



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

Jihočeská univerzita v Českých Budějovicích  
Pedagogická fakulta  
Katedra informatiky

## Bakalářská práce

Interaktivní edukační web se zaměřením na výuku  
španělského jazyka na ZŠ

Vypracoval: Vít Filipovský  
Vedoucí práce: PaedDr. Petr Pexa, Ph.D.

České Budějovice 2013

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

23.4.2013

.....

Děkuji vedoucímu bakalářské práce PaedDr. Petru Pexovi, Ph.D. za trpělivý a motivační přístup a také za cenné rady a metodické vedení práce.

## **Abstrakt**

Cílem bakalářské práce bude vytvořit webové stránky zaměřené na výuku španělského jazyka. Web bude umožňovat registrovat dva typy uživatelů. Učitele a žáka. Učitel bude moci vytvářet výukové materiály v podobě článků a testů. Ty bude moci kategorizovat podle náročnosti a tématu látky. Žák bude moci prohlížet články a vypracovávat testy, jejichž výsledky se budou archivovat. Stránky budou mimo jiné obsahovat databázi vybraných gramatických jevů a odkazů na ostatní webové portály zabývající se podobnou problematikou.

Ke zvolení uvedeného tématu mne přiměl fakt, že na českém webu je nedostatek neplacených otevřených databází španělské gramatiky a testů. Web nabídne vysvětlení jak základů pro úplné začátečníky, tak složitějších gramatických jevů určených pro pokročilé studenty. Web nebude sloužit jako placený e-learningový kurz, ale jako otevřená databáze gramatických jevů a testů doplňujících klasické studium či samostudium.

## **Klíčová slova**

Analýza výukových webů, databáze gramatických jevů, didaktický test, online testování, výuka španělského jazyka

## **Abstract**

Main aim of my work will be create web pages focused on teaching spanish. This web can allow sign up two types of users, teacher and student. Teacher will create tutorials in form of articles and achievement tests. This tutorials can be categorized in dependence of their level and content. Student will have possibility to browse articles and fill tests whose results will be stored. Page will have its own database of verb forms and hyperlinks to another webs in common.

I chose this issue because I had thought that in Czech web were not enough free web pages focused on this issue. My web page will offer as basics as advanced grammar. Web will not serve like paid e-learning course but like an open database of grammar and tests to extend study or selfstudy.

## **Keywords**

Analyze of learning sites, grammar databáze, online achievement test, spanish teaching

# Obsah

<b>1</b>	<b>ÚVOD</b> .....	<b>8</b>
<b>2</b>	<b>CÍLE</b> .....	<b>10</b>
<b>3</b>	<b>METODY</b> .....	<b>11</b>
<b>4</b>	<b>TEORETICKÁ ČÁST</b> .....	<b>12</b>
4.1	DIDAKTICKÝ TEST .....	12
4.1.1	<i>Druhy didaktických testů</i> .....	12
4.1.2	<i>Druhy testových úloh</i> .....	15
4.1.3	<i>Tvorba didaktických testů</i> .....	19
4.1.4	<i>Vlastnosti didaktického testu</i> .....	20
4.1.5	<i>Verifikace vlastností didaktického testu</i> .....	22
<b>5</b>	<b>SOUČASNÝ STAV PROBLEMATIKY</b> .....	<b>29</b>
5.1	ANALÝZA ČESKÝCH VÝUKOVÝCH WEBŮ ŠPANĚLSKÉHO JAZYKA .....	30
5.1.1	<i>www.spanelstina-online.cz</i> .....	30
5.1.2	<i>www.spanelstina-zdarma.cz</i> .....	31
5.1.3	<i>www.jazyky-online.info/spanelstina</i> .....	32
5.1.4	<i>www.estudiante.cz</i> .....	33
5.1.5	<i>jazyk.okhelp.cz/spanelstina</i> .....	34
5.1.6	<i>www.spanelstina-on-line.cz</i> .....	34
5.1.7	<i>Shrnutí</i> .....	35
<b>6</b>	<b>PRAKTICKÁ ČÁST</b> .....	<b>36</b>
6.1	POUŽITÉ TECHNOLOGIE .....	36
6.1.1	<i>HTML</i> .....	37
6.1.2	<i>CSS</i> .....	37
6.1.3	<i>PHP</i> .....	39
6.1.4	<i>MYSQL</i> .....	39
6.2	STRUKTURA APLIKACE .....	40
6.2.1	<i>Adresáře a jejich obsah</i> .....	41
6.2.2	<i>Rozdělení a funkce souborů</i> .....	42
6.3	STRUKTURA DATABÁZE .....	51
6.4	FUNKCE .....	54
6.4.1	<i>Inicializační</i> .....	54
6.4.2	<i>Databázové</i> .....	55
6.4.3	<i>Uživatelské</i> .....	56
6.4.4	<i>Formátovací</i> .....	57
6.4.5	<i>Kontrolní</i> .....	63
6.5	ZABEZPEČENÍ .....	63
6.5.1	<i>Kódování hesel</i> .....	64
6.5.2	<i>Omezení přístupu pomocí .htaccess</i> .....	64
6.6	OVLÁDÁNÍ .....	65
6.6.1	<i>Vkládání testů</i> .....	65
6.6.2	<i>Vkládání článků</i> .....	66
6.6.3	<i>Správa souborů</i> .....	67
6.6.4	<i>Seskupování studijních materiálů</i> .....	68
<b>7</b>	<b>ZHODNOCENÍ VÝSLEDKŮ</b> .....	<b>69</b>
7.1	<i>VYUŽITÍ</i> .....	69
7.2	<i>MOŽNOSTI</i> .....	69
<b>8</b>	<b>ZÁVĚR</b> .....	<b>70</b>

