vyhledávání nalezne pouze ty dokumenty nebo testy, které jsou označeny jako veřejné. Při prohlížení detailů dokumentu nebo testu jsou zobrazeny i komentáře k ním patřící. Nepřihlášený uživatel ale nemá právo komentáře přidávat. Dokumenty označené jako veřejné mohou nepřihlášený uživatelé stahovat a testy označené jako veřejné mohou spouštět. Pokud uživatel nemá vytvořený v systému uživatelský účet, může ho vytvořit prostřednictvím registrace. Po přihlášení se z nepřihlášeného uživatele stává uživatel přihlášený. Přihlášení je možné pouze pro ty uživatele, kteří mají v systému svůj uživatelský účet. Model z pohledu nepřihlášeného uživatele je na obrázku 3.2.

Nepřihlášený uživatel



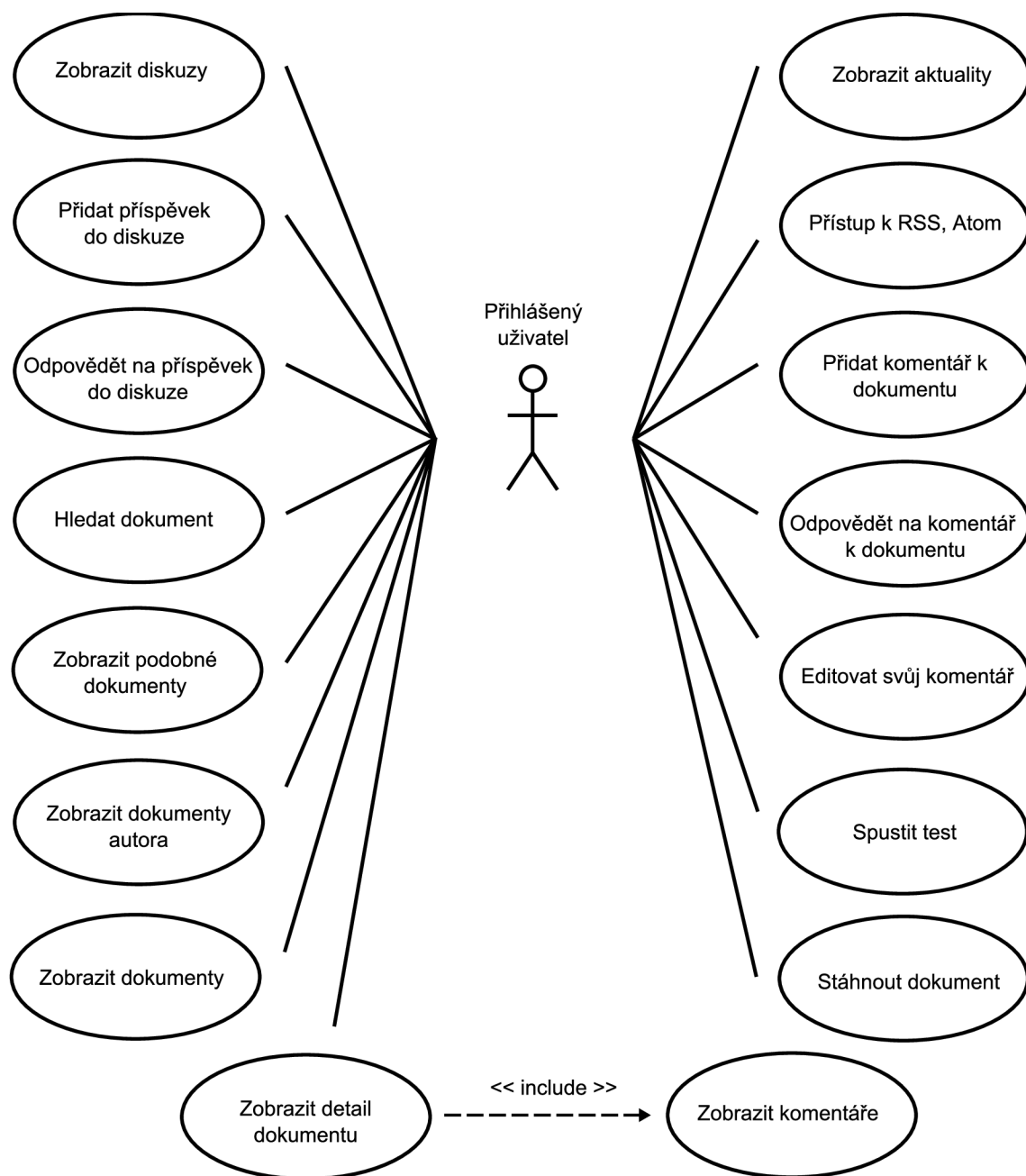
Obrázek 3.2: Nepřihlášený uživatel

Přihlášený uživatel

Je to uživatel, který má v systému vytvořen uživatelský účet a je zalogován. Přihlášený uživatel může provádět vše co nepřihlášený uživatel. Má přístup už ke všem dokumentům, veřejným i neveřejným. Dokumenty může zobrazovat podle následujících kategorií: přednáška, opora, záznam a ostatní. Vybraný dokument může zobrazit, stáhnout, může přidávat komentáře k prohlíženému dokumentu, odpovídat na komentáře ostatních a komentáře jím vložené může editovat. Dále může zobrazit podobné dokumenty k právě prohlíženému dokumentu a zobrazit všechny dokumenty od autora právě prohlíženého dokumentu. V dokumentech může vyhledávat. Má také přístup ke všem testům, veřejným i neveřejným. Tyto testy může spouštět. Stejně jako u dokumentů může k testům přidávat komentáře, odpovídat

na komentáře ostatních a editovat komentáře jím vložené. Je mu umožněn přístup do diskuze, kde může přidávat nové příspěvky a odpovídat na příspěvky už vložené. Dále má přístup do sekce, kde jsou zobrazovány aktuality. Tyto aktuality může sledovat prostřednictvím kanálu RSS nebo Atom. Prostřednictvím RSS a Atom může také sledovat i informace o nejnovějších dokumentech. Model z pohledu přihlášeného uživatele je na obrázku 3.3.