<b>9</b>	<b>SEZNAM POUŽITÝCH ZDROJŮ .....</b>	<b>72</b>
<b>10</b>	<b>SEZNAM TABULEK, OBRÁZKŮ A PŘÍLOH .....</b>	<b>73</b>
10.1	SEZNAM TABULEK .....	73
10.2	SEZNAM OBRÁZKŮ .....	73
10.3	SEZNAM PŘÍLOH .....	73
<b>11</b>	<b>PŘÍLOHY .....</b>	<b>74</b>

# 1 Úvod

Během svého studia španělštiny jsem několikrát narazil na problém absence česky lokalizovaných webů věnujících se výuce tohoto románského jazyka. Ačkoli světový internet nabízí velké množství stránek, které se věnují výuce slovíček či například testování, český internet je zatím, dle mého názoru, k výuce španělštiny nehostinný. V rámci teoretické části mé bakalářské práce jsem se proto rozhodl zmapovat několik webů, které se tématu výuky španělštiny věnují a na základě jejich analýzy a mého výzkumu připravit schéma aplikace, která by pokryla nároky začínajících i pokročilých hispanistů. Jako hlavní kritérium jsem si stanovil možnost uživatelů se spolupodílet na vzniku obsahu webu.

V praktické části bakalářské práce připravím na základě analýzy a výzkumu webovou aplikaci, která bude sloužit ke sdílení testů, článků, komentářů a učebních materiálů mezi učiteli a studenty španělského jazyka. Můj záměr je poskytnout učitelům a studentům takové prostředí, které umožní získat kvalitní zpětnou vazbu, jak směrem k učiteli, tak ke studentovi. Aplikace je zaměřena hlavně na vytváření, editaci, vyplňování a vyhodnocování didaktických testů. Učitelům umožní pomocí jednoduchého grafického rozhraní vytvářet didaktické testy. Ty pak poskytnou studentům ke zpracování. Na základě statistických údajů z vypracovaných testů pak bude učitel moci výsledky analyzovat a testy podle potřeby aktualizovat.

Tato metoda není jistě ničím novým a v rámci vzdělávacích institucí je v současnosti široce aplikována. Jmenujme například systémy moodle nebo eamos, které jsou pro komunikaci mezi učiteli a žáky na Jihočeské Univerzitě využívány asi nejvíce. Nevýhodou těchto systémů ale je, dle mého názoru, jejich podřazenost určité instituci. Většinou jsou materiály učitelů dostupné jen žákům, kteří danou instituci navštěvují, což zejména učitelům jazyků, kteří mohou pracovat ve více školách, značně ztěžuje situaci. Na řadu se potom dostávají různé způsoby, jak učitelé svým studentům mohou materiály napříč institucemi poskytovat. Mezi tyto metody patří zejména sdílení učiva pomocí internetových úložišť, emailových schránek nebo stránek na sociálních sítích. Učitel má tímto způsobem možnost poskytnout svoje materiály širší komunitě mimo instituci, na druhou stranu tyto metody komunikace umožní studentům převážně jen získat učební materiál.



Neodmyslitelným pomocníkem učitelů v současnosti jsou především didaktické testy, které při správném sestavení mohou v relativně krátkém čase pomoci odhalit studentovy slabiny. Didaktické testy, které po vyplnění okamžitě nabídnou výsledky, správné odpovědi, popřípadě komentáře a dají možnost studentovi subjektivně ohodnotit kvalitu testu jeho autorovi, jsou zajisté oboustranně prospěšné. Otevřeným systémem registrace uživatelů bych chtěl umožnit jednodušší rozšiřování a zkvalitnění obsahu webu.