Přihlášený uživatel



Obrázek 3.3: Přihlášený uživatel

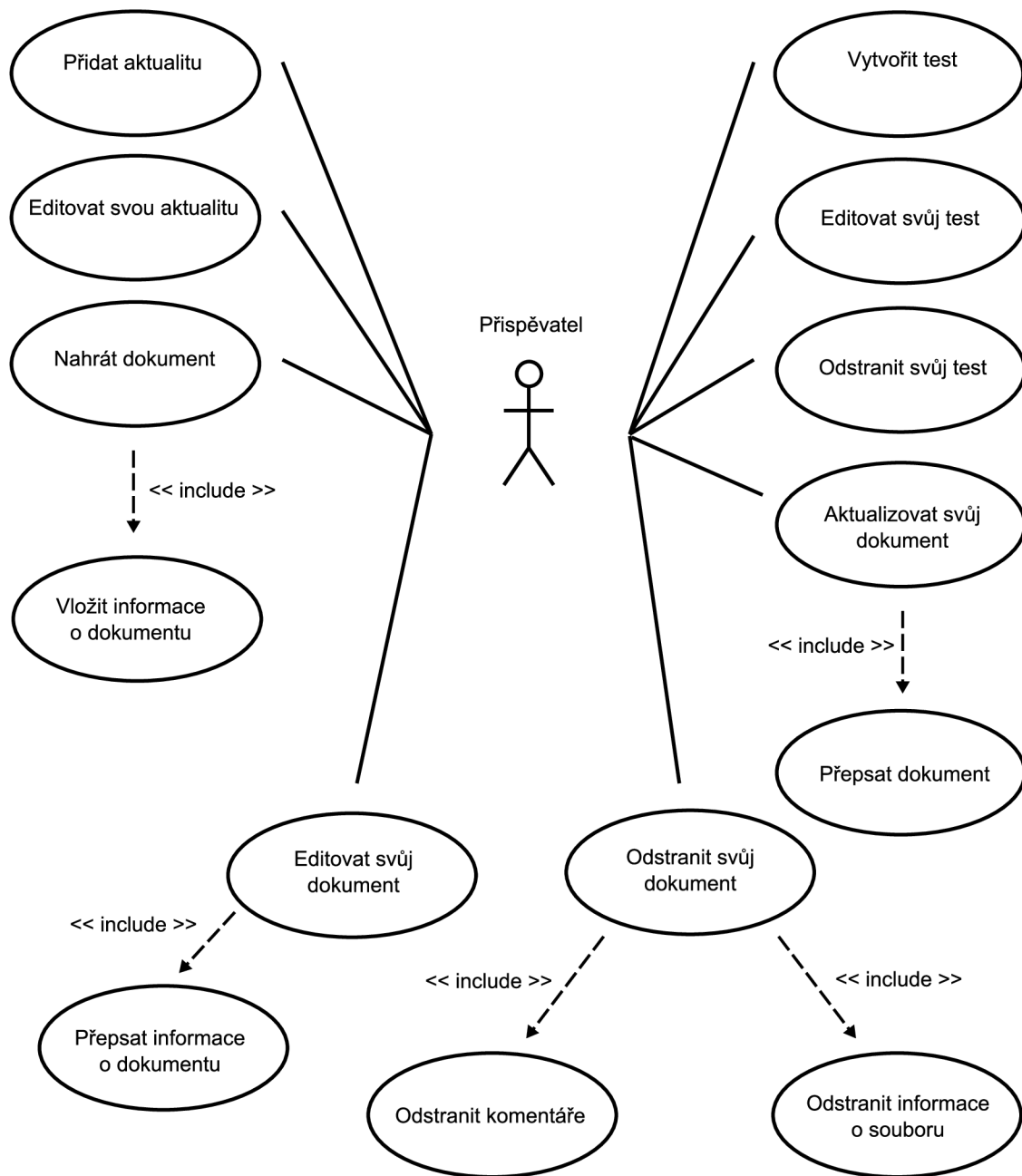
Příspěvatel

Příspěvatelem se může stát po domluvě s administrátorem kdokoli, kdo má v systému vytvořen účet. Příspěvatel může provádět vše co přihlášený uživatel, navíc může přidávat informace do sekce aktualit a editovat aktuality jím vložené. Dále může nahrávat různé dokumenty na server, které pak budou přístupné ostatním uživatelům, k těmto dokumentům může připojit popis, klíčová slova, nastavit práva dokumentu (veřejný, neveřejný), zvolit, zda bude dokument zobrazován pod svým jménem nebo mu přiřadit jiné a vybrat kategorii, do které dokument spadá. Podle zvolené kategorie bude tento dokument zobrazován ostatním uživatelům. Tyto vlastnosti může u jím vložených dokumentů editovat. Své dokumenty může odstranit nebo nahradit aktualizovanou verzí dokumentu. Dále může vytvářet testy, jím vložené testy editovat a mazat. Editace testu spočívá v následujících možnostech: přidat otázku, odebrat otázku, změnit znění otázky, změnit počet odpovědí u vybrané otázky, změnit znění odpovědí u vybrané otázky a změnit správnost těchto odpovědí. Model z pohledu přihlášeného uživatele je na obrázku 3.4.

Administrátor

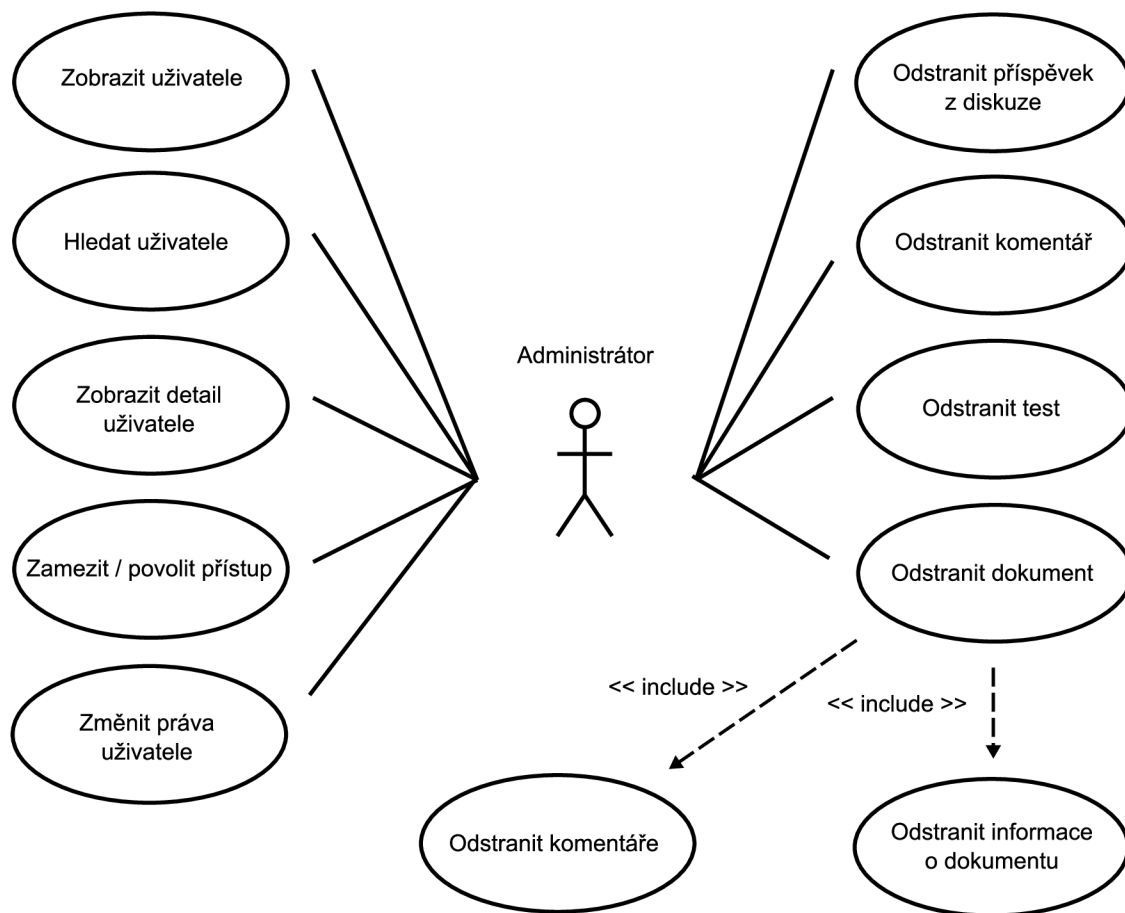
Administrátor spravuje uživatelské účty a obsah aplikace. Může provádět vše co příspěvatel, navíc může zobrazit všechny uživatele, kteří mají v systému uživatelský účet. Tyto uživatele může vyhledávat podle jména, příjmení, uživatelského jména, podle jejich přístupových práv a podle toho, jestli se jedná o uživatele s povoleným nebo zamezeným přístupem. Vyhledávání je možné buď podle jednotlivých klíčů nebo podle jejich kombinace. Administrátor dále může zobrazit detailní informace o uživateli, zamezit nebo povolit mu přístup do systému a změnit uživatelova přístupová práva. Dále může odstranit libovolný nevhodný dokument, test, komentář, nebo příspěvek v diskuzi. Model z pohledu přihlášeného uživatele je na obrázku 3.5.

Prispěvatel



Obrázek 3.4: Prispěvatel

Administrátor



Obrázek 3.5: Administrátor

3.4 ER Diagram

Výsledkem nejčastěji používané techniky při konceptuálním modelování je ER diagram. Konceptuální modelování je fáze návrhu databáze, ve které se analyzují data případně objekty, které jsou v systému potřebné. Tento diagram obsahuje entitní množiny a vztahy mezi nimi. Entita je věc nebo objekt reálného světa, o které chceme mít informace uložené v databázi. Entitní množina je množina entit, které mají stejné vlastnosti. Vlastnosti entity se nazývají atributy. Příkladem entitní množiny je uživatel nebo dokument. ER diagram popisuje data v klidu, neukazuje jaké operace s daty budou probíhat. Obecně lze říci, že ER diagram zobrazuje objekty reálného světa se kterými se bude v systému pracovat a vlastnosti, které nás na těchto objektech zajímají.

Na obrázku 3.6 je zobrazen ER diagram vytvářeného systému. Z něj je zřejmé, s jakými objekty a jeho vlastnostmi se bude v systému pracovat.

O uživateli budou v databázi udržovány následující informace: pořadové číslo uživatele (`idUzivatele`, pro jedinečnou identifikaci uživatele), uživatelské jméno, křestní jméno, pří-

jmení, email uživatele, přihlašovací heslo uživatele, přístupová práva uživatele (pro rozlišení typu uživatele) a datum jeho registrace.

O dokumentu nahraném na server budou v databázi udržovány následující informace: pořadové číslo dokumentu (`idDokumentu`, pro jedinečnou identifikaci dokumentu), jeho jméno, jeho pojmenování, jeho popis, klíčová slova, jakého je typu, jeho mime typ, jeho práva (veřejný, neveřejný), velikost a datum kdy byl vložen. Dokument je spjat s uživatelem, který ho vložil. Jeden uživatel může vložit 0 až N dokumentů a jednotlivé dokumenty mohou mít pouze jednoho vlastníka.

O novinkách budou v databázi udržovány následující informace: pořadové číslo novinky (`idNovinky`, pro jedinečnou identifikaci novinky), datum vložení novinky, nadpis novinky a obsah novinky. Novinka je spjata s uživatelem, který jí vložil. Uživatel může vložit 0 až N novinek a jednotlivé novinky mohou mít pouze jednoho vlastníka.

O příspěvku do diskuze budou v databázi udržovány následující informace: pořadové číslo příspěvku (`idPrispevku`, pro jedinečnou identifikaci příspěvku), datum vložení, nadpis a obsah příspěvku. Příspěvek je spjat s uživatelem, který ho vložil a s rodičovským příspěvkem. Uživatel může vložit 0 až N příspěvků a jeden příspěvek může mít pouze jednoho vlastníka. Příspěvek může být spjat s jedním nebo žádným rodičovským příspěvkem. Rodičovský příspěvek může mít 0 až N příspěvků (odpovědí).

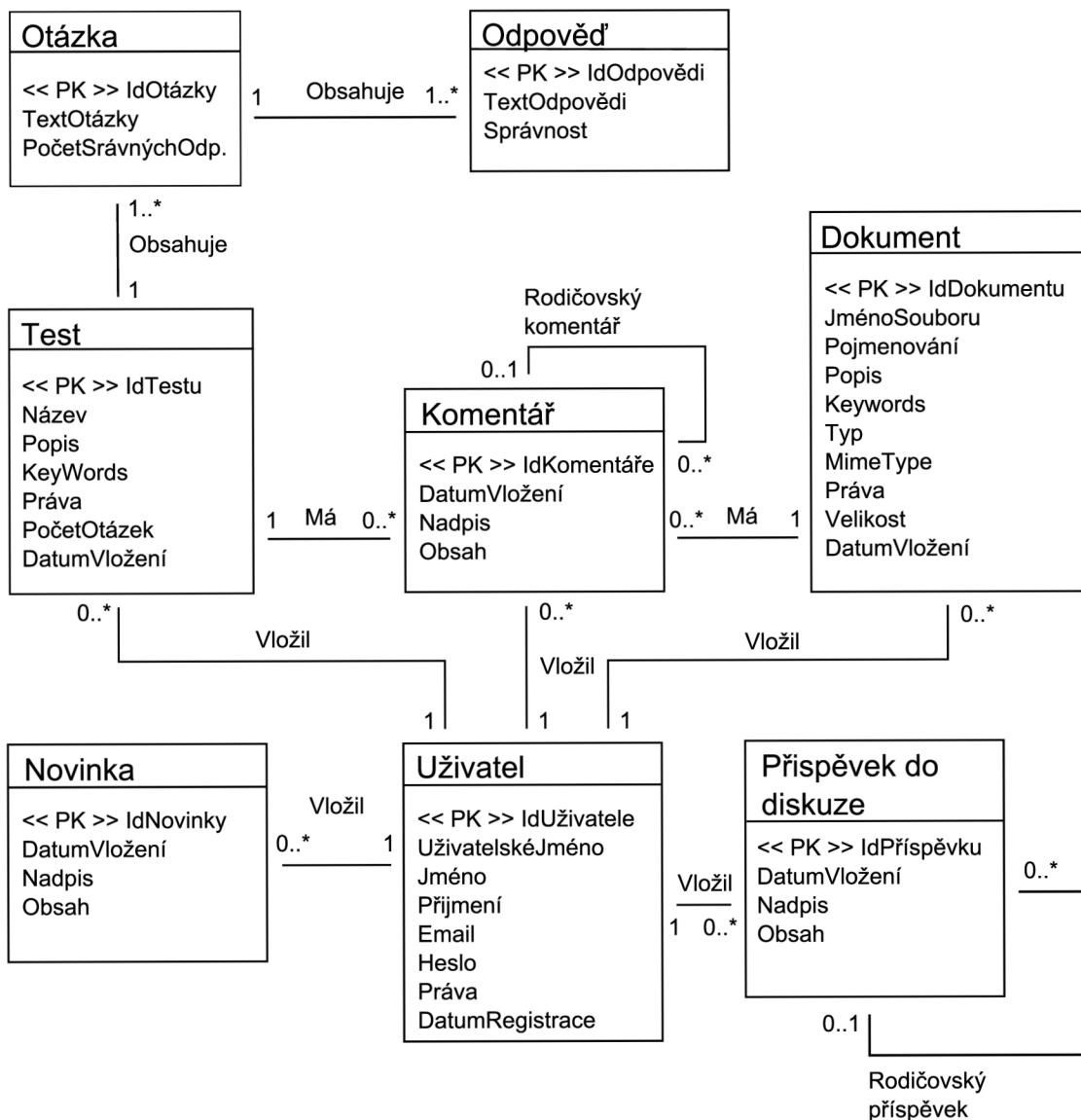
O komentáři budou v databázi udržovány následující informace: pořadové číslo komentáře (`idKomentare`, pro jedinečnou identifikaci komentáře), datum vložení, nadpis a obsah komentáře. Komentář je spjat s uživatelem, který ho vložil, se svým rodičovským komentářem a s dokumentem nebo testem, ke kterému patří. Komentář může mít pouze jednoho vlastníka a být přiřazen pouze k jednomu testu nebo dokumentu. Dále může být spjat s jedním nebo žádným rodičovským komentářem. Rodičovský komentář může obsahovat 0 až N komentářů (odpovědí). Uživatel může vložit 0 až N komentářů. K dokumentu nebo testu může být přiřazeno 0 až N komentářů.

O testech budou v databázi udržovány následující informace: pořadové číslo testu (`idTestu`, pro jedinečnou identifikaci testu), název testu, popis testu, klíčová slova, přístupová práva k testu (veřejný, neveřejný), kolik otázek test obsahuje a datum jeho vytvoření. Test je spjat s uživatelem který ho vložil. Uživatel může vytvořit 0 až N testů a test smí mít pouze jednoho vlastníka.

O otázkách budou v databázi udržovány následující informace: pořadové číslo otázky (`idOtazky`, pro jedinečnou identifikaci otázky), znění otázky a kolik obsahuje správných odpovědí. Otázka je spjata s testem. Test může mít 1 až N otázek. Otázka může patřit pouze k jednomu testu.

O odpovědích budou v databázi udržovány následující informace: pořadové číslo odpovědi (`idOdpovedi`, pro jedinečnou identifikaci odpovědi), znění odpovědi a informace o tom, jestli je správná. Odpověď je spjata s otázkou. Otázka může mít 1 až N odpovědí. Odpověď může patřit pouze k jedné otázce.

Z ER diagramu se navrhnu databázové tabulky. Při této transformaci se informace udržované v databázi mohou rozšířit. Rozšíření je nutné například pro zachycení informací o vzájemných vztazích mezi tabulkami. Tato transformace ER diagramu na databázové tabulky a popis z jakých důvodů jsou v tabulkách právě tyto informace, je uveden v následující kapitole.



Obrázek 3.6: E-R Diagram

3.5 Návrh databáze

Převod ER diagramu na tabulky relační databáze je dalším krokem při návrhu systému. Výsledkem této transformace budou tabulky, které budou obsahovat už všechny informace, s nimiž budeme v databázi pracovat. Po přiřazení datových typů k jednotlivým sloupcům bude už možné jednotlivé tabulky v databázi vytvořit. Prvním krokem transformace je převod složených a vícehodnotových atributů (převod do první normální formy). Například atribut adresa se rozepíše na ulici, město a PSČ. Dalším krokem je převod do druhé normální formy. Pravidla druhé normální formy vyžadují, aby databáze odpovídala první normální formě a aby všechny sloupce, jejichž hodnoty jsou ve více řádcích stejné, byly převedeny do

vlastní tabulky [5]. Aby databáze odpovídala třetí normální formě, musí odpovídat první a druhé normální formě a všechny sloupce, které nejsou součástí primárního klíče, jsou na primárním klíči přímo závislé. Databáze odpovídající třetí normální formě jsou stabilní a odolné. Ne všechny databáze je možné do třetí normální formy převést. Normalizované databáze jsou oproti nenormalizovaným pomalejší, proto někteří vývojáři normalizaci neprovádějí. Tato databáze odpovídá první normální formě. Dalším krokem je převod vztahů mezi tabulkami. Všechny vztahy mezi tabulkami musejí být vhodným způsobem reprezentovány v databázi.

Jako databáze bude použit databázový systém MySQL, proto se tento návrh musí řídit datovými typy, které MySQL nabízí. Následující tabulky obsahují popis jednotlivých atributů entitních množin, datové typy těchto atributů a jejich parametry. K tabulkám je připojen i popis, z jakých důvodů v systému jsou a z jakých důvodů v nich jsou obsaženy právě tyto atributy. Zdrojový kód pro vytvoření databázových tabulek je uveden v příloze A. Z důvodu použití cizích klíčů je všem tabulkám explicitně přiřazen typ InnoDB. Integritní omezení a reprezentace vztahů mezi tabulkami jsou popsány u jednotlivých tabulek.

3.5.1 Popis jednotlivých tabulek

Tabulka *uzivatele*

V této tabulce jsou udržovány informace o uživateli, kteří mají v systému uživatelský účet. Jednotlivé atributy jsou popsány přímo v tabulce. Sloupec *aktivni*, byl přidán jako reakce na požadovanou možnost administrátora pro zablokování a odblokování přístupu uživatele do systému. Sloupec *idUzivatele* bude sloužit jako primární klíč. Toto číslo bude automaticky inkrementováno při každém přidání uživatele. Všechny sloupce, kromě sloupce *email*, musejí být vyplněny. Hodnoty ve sloupci *uzivatelskeJmeno* musí být unikátní.

Sloupec	Typ	Parametry	Popis
<i>idUzivatele</i>	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo uživatele
<i>jmeno</i>	VARCHAR	NOT NULL	Křestní jméno uživatele
<i>prijmeni</i>	VARCHAR	NOT NULL	Příjmení uživatele
<i>uzivatelskeJmeno</i>	VARCHAR	NOT NULL, UNIQUE	Unikátní uživatelské jméno
<i>email</i>	VARCHAR		Email uživatele (nepovinné)
<i>heslo</i>	VARCHAR	NOT NULL	Heslo uživatele (uloženo jako výstup funkce PASSWORD)
<i>prava</i>	VARCHAR	NOT NULL	Uživatelská práva pro přístup do systému
<i>datumRegistrace</i>	DATETIME	NOT NULL	Datum registrace uživatele
<i>aktivni</i>	VARCHAR	NOT NULL	Pro případné zablokování přístupu uživatele

Tabulka *onlineUzivatele*

Tato pomocná tabulka je v systému pouze pro ošetření timeoutu a udržování online uživatelů. Sloupec `cas` obsahuje čas poslední uživatelovy akce. Pokud bude rozdíl času poslední uživatelovy akce a aktuálního času větší než zadaný počet sekund, uživatel bude odhlášen. Primárním klíčem v této tabulce je `idUzivatele`. Všechny sloupce v tabulce musejí být vyplněny.