## 2 Cíle

Cílem práce je vytvořit výukový web španělského jazyka a popsat jeho funkcionalitu. Mimo popisu funkcionalit se teoretická část zaměří na popis postupu při tvorbě didaktických testů.

Web bude ve finální úpravě sloužit učitelům a studentům španělského jazyka. Učitel bude moci přidávat vlastní články a testy, zobrazovat o nich statistické údaje a psát komentáře. Žák bude moci procházet přidané články a testy vypracovávat. U každého žáka se budou uchovávat statistické údaje o vypracovaných testech. Všichni uživatelé budou mezi sebou moci sdílet soubory, posílat jednoduché zprávy a využívat databáze slovesných tvarů.

### **3 Metody**

Celá práce se bude dělit na praktickou a teoretickou část. V části teoretické bych rád prostudoval několik existujících výukových webů a zhodnotil jejich klady a zápory. Rád bych zjistil, jaké možnosti uživatelům nabízejí a do jaké míry se mohou uživatelé podílet na tvorbě obsahu webu. Dále budu zjišťovat stav českých webů zabývajících se výukou španělského jazyka. Zejména mě zajímá struktura a uspořádání učebních materiálů, typy testových otázek, jejich návaznost na učební materiály a kvalita zpětné vazby, kterou poskytují uživatelé. Do teoretické části bude rovněž patřit prostudování publikace věnující se tvoření didaktických testů. Chtěl bych též věnovat velkou část popisu mého postupu při návrhu a realizaci vlastního výukového webu. V praktické části se budu věnovat vyhodnocení všech kritérií, která jsem v teoretické přípravě získal, tvorbě samotného webu a jeho ladění.

## 4 Teoretická část

Vzhledem k praktické části bakalářské práce, kterou je vytvoření webového rozhraní pro výuku španělštiny, jsem se rozhodl rozdělit teoretickou přípravu na dvě části. V části první analyzuji stav výukových webů španělského jazyka na českém internetu a v části druhé se věnuji přípravě a tvoření didaktických testů. Možnost tvorby a získávání statistických údajů pomocí didaktických testů je totiž hlavní prioritou při navrhování mé vlastní aplikace.

### 4.1 Didaktický test

Většina autorů, kteří se v minulosti zabývali problematikou didaktických testů, stanovila různé definice toho, co vlastně didaktický test ve skutečnosti je. Všichni se ale shodli na tom, že se didaktické testy používají ke stanovení úrovně osvojení učiva určitou skupinou studujících. Na rozdíl od ústního zkoušení nebo klasického zkoušení písemnou formou se používání didaktických testů dá normovat. Respektive již před tím, než učitel test připraví, stanoví si určitá pravidla pro jeho návrh, ověření, hodnocení a interpretaci výsledků. Hlavně jasná pravidla pro vyhodnocování testů z nich dělají velice dobrý nástroj pro objektivní zjišťování úrovně ovládnutí učiva.<sup>1</sup>

#### 4.1.1 Druhy didaktických testů

Didaktické testy můžeme rozlišit na několik druhů. Rozlišovacích kritérií je velký počet a většina autorů používá částečně upravené názvosloví. Podle formy zadání rozlišujeme testy zadané na papíře, zadané ústně nebo pomocí audionahrávky, zadané elektronicky, zadané speciálním způsobem a testy s kombinovaným zadáním. Dále můžeme testy třídit podle toho, zda ověřují kognitivní nebo psychomotorické dovednosti. Velmi často se při testování zaměřujeme na měření charakteristik výkonu, jakými jsou rychlost a úroveň. Také rozlišujeme didaktické testy podle interpretace výsledků na rozlišující a ověřující. Přehledný seznam druhů a kategorií didaktických testů je uspořádán v Tabulce č.1.