Sloupec	Typ	Parametry	Popis
<code>idUzivatele</code>	MEDIUMINT	UNSIGNED, NOT NULL, PRIMARY KEY	Unikátní číslo uživatele
<code>uzivatelskeJmeno</code>	VARCHAR	NOT NULL	Unikátní uživatelské jméno
<code>cas</code>	BIGINT	NOT NULL	Čas poslední uživatelovy akce (pro ošetření timeoutu).
<code>prava</code>	VARCHAR	NOT NULL	Uživatelská práva pro přístup do systému

Tabulka *novinky*

V této tabulce jsou uloženy novinky přidávané přispěvatelem. Primárním klíčem je sloupec `idNovinky`. Číslo v tomto sloupci se bude automaticky inkrementovat při každém vložení novinky. Sloupec `uzivatelskeJmeno` obsahuje uživatelské jméno přispěvatele, který novinku vložil. Takto je reprezentován vztah mezi tabulkou `uzivatele` a tabulkou `novinky`. Uživatelská jména jsou unikátní, takže je lze využít k jednoznačné identifikaci uživatele. Uživatelské jméno je použito namísto `id` uživatele z důvodu ušetření přístupu do databáze. Kdyby bylo použito `id` uživatele, musel by se pro zjištění uživatelského jména provést dotaz do databáze, který by podle `id` uživatele určil jeho uživatelské jméno. Všechny sloupce musejí být vyplněny.

Sloupec	Typ	Parametry	Popis
<code>idNovinky</code>	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo novinky
<code>uzivatelskeJmeno</code>	VARCHAR	NOT NULL	Jméno uživatele, který novinku vložil
<code>nadpis</code>	TINYTEXT	NOT NULL	Nadpis novinky
<code>obsah</code>	TEXT	NOT NULL	Obsah novinky
<code>datumRegistrace</code>	DATETIME	NOT NULL	Datum vložení novinky

Tabulka *dokumenty*

V této tabulce jsou udržovány informace o dokumentech uložených na serveru. Primárním klíčem je sloupec `idDokumentu`. Číslo v tomto sloupci je automaticky inkrementováno při

každém vložení dokumentu. Sloupec `jmenoSouboru` obsahuje celé jméno, pod jakým byl dokument na server nahrán. Toto jméno je využíváno pro fyzické vyhledání souboru ve složce, například při požadavku na jeho stažení. Sloupec `pojmenovani` je zde pro alternativní pojmenování dokumentu, které bude zobrazováno uživatelům. Sloupec `mimeType` je využíván pro rozčlenění dokumentů podle jeho typu mime. Sloupec `keywords` je zde pro usnadnění vyhledávání v dokumentech. Ve sloupci `prava`, je udržována informace o tom, jestli může být dokument zobrazen i nepřihlášeným uživatelům nebo pouze přihlášeným. Sloupce `idDokumentu`, `jmenoSouboru`, `vlozil`, `typ`, `mimeType`, `prava`, `velikost` a `datumVlozeni` musejí být vyplněny. Ostatní sloupce vyplněny být nemusejí. Pro sloupec `vlozil` platí to stejné, co pro sloupec `uzivatelskeJmeno` v tabulce `novinky`. Tento sloupec reprezentuje vztah mezi tabulkou `uzivatele` a tabulkou `dokumenty`.

Sloupec	Typ	Parametry	Popis
<code>idDokumentu</code>	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo dokumentu
<code>jmenoSouboru</code>	TINYTEXT	NOT NULL	Jméno souboru
<code>pojmenovani</code>	TINYTEXT		Pojmenování souboru (nepovinné), uživatel může zvolit pojmenování namísto reálného názvu souboru
<code>popis</code>	TEXT		Popis dokumentu (nepovinné)
<code>vlozil</code>	VARCHAR	NOT NULL	Jméno uživatele, který dokumentu vložil
<code>typ</code>	VARCHAR	NOT NULL	Pro zařazení dokumentu podle typu
<code>mimeType</code>	VARCHAR	NOT NULL	Mime typ dokumentu
<code>keywords</code>	TINYTEXT		Klíčová slova (pro vyhledávání)
<code>prava</code>	VARCHAR	NOT NULL	Nastavení práv dokumentu (veřejný, neveřejný)
<code>velikost</code>	VARCHAR	NOT NULL	Velikost dokumentu
<code>datumVlozeni</code>	DATETIME	NOT NULL	Datum vložení dokumentu

Tabulka *testy*

Tato tabulka slouží pro udržování informací o testech. Otázky a odpovědi k testům patříčí jsou uloženy ve zvláštních tabulkách, což umožňuje libovolný počet otázek. Primárním klíčem je sloupec `idTestu`. Číslo v tomto sloupci je automaticky inkrementováno při každém vložení testu. Sloupec `keywords` je zde pro usnadnění vyhledávání v testech. Pro sloupce `vlozil` a `prava` platí to stejné, co pro ně platí v tabulce `dokumenty`. Sloupec `vlozil` zde reprezentuje vztah mezi tabulkou `uzivatele` a tabulkou `testy`. Všechny sloupce v této tabulce musejí být vyplněny.

Sloupec	Typ	Parametry	Popis
idTestu	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo testu
nazev	TINYTEXT	NOT NULL	Název testu
popis	TEXT	NOT NULL	Popis testu
keywords	TINYTEXT	NOT NULL	Klíčová slova (pro vyhledávání)
vlozil	VARCHAR	NOT NULL	Jméno uživatele, který test vložil
prava	VARCHAR	NOT NULL	Nastavení práv testu (veřejný, neveřejný)
pocetOtazek	SMALLINT	NOT NULL	Počet otázek, který test obsahuje
datumVlozeni	DATETIME	NOT NULL	Datum vložení testu

Tabulka *otazky*

Tato tabulka obsahuje otázky patřící k jednotlivým testům. Primárním klíčem je sloupec `idOtazky`. Číslo v tomto sloupci je automaticky inkrementováno při každém vložení otázky. Tato tabulka je sloupcem `idTestuRod` (cizí klíč) svázána s tabulkou `testy`. Tento sloupec reprezentuje vztah mezi tabulkou `testy` a tabulkou `otazky`. Obsahuje id testu, ke kterému tato otázka patří. Tímto způsobem je možné k testu vytvořit libovolné množství otázek. Referenční akce při smazání rodičovského testu je `CASCADE`. To znamená, že se smažou i otázky k testu patřící. Tímto je zajištěna integrita databáze. Všechny sloupce, kromě sloupce `pocetSpravnych`, musejí být vyplněny. Sloupec `pocetSpravnych` je pomocný sloupec využíváný při generování testů.

Sloupec	Typ	Parametry	Popis
idOtazky	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo otázky
idTestuRod	MEDIUMINT	UNSIGNED, NOT NULL, FOREIGN KEY (idTestuRod) REFERENCES testy(idTestu) ON DELETE CASCADE	Id testu, ke kterému otázka patří
textOtazky	TEXT	NOT NULL	Text otázky
pocetSpravnych	SMALLINT		Počet správných odpovědí

Tabulka *odpovedi*

Tato tabulka slouží pro udržování jednotlivých odpovědí a informací o jejich správnosti. Primárním klíčem je sloupec `idOdpovedi`. Číslo v tomto sloupci je automaticky inkrementováno při každém vložení odpovědi. Tabulka je sloupcem `idOtazkyRod` (cizí klíč) svázána s tabulkou `otazky`. Tento sloupec reprezentuje vztah mezi tabulkou `otazky` a tabulkou `odpovedi`. Obsahuje id otázky, ke které odpověď patří. Při smazání rodičovské otázky jsou smazány všechny odpovědi k otázce patřící. To je ošetřeno stejně jakou u otázek referenční akcí `CASCADE`. Tímto je opět zajištěna integrita databáze. Všechny sloupce v této tabulce musejí být vyplněny.

Sloupec	Typ	Parametry	Popis
<code>idOdpovedi</code>	<code>MEDIUMINT</code>	<code>UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY</code>	Unikátní číslo odpovědi
<code>idOtazkyRod</code>	<code>MEDIUMINT</code>	<code>UNSIGNED, NOT NULL, FOREIGN KEY (idOtazkyRod) REFERENCES otazky(idOtazky) ON DELETE CASCADE</code>	Id otázky, ke které odpověď patří
<code>textOdpovedi</code>	<code>TEXT</code>	<code>NOT NULL</code>	Text odpovědi
<code>anoNe</code>	<code>VARCHAR</code>	<code>NOT NULL</code>	Pro určení správnosti odpovědi

Tabulka *komentare*

Tabulka `komentare` obsahuje informace o komentáři a samotné znění komentáře. Primárním klíčem je sloupec `idKomentare`. Číslo v tomto sloupci je automaticky inkrementováno při každém vložení komentáře. Sloupec `idNadKomentare` reprezentuje vztah mezi komentářem a jeho rodičovským komentářem. Tento sloupec určuje, ke kterému rodičovskému komentáři je komentář vázán. To umožňuje zobrazovat komentáře a odpovědi k nim patřící ve strukturované formě což zlepšuje přehlednost. Pokud sloupec `idNadKomentare` obsahuje nějaké číslo, pak je tento komentář odpovědí. Pokud obsahuje symbol návrhárem zvolený jako označení toho, že komentář nemá žádný rodičovský komentář, pak je tento komentář komentářem kořenovým. Sloupec `idDokumentu` reprezentuje vztah mezi tabulkou `dokumenty` a tabulkou `komentare`, obsahuje id dokumentu, ke kterému komentář patří (pokud nepatří k testu). Sloupec `idTestu` reprezentuje vztah mezi tabulkou `testy` a tabulkou `komentare`, obsahuje id testu, ke kterému komentář patří (pokud nepatří k dokumentu). Pro sloupec `uzivatelskeJmeno` platí to stejné, co pro sloupec `uzivatelskeJmeno` v tabulce `novinky`. Tento sloupec reprezentuje vztah mezi tabulkou `uzivatele` a tabulkou `komentare`. Všechny sloupce v této tabulce musejí být vyplněny. Při odebrání dokumentu nebo testu musí být zajištěno i odebrání všech komentářů k němu patřících. To je zajištěno pomocí PHP.