Klíčovým třídícím hlediskem pro konstrukci didaktického testu je interpretace výsledků. Z Tabulky č.1 je patrné, že interpretace může být dvojího druhu. Existují

---

<sup>1</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 184.

tedy testy rozlišující, jinými slovy také testy relativního výkonu a testy ověřující, které se rovněž nazývají testy absolutního výkonu.<sup>2</sup>

#### 4.1.1.1 Testy relativního výkonu NR – testy

Cílem těchto testů je porovnat výkony žáků z testované skupiny mezi sebou. Na základě porovnání poté sestavíme pořadí žáků ve skupině. V tomto případě závisí hodnocení jednotlivce přímo na výsledcích ostatních členů skupiny. Tento typ testu se využívá zejména při přijímacích řízeních, kdy potřebujeme porovnat výkon všech žáků mezi sebou. Didaktické testy tohoto typu by měly mít vysokou citlivost. Citlivostí testu a dalšími vlastnostmi se budeme více zabývat v kapitole 4.1.4 Vlastnosti didaktického textu. Testy relativního výkonu jsou používány hlavně kvůli vysokému stupni objektivity a diferenciaci.<sup>3</sup> Objektivita je docílena právě porovnáváním výsledků ve skupině, ale také stejnými podmínkami všech studentů během procesu testování.<sup>4</sup>

#### 4.1.1.2 Testy absolutního výkonu CR – testy

Tento typ testů se od předchozího typu liší hlavně tím, že výsledky neporovnává mezi sebou. Výsledky jsou porovnávány s normou, nebo-li s předem stanoveným vzorem, jakýmsi ideálním žákem. Na základě tohoto porovnání určíme hodnocení každého studenta. Hlavním cílem tedy není porovnat výsledky studentů, ale zjistit, zda daný student zvládl vybrané učivo. Při konstrukci tohoto testu musíme dbát na to, aby výběr učiva pro test byl skutečně reprezentativní a mohl o žakových vědomostech skutečně vypovídat.<sup>5</sup>

Kritérium zadání	Testy	Popis
Forma zadání	Zadané na papíře	Test je předkládán v tištěné podobě. Úlohy jsou zadávány formou textu nebo obrázku.

<sup>2</sup> Centrum pro zjišťování výsledků vzdělávání [online]. 2010 [cit. 2013-04-13]. Dostupné z: <<http://www.cermat.cz/didakticke-testy-1404034141.html>>.

<sup>3</sup> Druhy a příklady didaktických testů. *Infogram: portál pro podporu informační gramotnosti* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <<http://www.infogram.cz/article.do?articleId=2149>>.

<sup>4</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 186-187.

<sup>5</sup> Druhy a příklady didaktických testů. *Infogram: portál pro podporu informační gramotnosti* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <<http://www.infogram.cz/article.do?articleId=2149>>.

	Zadané ústně	Zadání testu je předčítáno učitelem a nebo je pouštěno pomocí audiozařízení.
	Zadané elektronicky	Testování probíhá pomocí počítače nebo jiného elektronického zařízení.
	Speciální	Testy pro studenty se zvláštními nároky a zdravotními postiženími. Mohou být zadávány například v podobě znakové řeči nebo Braillova písma.
	Kombinované	Test se skládá z několika podtestů, různých forem. Užívá se například u jazykových testů, kdy se test skládá z poslechové a textové části.
Ověřovaný výsledek učení	Kognitivní	Test ověřuje znalosti nebo textové dovednosti.
	Psychomotorické	Test ověřuje psychomotorické dovednosti.
Měřená charakteristika výkonu	Rychlost	Tento typ testu ověřuje rychlost, jakou je student schopen dané úkoly řešit. Menší náročnost ale větší počet úloh.
	Úroveň	Test ověřuje schopnost žáka řešit specifické úlohy. V tomto případě jsou úlohy náročnější, ale na jejich vypracování je dostatek času.
Interpretace výsledků	Rozlišující	Cílem je porovnat výsledky všech testovaných mezi sebou.
	Ověřující	Výsledky nejsou porovnávány s ostatními testovanými, ale s předem stanovenými kritérii. <sup>6</sup>
Dokonalost přípravy testu	Standardizované	Připravované speciální institucí a vybavené manuálem s vlastnostmi a výsledky testu.
	Nestandardizované	Připravovány učitelem.
Míra specifičnosti učení	Test výsledků výuky	Tento typ testu měří úroveň získaných vědomostí.
	Test studijních předpokladů	Tyto testy měří úroveň schopností jedince. Používá se zejména u přijímacích řízení.
Časové zařazení do výuky	Vstupní	Zjišťují počáteční úroveň znalostí.
	Průběžné	Dávají zpětnou vazbu učiteli během procesu výuky.
	Výstupní	Zadávají se na konci určitého období a slouží k hodnocení žáků.
Tématický obsah	Monotématické	Tyto testy zkouší učivo pouze z jednoho tématu.
	Polytématické	Tyto testy zkouší učivo z několika témat.
Míra objektivitý skórování	Objektivně skórovatelné	Skórují se na základě předem stanovených kritérií a jejich skórování může provádět jakákoli osoba.
	Subjektivně skórovatelné	U těchto úloh není možno stanovit jednoznačná pravidla pro skórování. <sup>7</sup>

**Tabulka č. 1: Kategorizace didaktických testů**

<sup>6</sup> Centrum pro zjišťování výsledků vzdělávání [online]. 2010 [cit. 2013-04-13]. Dostupné z: <<http://www.cermat.cz/didakticke-testy-1404034141.html>>.