Sloupec	Typ	Parametry	Popis
idKomentare	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo komentáře
idNadKomentare	MEDIUMINT	UNSIGNED, NOT NULL	Číslo rodičovského komentáře
idDokumentnu	MEDIUMINT	UNSIGNED, NOT NULL	Číslo dokumentu, ke kterému komentáře patří
idTestu	MEDIUMINT	UNSIGNED, NOT NULL	Číslo testu, ke kterému komentáře patří
uzivatelskeJmeno	VARCHAR	NOT NULL	Jméno uživatele, který komentář vložil
nadpis	TINYTEXT	NOT NULL	Nadpis komentáře
obsah	TEXT	NOT NULL	Obsah komentáře
datumRegistrace	DATETIME	NOT NULL	Datum vložení komentáře

Tabulka *diskuze*

V této tabulce jsou uloženy příspěvky do diskuze a informace o nich. Primárním klíčem je sloupec `idPrispevku`. Číslo v tomto sloupci je automaticky inkrementováno při každém vložení příspěvku. Pro sloupec `uzivatelskeJmeno` platí to stejné, co pro sloupec `uzivatelskeJmeno` v tabulce `novinky`. Tento sloupec reprezentuje vztah mezi tabulkou `uzivatele` a tabulkou `diskuze`. Sloupec `idRodice` reprezentuje vztah mezi příspěvkem a jeho rodičovským příspěvkem. Obsahuje `id` rodičovského příspěvku. Je zde uveden ze stejného důvodu jako sloupec `idNadKomentare` v tabulce `komentare`. Všechny sloupce musejí být vyplněny.

Sloupec	Typ	Parametry	Popis
idPrispevku	MEDIUMINT	UNSIGNED, NOT NULL, AUTOINCREMENT, PRIMARY KEY	Unikátní číslo příspěvku
idRodice	MEDIUMINT	UNSIGNED, NOT NULL	Unikátní číslo rodičovského příspěvku
uzivatelskeJmeno	VARCHAR	NOT NULL	Jméno uživatele, který příspěvek vložil
nadpis	TINYTEXT	NOT NULL	Nadpis příspěvku
obsah	TEXT	NOT NULL	Obsah příspěvku
datumRegistrace	DATETIME	NOT NULL	Datum vložení příspěvku

Kapitola 4

Implementace

Aplikace je implementována v jazyku PHP 5 procedurálním způsobem. Struktura stránek je napsána v jazyku XHTML 1.0 Transitional. Pro definici vzhledu je použit jazyk CSS. Jazyk JavaScript je využit pro práci s DOM objekty. Jako databázový server je využit systém MySql. Pro přístup do databáze je využit SQL dialekt jazyka PHP. V aplikaci je implementováno vše co bylo zadavatelem požadováno. Způsob implementace jednotlivých částí systému je popsán v následující kapitole. Ovládání aplikace je intuitivní, tam kde by se mohl vyskytnout problém, je připojena nápověda. Uživatelská příručka a návod na instalaci jsou uvedeny v příloze.

4.1 Struktura kódu

Hlavním souborem je `index.php`. Tento soubor se provádí téměř při všech akcích uživatele. Jeho struktura je následující. Při je vykonávání jsou nejprve načteny potřebné funkce, poté je načteno nastavení z konfiguračního souboru `nastaveni.inc`, poté je navázáno spojení s databází. Následuje kontrola uživatelských nastavení uložených v cookies, pokud nejsou nalezeny použije se defaultní nastavení ze souboru `nastaveni.inc`. Dále se vloží kód ze souboru `osetri_timeout.php`, který se postará o odpojení uživatele pokud doba jeho nečinnosti přesáhne nastavenou mez, pokud je vše v pořádku, aktualizuje čas poslední uživatelské akce. Následuje načtení hlavičky (obsahuje metadata a hlavičku XHTML dokumentu, jsou zde načteny funkce v JavaScriptu), inkudování kódu z požadovaného souboru (požadavky na soubory jsou předávány metodou GET v globální proměnné `$_GET['page']`) a načtení patky (uzavření XHTML dokumentu). Pokud je požadavek na soubor, ve kterém je nutné, aby před ním nebylo do webového prohlížeče nic odesláno (například soubor, který obsahuje funkce `setcookie` nebo `header`), je tento soubor načten ještě před načtením hlavičky a veškeré jeho generované výstupy jsou ukládány do proměnné. Tato proměnná je poté mezi hlavičkou a patkou vypsána. Logování uživatelů je řešeno v samostatném souboru `login.php`. Soubory, které používají jiný layout (`spust_test.php`) nebo layout nepoužívají vůbec (`rss.php`, `download.php`, `generuj_obrazek.php`), nejsou inkudovány mezi hlavičku a patku v souboru `index.php`, ale jsou volány samostatně.

4.2 Popis jednotlivých částí systému

Následuje popis jednotlivých částí aplikace.

4.2.1 Registrace

Registrací je vytvořen uživatelský účet, který je nezbytný pro přístup do aplikace. Registrace je přístupná každému uživateli. Osobní údaje uživatel zadává do registračního formuláře, kde jsou ošetřeny regulárními výrazy. Křestní jméno může obsahovat 2 až 30 znaků. Povolené znaky jsou velká a malá písmena abecedy i s diakritikou, znak tečky, apostrofu, mezery a pomlčky. Pro příjmení platí stejné omezení. Uživatelské jméno může obsahovat 4 až 20 znaků. Mezi povolené znaky patří velká a malá písmena abecedy i s diakritikou, znak podtržítka a číslice. Email musí být v následujícím tvaru: musí začínat písmenem nebo číslicí a musí pokračovat kombinací číslic, písmen, podtržitek, teček a pomlček. Poté musí následovat symbol @(zavináč) a kombinace písmen, číslic, teček a pomlček. Nakonec musí být uvedena tečka a dvou až čtyřmístný řetězec malých písmen. Heslo může obsahovat 4 až 20 znaků. Povolené znaky jsou velká a malá písmena bez diakritiky, malá písmena s diakritikou a číslice. Heslo musí být zadáno pro kontrolu dvakrát. Jako požadovaná ochrana proti automatické registraci je generován obrázek se šestimístnou kombinací náhodných, různě umístěných číslic. Tato kombinace musí být opsána do poslední položky registračního formuláře. Tento obrázek je generován pomocí PHP. Číslice jsou náhodné, stejně jako jejich barva a font. Fonty jsou vybírány z deseti možných. Všechny údaje kromě emailu jsou povinné.

Po odeslání registračního formuláře jsou zadané informace uloženy do tabulky `uzivatele`. Heslo je uloženo v zakódované podobě.

4.2.2 Přihlášení do systému

Pro přihlášení do systému musí mít uživatel vytvořen uživatelský účet. Přihlašovací údaje (uživatelské jméno a heslo) jsou zadávány do přihlašovacího formuláře. Po odeslání je vyhledána tato kombinace v tabulce `uzivatele`. Pokud je nalezena a uživatel nemá administrátorem zamezen přístup, jsou nastavena cookies `idUzivatele`, `stav` a `cas`, dále je uživatel vložen do tabulky `onlineUzivatele`. Pokud kombinace nalezena není, uživateli je přístup do systému zamítnut. Po úspěšném přihlášení, je podle uživatelových práv zobrazena odpovídající část systému.

4.2.3 Zobrazení a vyhledávání dokumentů

Dokumenty jsou zobrazovány v záložkách podle kategorie zvolené při uploadu přispěvatelem. V jednotlivých kategoriích jsou dokumenty dále členěny podle svého typu. V kategorii přednášky a opory jsou členěny na dokumenty typu Pdf, MS Word, postscript a ostatní. V kategorii záznamy jsou členěny na audio a video. Toto členění je na základě mime typu souboru. Uživatel má možnost nastavení počtu zobrazených dokumentů na stránku. Vyhledávání dokumentů probíhá na základě SQL dotazu do databáze. Dotaz je generován v závislosti na nastavených volbách pro vyhledávání. Testy a informace o dokumentech jsou uloženy v různých databázových tabulkách, takže se generování dotazu při vyhledávání testů a dokumentů liší. Při vyhledávání testů je ignorována zadaná velikost a typ souboru. Uživatel má možnost volby podle čeho vyhledávat (názvu, klíčových slov, popisu), dále může zvolit kategorii, ve které vyhledávat (při volbě test jsou vyhledávány testy, při ostatních volbách soubory), typ souboru (toto se ignoruje při vyhledávání testů), může zadat rozmezí velikosti (toto se ignoruje při vyhledávání testů) a data vložení souboru nebo testu. Nalezené soubory nebo testy má možnost seřadit buď podle data (vzestupně, sestupně), nebo podle velikosti (vzestupně, sestupně, toto se ignoruje při vyhledávání testů).

Vyhledávat se může podle zadaného řetězce, podle velikosti souboru (neplatí pro testy), podle data vložení souboru/testu nebo podle kombinace všech možností. Při vyhledávání nepřihlášeným uživatelem jsou zobrazeny pouze veřejné soubory/testy. Při vyhledávání víceslovního řetězce jsou zobrazeny dokumenty, ve kterých byly nalezeny všechny části řetězce. To znamená, že mezi jednotlivými částmi platí vztah AND. Vyhledávání ignoruje velikosti znaků (Case-insensitive).

4.2.4 Komentářový subsystém

Komentáře jsou vkládány prostřednictvím formuláře. Obsah vloženého textu není nijak omezován, pouze délkou. Po odeslání formuláře je zadaný text ošetřen proti nebezpečným znakům a uložen do tabulky `komentare`. Spolu s textem je do tabulky uloženo i kým byl komentář vložen, čas jeho vložení, id dokumentu nebo testu ke kterému patří a id rodičovského komentáře. Při vykreslování komentářů je využito rekurze pro vytvoření stromové struktury. Komentáře jsou zobrazovány u odpovídajícího testu nebo dokumentu a nejsou stránkovány. Možnost editace je zobrazována pouze u těch komentářů, kde se uživatelské jméno vlastníka komentáře z databáze shoduje s uživatelským jménem právě přihlášeného uživatele. Mazání komentáře administrátorem spočívá pouze v přepsání nadpisu a obsahu komentáře textem, informujícím o odebrání komentáře administrátorem.

4.2.5 Diskuze

Příspěvky do diskuze jsou zadávány stejným způsobem jako komentáře. Po odeslání formuláře jsou spolu s textem do tabulky `diskuze` uloženy informace o tom, kým byl příspěvek vložen, datum vložení a id rodičovského příspěvku. Vykreslení diskuze je opět pro zajištění stromové struktury provedeno rekurzí. Uživatel má možnost nastavit počet zobrazovaných příspěvků na jednu stranu. Editace není implementována a odstranění administrátorem probíhá stejným způsobem jako u komentářů.

4.2.6 Aktuality

Vkládání aktualit probíhá také prostřednictvím formuláře. Po jeho odeslání jsou spolu s textem do tabulky `novinky` uloženy datum vložení a uživatelské jméno právě přihlášeného uživatele. Možnost editace je zobrazována pouze u těch aktualit, u kterých se uživatelské jméno vlastníka aktuality shoduje s uživatelským jménem právě přihlášeného uživatele. Odstranění aktuality není implementováno.

4.2.7 Upload souborů

Upload souborů na server se děje prostřednictvím formuláře metodou POST. Maximální velikost souboru je dána nastavením PHP. V případě této aplikace je to 20MB. Formulář pro upload souboru obsahuje pole pro popis souboru a pro klíčová slova. Popis i klíčová slova jsou nepovinné. Dále má přispěvatel možnost zvolit pod jakým jménem bude soubor zobrazován ostatním uživatelům. Defaultně je název souboru zobrazován ve tvaru v jakém je uploadován (i s koncovkou). Při zvolení jiného pojmenování souboru má přispěvatel možnost zadat jiný název, pod kterým bude soubor zobrazen pro ostatní uživatele. Dále přispěvatel nastaví práva souboru (veřejný - soubor je přístupný všem, neveřejný - soubor je přístupný pouze přihlášeným uživatelům) a kategorii do které soubor spadá. Po odeslání je soubor přejmenován podle přiděleného id (jméno souboru ve složce musí být unikátní), uložen do

složky na serveru a informace o něm uloženy do databáze. Informace o souboru lze editovat, je možné uploadovat i aktuální verzi souboru, kterou se přepíše verze původní. Editace je možná pouze u souborů, kde se uživatelské jméno vlastníka shoduje s uživatelským jménem právě přihlášeného uživatele.

4.2.8 Testy

Testy se tvoří prostřednictvím formulářů. Úvodní formulář slouží pro vyplnění informací o testu jako je název, popis, klíčová slova a přístupová práva k souboru. Všechny tyto položky jsou povinné. Po odeslání je možné přidávat otázky. Otázka může obsahovat dvě až osm odpovědí. Správnost odpovědí se nastaví zatrhávacími tlačítky. Správných odpovědí může být 0 až N. Při jedné správné odpovědi se při generování testu pro odpovědi použijí radiobuttony jinak checkboxy. Test lze v každé fázi tvoření prohlížet a editovat. Editace spočívá v možnosti odebrání otázky, přidání otázky, změně znění otázky, přidání odpovědí, odebrání odpovědí, změně znění odpovědí, změně správnosti odpovědí. Při odstranění testu, jsou odstraněny všechny otázky, odpovědi a komentáře které k němu náležejí. To je zajištěno pomocí cizích klíčů v databázových tabulkách a pomocí PHP.

Výsledný test je generován z informací uložených v databázi. Po spuštění a odeslání testu uživatelem je test vyhodnocen. Testy nejsou časově omezené a mohou se spouštět opakovaně.

4.2.9 Administrace systému

Uživatel s právy administrátora má stejné možnosti jako přispěvatel, navíc spravuje uživatelské účty a kontroluje obsah systému. Má možnost odebírat nevhodné komentáře, příspěvky v diskuzi, dokumenty a testy. S odebráním dokumentu nebo testu je svázáno odebrání všech komentářů k dokumentu nebo testu připojených. Administrátor má prostředky pouze pro odebírání, editovat může pouze příspěvky jím vložené. Správa uživatelských účtů spočívá v možnostech změny přístupových práv uživatele a zablokování/odblokování účtu. Kvůli splnění integritních omezení, nejsou uživatelé odstraňováni, pouze jsou označeni jako neaktivní a je jim zamezen přístup. Ze stejného důvodu nejsou z databáze odstraňovány ani komentáře a příspěvky v diskuzi, ale jsou pouze nahrazeny textem informujícím o odebrání příspěvku administrátorem. Přístupová práva mohou být nastavena na všechny úrovně, to znamená že v systému může být i více administrátorů. Administrátor nemůže editovat sám sebe, takže v systému vždy zůstane minimálně jeden administrátor. Po registraci se uživatel stává běžným uživatelem, pro změnu práv na přispěvatele musí kontaktovat administrátora.

4.2.10 Sledování novinek prostřednictvím RSS

Uživatelé mají možnost odebírat aktuality a informace o nejnovějších dokumentech prostřednictvím kanálů RSS 2.0 a Atom 1.0. Zobrazováno je vždy deset nejnovějších aktualit a dokumentů. Ověřování práv uživatelů je implementováno pomocí HTTP autentizace.

4.2.11 Uživatelská nastavení

Uživatelé mají možnost nastavit počet zobrazovaných příspěvků na jednu stranu (týká se aktualit a diskuze), počet zobrazovaných dokumentů na jednu stranu (při zobrazování dokumentů podle jednotlivých kategorií) a timeout. Toto nastavení je zadáváno prostřednictvím

formuláře. Po jeho odeslání jsou informace uloženy u uživatele prostřednictvím cookies. Pokud uživatel nenastavil žádné hodnoty, jsou použity defaultní hodnoty ze souboru `nastaveni.inc`.

4.3 Použité prostředky

Při vývoji byl použit následující software:

- Webový server Apache 2 s modulem PHP 5.
- Databázový server MySql 5.0.26.
- phpMyAdmin 2.10.0.2 (pro správu databáze).
- PSPad freeware editor 4.5.2 (textový editor).
- Adobe Photoshop (tvorba grafických součástí aplikace).
- Systém \LaTeX (pro sazbu textové části).

Kapitola 5

Závěr

Aplikace splňuje požadavky na ní kladené zadavatelem a je připravena k běžnému užití. V této podobě je použitelná jako všeobecně přístupná webová aplikace pro uživatele zajímající se o problematiku formálních jazyků a překladačů. Materiály týkající se této problematiky nejsou v elektronické podobě pro nejšířší veřejnost běžně dostupné. Tato webová aplikace značně usnadní přístup k těmto materiálům, což umožní všem zájemcům o problematiku formálních jazyků a překladačů rozšířit si své znalosti. Přispěvatelem se po domluvě s administrátorem může stát každý uživatel, ochotný podělit se o své znalosti. To v ideálním případě povede k nahromadění informací jak všeobecného, tak i velmi odborného rázu.

Po drobných úpravách (zrušení registrace, změně způsobu přihlašování uživatelů) by aplikace mohla být využita čistě pro potřeby výuky formálních jazyků a překladačů na akademické úrovni. Přístup by měli pouze studenti zapsaní do kurzu a v roli přispěvatele by vystupoval vyučující. V této podobě by aplikace umožňovala velmi snadné šíření dokumentů potřebných pro výuku a aktuálních informací týkajících se výuky. Pro studenty by aplikace nabízela možnost diskutovat své problémy jak s ostatními studenty, tak s vyučujícími.

Další vývoj aplikace by měl zajistit rozšíření redakční části systému jako například implementaci článkového subsystému, který by umožňoval vkládání článků prostřednictvím WYSIWYG editoru a správu článků, vytvoření a začlenění do systému webového rozhraní pro FTP server pro práci s rozměrnými soubory, vytvoření anketního systému, sledování různých statistik a propracování testové části. Dalším rozšířením by mohlo být využití Ajaxu například pro našeptávání při vyhledávání nebo pro zobrazení průběhu uploadu souboru na server.

Seznam použitých zdrojů

- [1] Hruška, T.: Informační systémy. [pdf], Poslední modifikace: prosinec 2006.
- [2] phpRS: Redakční systém. [online], [cit. 2007-04-28].
URL <http://www.supersvet.cz/dokumentace/cojeto.html>
- [3] The Apache HTTP Server Project: The Number One HTTP Server On The Internet. [online], [cit. 2007-04-27].
URL <http://httpd.apache.org/>
- [4] The phpMyAdmin Project: NEWS. [online], Poslední modifikace: 23. dubna 2007. [cit. 2007-04-27].
URL http://www.phpmyadmin.net/home_page/index.php
- [5] Ullman, L.: *PHP a MySQL: Návod k tvorbě dynamických WWW stránek*. Brno: Computer Press, 2004, ISBN 80-251-0063-4.
- [6] Wikipedia - the free encyclopedia: HTML. [online], Poslední modifikace: 24. dubna 2007. [cit. 2007-04-27].
URL <http://en.wikipedia.org/wiki/Html>
- [7] Wikipedie - otevřená encyklopedie: AJAX. [online], Poslední modifikace: 24. dubna 2007. [cit. 2007-04-27].
URL <http://cs.wikipedia.org/wiki/AJAX>
- [8] Wikipedie - otevřená encyklopedie: MySQL. [online], Poslední modifikace: 24. dubna 2007. [cit. 2007-04-27].
URL <http://cs.wikipedia.org/wiki/MySQL>
- [9] Wikipedie - otevřená encyklopedie: PHP. [online], Poslední modifikace: 24. dubna 2007. [cit. 2007-04-27].
URL <http://cs.wikipedia.org/wiki/Php>
- [10] Wikipedie - otevřená encyklopedie: Informační systémy. [online], Poslední modifikace: 28. dubna 2007. [cit. 2007-04-28].
URL http://cs.wikipedia.org/wiki/Informa%C4%8Dn%C3%AD_%syst%C3%A9m

Seznam příloh

- A** Zdrojový kód pro vytvoření databázových tabulek
- B** Návod na instalaci
- C** Uživatelská příručka
- D** Různé pohledy uživatelů

Příloha A

Zdrojový kód pro vytvoření databázových tabulek

```
DROP TABLE IF EXISTS uzivatele;  
DROP TABLE IF EXISTS onlineUzivatele;  
DROP TABLE IF EXISTS novinky;  
DROP TABLE IF EXISTS dokumenty;  
DROP TABLE IF EXISTS odpovedi;  
DROP TABLE IF EXISTS otazky;  
DROP TABLE IF EXISTS testy;  
DROP TABLE IF EXISTS komentare;  
DROP TABLE IF EXISTS diskuze;
```

```
CREATE TABLE uzivatele (  
    idUzivatele MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,  
    jmeno VARCHAR(30) NOT NULL,  
    prijmeni VARCHAR(30) NOT NULL,  
    uzivatelskeJmeno VARCHAR(30) NOT NULL,  
    email VARCHAR(40),  
    heslo VARCHAR(41) NOT NULL,  
    prava VARCHAR(16) NOT NULL,  
    datumRegistrace DATETIME NOT NULL,  
    aktivni VARCHAR (3) NOT NULL,  
    PRIMARY KEY (idUzivatele),  
    UNIQUE (uzivatelskeJmeno)  
) ENGINE=InnoDB;
```

```
CREATE TABLE onlineUzivatele (  
    idUzivatele MEDIUMINT(8) UNSIGNED NOT NULL,  
    uzivatelskeJmeno VARCHAR(30) NOT NULL,  
    cas BIGINT NOT NULL,  
    prava VARCHAR(16) NOT NULL,  
    PRIMARY KEY (idUzivatele)  
) ENGINE=InnoDB;
```



```

CREATE TABLE novinky (
    idNovinky MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    uzivatelskeJmeno VARCHAR(30) NOT NULL,
    datumVlozeni DATETIME NOT NULL,
    nadpis TINYTEXT NOT NULL,
    obsah TEXT NOT NULL,
    PRIMARY KEY (idNovinky)
) ENGINE=InnoDB;

CREATE TABLE dokumenty (
    idDokumentu MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    jmenoSouboru TINYTEXT NOT NULL,
    pojmenovani TINYTEXT,
    popis TEXT,
    vlozil VARCHAR(30) NOT NULL,
    typ VARCHAR(16) NOT NULL,
    mimeType VARCHAR(30) NOT NULL,
    keywords TINYTEXT,
    prava VARCHAR(16) NOT NULL,
    velikost VARCHAR(16) NOT NULL,
    datumVlozeni DATETIME NOT NULL,
    PRIMARY KEY (idDokumentu)
) ENGINE=InnoDB;

CREATE TABLE testy(
    idTestu MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    nazev TINYTEXT NOT NULL,
    popis TEXT NOT NULL,
    keywords TINYTEXT NOT NULL ,
    vlozil VARCHAR(30) NOT NULL,
    prava VARCHAR(16) NOT NULL,
    pocetOtazek SMALLINT NOT NULL,
    datumVlozeni DATETIME NOT NULL,
    PRIMARY KEY (idTestu)
) ENGINE=InnoDB;

CREATE TABLE otazky(
    idOtazky MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    idTestuRod MEDIUMINT(8) UNSIGNED NOT NULL,
    textOtazky TEXT NOT NULL,
    pocetSpravnych SMALLINT,
    PRIMARY KEY (idOtazky),
    FOREIGN KEY (idTestuRod) REFERENCES testy(idTestu) ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE odpovedi(
    idOdpovedi MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    idOtazkyRod MEDIUMINT(8) UNSIGNED NOT NULL,

```

```

    textOdpovedi TEXT NOT NULL,
    anoNe VARCHAR(5) NOT NULL,
    PRIMARY KEY (idOdpovedi),
    FOREIGN KEY (idOtazkyRod) REFERENCES otazky(idOtazky) ON DELETE CASCADE
) ENGINE=InnoDB;

```

```

CREATE TABLE komentare (
    idKomentare MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    idNadKomentare MEDIUMINT(8) UNSIGNED NOT NULL,
    idDokumentu MEDIUMINT(8) UNSIGNED NOT NULL,
    idTestu MEDIUMINT(8) UNSIGNED NOT NULL,
    uzivatelskeJmeno VARCHAR(30) NOT NULL,
    datumVlozeni DATETIME NOT NULL,
    nadpis TINYTEXT NOT NULL,
    obsah TEXT NOT NULL,
    PRIMARY KEY (idKomentare)
) ENGINE=InnoDB;

```

```

CREATE TABLE diskuze (
    idPrispevku MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
    idRodice MEDIUMINT(8) UNSIGNED NOT NULL,
    uzivatelskeJmeno VARCHAR(30) NOT NULL,
    datumVlozeni DATETIME NOT NULL,
    nadpis TINYTEXT NOT NULL,
    obsah TEXT NOT NULL,
    PRIMARY KEY (idPrispevku)
) ENGINE=InnoDB;

```

```

INSERT INTO uzivatele (jmeno,prijmeni,uzivatelskeJmeno,email,
heslo, prava,datumRegistrace,aktivni) VALUES('xxxxx','xxxxx','xxxxx',
'xxxxx',password('xxxxx'),'admin',now(), 'ano');

```

Příloha B

Návod na instalaci

Aplikace byla testována na webovém serveru Apache 2 s modulem PHP 5. Verze MySQL: 5.0.26.

B.1 Vytvoření databáze

Pomocí phpMyAdminu vytvořte databázi s porovnáním `utf8_bin` a s kódováním UTF-8. SQL příkaz:

```
CREATE DATABASE 'vaseDatabaze' DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
```

B.2 Vytvoření databázových tabulek

Upravte SQL dotaz `INSERT INTO` pro naplnění tabulky `uzivatele` v kódu uvedeném v příloze [A](#) tak, aby odpovídal údajům o budoucím administrátorovi systému.

Příklad:

```
INSERT INTO uzivatele (jmeno,prijmeni,uzivatelskeJmeno,email, heslo,prava, datumRegistrace,aktivni) VALUES('Josef','Novák','admin','admin@seznam.cz', password('admin'),'admin',now(),'ano');
```

Poté tento celý kód nad vaší databází vykonajte.

B.3 Vytvoření adresářové struktury na webovém serveru

Do kořenového adresáře vašeho webového serveru nakopírujte přiložené zdrojové kódy. O jednu úroveň výše vytvořte složku `upload`. V této složce budou uchovávány uložené dokumenty. Složka musí mít nastaveny příslušná práva. V kořenovém adresáři webového serveru upravte soubor `mysql_spojiti.php` tak, aby odpovídal vaší databázi.

Příklad:

```
<?php
$db = mysql_connect('localhost', 'vaseDatabaze', 'heslo');
if (!$db) die('Nelze se připojit '.mysql_error());
```

```
if (!mysql_select_db('vaseDatabaze', $db))
die('Databáze není dostupná '.mysql_error());
mysql_query('SET NAMES 'utf8'');
```

?>

V tuto chvíli se již můžete přihlásit jako administrátor s uživatelským jménem a heslem zvoleným v příkazu INSERT INTO uzivatele.

Příloha C

Uživatelská příručka

V této příručce jsou popsány jen akce, které nejsou na první pohled zřejmé. Ostatní akce jako je vkládání komentářů, novinek a jejich editace jsou intuitivní.

C.1 Registrace

Pro vytvoření uživatelského účtu je nutné se zaregistrovat. Všechny údaje kromě emailu jsou povinné. Jméno může obsahovat velká/malá písmena s nebo bez diakritiky, pomlčku, apostrof, mezeru a tečku. Znaků může být 2 až 30. Pro příjmení platí stejné pravidlo. Uživatelské jméno může obsahovat velká/malá písmena s nebo bez diakritiky, podtržítka a číslice. Heslo může obsahovat velká/malá písmena bez diakritiky, malá písmena z diakritikou a číslice. Znaků může být 4 až 20. Heslo je nutné zadat pro kontrolu dvakrát. Do poslední položky je nutné opsat šestimístné číslo vygenerované na obrázku. Ukázka registračního formuláře je na obrázku [C.1](#).

C.2 Odebírání novinek

Odebírat je možné 10 nejnovějších aktualit a dokumentů. K dispozici jsou ve formátu RSS 2.0 a Atom 1.0. Odebírat je můžete v každé RSS čtečce, která podporuje HTTP autentizaci.

C.3 Vyhledávání

Vyhledávání ignoruje velikosti písmen. Lze vyhledávat jednoslovní nebo víceslovné řetězce. Při vyhledávání víceslovního řetězce jsou zobrazeny dokumenty, ve kterých se vyskytují všechny části hledaného řetězce. K vyhledávacímu formuláři je připojena nápověda. Ukázka vyhledávacího formuláře je na obrázku [C.2](#).

Registrace

Křestní jméno:

Příjmení:

Uživatelské jméno:

Email:

Heslo:

Potvrďte heslo:

Opište číslo z obrázku



Obrázek C.1: Registrační formulář

Vyhledávání ?

Hledej podle:

Kategorie:

Typ souboru:

Hledaný řetězec:

Seřadit podle: datumu velikosti

Velikost: (kB) Od: Do:

Datum: (dd.mm.rrrr) Od: Do:

Obrázek C.2: Vyhledávací formulář

C.4 Upload souborů

Pouze pro přispěvatele a administrátora. Maximální velikost souboru může být 20MB. Klíčová slova ani popis nejsou povinná. Klíčová slova musí být oddělena mezerou nebo čárkou. Pokud dokument neodpovídá žádné z nabízených kategorií zvolte kategorii ostatní. Při nastavení práv na veřejné bude dokument přístupný všem uživatelům, při nastavení na neveřejné bude dokument přístupný pouze uživatelům přihlášeným. K formuláři je připojena nápověda. Ukázka formuláře je na obrázku C.3.



Upload

Soubor: Procházet...

(max 20MB)

Uložit pod jiným názvem?

Klíčová slova: ?

Popis:

Typ souboru:

Práva:

Ulož soubor

Obrázek C.3: Formulář pro upload souboru

C.5 Tvorba testů

Pouze pro přispěvatele a administrátora. Ve formuláři pro tvorbu testů je nutné vyplnit název testu, klíčová slova (oddělená mezerou nebo čárkou), popis testu a práva. Po odeslání je možné přidávat otázky. Otázka musí mít minimálně dvě, maximálně osm odpovědí. Správnost odpovědí se nastaví zatrhávacími tlačítky. Správných může být 0 až N odpovědí. Při pouze jedné správné odpovědi je otázka vygenerována jako 1 z N možných správných odpovědí, jinak je otázka generována jako 0 až N možných správných odpovědí. K formulářům je připojena nápověda.

C.6 Editace testů

Editace testů spočívá v odebrání otázky, přidání otázky, změně znění otázky či odpovědi, odebrání nebo přidání odpovědi a změně správnosti odpovědi. Ukázka testu v editačním módu je na obrázku C.4.

Přidat otázku

Otazka č. 1

otázka 1, 4 odpovědi a,d správně

- a. odpověď a
- b. odpověď b
- c. odpověď c
- d. odpověď d

Edituj tuto otázku Odeber tuto otázku

Otazka č. 2

Otázka 2, 3 odpovědi, správná a

- a. odpověď a
- b. odpověď b
- c. odpověď c

Edituj tuto otázku Odeber tuto otázku

Obrázek C.4: Ukázka testu v editačním módu

C.7 Administrace systému

Pouze pro administrátora. Administrátor spravuje uživatelské účty a obsah systému. Má možnost odstranit nevhodné komentáře, dokumenty, testy, příspěvky v diskuzi. Dále má možnost změnit práva uživatele a zamezit/povolit jeho přístup do systému. Obrázek D.3 ukazuje administrátorův pohled na uživatele.

Příloha D

Různé pohledy uživatelů

D.1 Pohled přihlášeného uživatele na dokumenty

The screenshot shows a web application interface for 'formální jazyky překladače'. The page is divided into several sections:

- Header:** 'formální jazyky překladače' in orange text on a blue background. Navigation links: 'Registrace', 'Přihlásit', 'Odhlásit'.
- Left Sidebar:** A vertical menu with items: 'Úvod', 'Novinky', 'Diskuze', 'Dokumenty', 'Nastavení'.
- Main Content Area:**
 - Dokumenty:** A sub-menu with 'Přednášky', 'Opory', 'Záznamy', 'Testy', 'Ostatní', 'Vyhledávání'. Below it, a filter menu with 'Pdf', 'Ms Word', 'Postscript', 'Ostatní'.
 - Table:** A table listing documents with columns: 'Název', 'Popis', 'Velikost', 'Datum'.

Název	Popis	Velikost	Datum
Serial-PHP.pdf	php	2349kb	2007-05-08 13:23:01
use_case_nepr.pdf	test	205kb	2007-05-08 13:14:33
ěščřžýáí	testovací dokument	713kb	2007-05-08 13:10:32
testovací dokument.pdf	testovací dokument	713kb	2007-05-08 13:10:07
er.pdf	Er diagram	713kb	2007-05-08 12:57:41
Serial-PHP.pdf	Seiál o php	2349kb	2007-05-08 12:54:45
- Right Sidebar:**
 - Přihlášen:** 'uživatel od: 13:40'
 - Nové komentáře:** 'komentář 5 | komentář 4 | Re: Komentář 3 | Re: Komentář 3 | Komentář 3 | Komentář 3 | komentář 2 | Re: koment 1 | koment 1 |'
 - Nové dokumenty:** 'test.ps | Serial-PHP.pdf | use_case_nepr.pdf | testovací dokument.pdf | testovací dokument.pdf | Rozhovor.wmv | Popelari.mpg | 1MKk.jpg | ladovska zima.mp3 | opice.wmv |'
 - Nové testy:** 'test 5 | test 4 | test 3 | test 2 | test 1 |'

At the bottom, there are RSS and Atom feeds for 'Novinky' and 'Dokumenty', and a 'W3C XHTML 1.0' logo.

Obrázek D.1: Pohled přihlášeného uživatele na dokumenty

D.2 Pohled přispěvatele na editaci svého testu

formální jazyky
překladače

Registrace Přihlásit Odhlásit

Přidat otázku

Otazka č. 1

otázka 1

a. odpověď a
b. odpověď b
c. odpověď c

Edituj tuto otázku Odeber tuto otázku

Otazka č. 2

otázka 2

a. odpověď a
b. odpověď b
c. odpověď c
d. odpověď d
e. odpověď e

Edituj tuto otázku Odeber tuto otázku

Přihlášen
prispevatel
od: 13:49

Nové komentáře
komentář 5 | komentář 4 |
Re: Komentář 3 | Re:
Komentář 3 | Komentář 3 |
komentář 2 | Re: koment 1
| koment 1 |

Nové dokumenty
test.ps | Serial-PHP.pdf |
use_case_nepr.pdf |
testovací dokument.pdf |
testovací dokument.pdf |
Rozhovor.wmv |
Popelari.mpg | 1Mkk.jpg |
ladovska zima.mp3 |
opice.wmv |

Nové testy
test 5 | test 4 | test 3 | test
2 | test 1 |

Mě dokumenty
ladovska zima.mp3 |
opice.wmv | php-triky.tgz |
sablon07.zip | er.pdf |
Serial-PHP.pdf |
php_manual_cs.chm |

RSS (Novinky) RSS (Dokumenty)
Atom (Novinky) Atom (Dokumenty)

W3C XHTML 1.0

Obrázek D.2: Pohled přispěvatele na editaci svého testu

D.3 Pohled administrátora na uživatele

formální jazyky
překladače

Registrace Přihlásit Odhlásit

Detail uživatele

Id	2	
Jméno	Filip	
Příjmení	Brychta	
Uživatelské jméno	uzivatel	
Email		
Práva	uživatel	Změnit práva
Datum registrace	2007-05-06 17:37:39	
Aktivní	Ano	Změnit

Přihlášen

admin
od: 13:55

Nové komentáře

komentář 5 | komentář 4 |
Re: Komentář 3 | Re:
Komentář 3 | Komentář 3 |
komentář 2 | Re: koment 1
| koment 1 |

Nové dokumenty

test.ps | Serial-PHP.pdf |
use_case_nepr.pdf |
testovací dokument.pdf |
testovací dokument.pdf |
Rozhovor.wmv |
Popelari.mpg | 1Mkk.jpg |
ladovska zima.mp3 |
opice.wmv |

Nové testy

test 5 | test 4 | test 3 | test
2 | test 1 |

Mé dokumenty

test.ps | Serial-PHP.pdf |
use_case_nepr.pdf |
testovací dokument.pdf |
testovací dokument.pdf |
Rozhovor.wmv |
Popelari.mpg | 1Mkk.jpg |

RSS (Novinky) RSS (Dokumenty)
Atom (Novinky) Atom (Dokumenty)

W3C XHTML 1.0

Obrázek D.3: Pohled administrátora na uživatele