<sup>7</sup> Druhy a příklady didaktických testů. *Infogram: portál pro podporu informační gramotnosti* [online]. 2013 [cit. 2013-04-13]. Dostupné z: <<http://www.infogram.cz/article.do?articleId=2149>>.

## 4.1.2 Druhy testových úloh

Testová úloha je hlavní částí každého didaktického testu. Úlohy mohou být různých druhů a každý druh má svá specifika a kritéria použití. Je důležité si uvědomit, že typ testové úlohy nemůžeme volit nahodile a že ne každý typ úlohy se hodí pro testování libovolné části učiva. Před samotnou volbou typu testové úlohy je nezbytné si určit přesně, jaké učivo má být zkoušeno a zda hodláme zjišťovat aktivní nebo pasivní vědomosti. V některých typech úloh totiž testovaný pouze vybírá z možností, a proto zkoušející nemá již možnost zjistit, zda látku aktivně ovládá nebo pouze rozpoznal správnou odpověď mezi variantami. Volba typu testové úlohy je tedy dost zásadní pro kvalitu a výpovědní hodnotu celého testu, tudíž bychom ji neměli zanedbávat, nýbrž důkladně promyslet.<sup>8</sup>

### 4.1.2.1 Uzavřené úlohy

#### Klasické uzavřené úlohy

Jedná se o velmi často používaný typ úloh. Z anglického Multiple Choice Items bývá překládáno jako položky mnohonásobné volby. Jedná se o typ úloh, které obsahují zadání definující problém a výběr z několika možností, kdy bývá nejčastěji jedna správná odpověď. Ostatní varianty, které obsahují klamnou odpověď, nazýváme distraktory. Při volbě distraktorů v těchto typech otázek je důležité volit adekvátní varianty. Pokud zvolíme distraktor, který logicky vůbec nezapadá do otázky, je již předem jasné, že ho testovaný nezvolí a jeho použití postrádá význam. Uzavřené úlohy mohou mít ještě dvě různé varianty použití. Pokud má zadání podobu přímé otázky, nazýváme tyto úlohy rozhodovacími. Zadání pak může být ještě ve formě neúplného tvrzení. V praxi to znamená, že zadání je věta s vynechaným slovem a testovaný toto slovo musí doplnit. Proto tyto úlohy nazýváme doplňovacími.<sup>9</sup>

---

<sup>8</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 19.

<sup>9</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 19.

## Úlohy situační

Tento typ úloh je velice podobný uzavřeným úlohám doplňovacím. Hlavní rozdíl spočívá v tom, že správné odpovědi a distraktory nejsou zadány přímo, nýbrž nějakou netradiční formou, jako například grafem či obrázkem. Za netradiční formu považujeme třeba i slovní úlohu v matematických testech. Hlavním kritériem tedy není forma zadání, ale fakt, že správná odpověď a distraktory ze zadání logicky vyplynou a nejsou předem vypsány jako seznam možných odpovědí. Například pokud mají žáci doplnit jednociferné číslo, mají teoreticky deset možností, ale v zadání tyto možnosti nejsou rozepsány. Tento typ úloh nám poskytne lepší obraz toho, jestli žák otázce skutečně porozuměl, na rozdíl od klasického typu s předem vypsány mi možnostmi, kde je šance, že žák vybral položku zcela náhodně.<sup>10</sup>

## Úlohy přiřazovací a uspořádací

Typ úloh hojně využíván ve výuce jazyků, kdy zejména pro zapamatování slovíček a frází je použito přiřazování originálu k překladu. Výhoda tohoto typu úloh oproti klasickým uzavřeným úlohám tkví v tom, že nabízíme větší počet distraktorů a tím snižujeme riziko náhodného uhodnutí. Pravděpodobnost náhodného uhádnutí můžeme vypočítat pomocí vzorce:

$$P = \frac{1}{k^n}$$

kde  $k$  je počet prvků jedné skupiny a  $n$  počet prvků skupiny druhé. Pokud bychom tedy například chtěli, aby testovaný přiřadil ke čtyřem českým slovům jejich správné překlady opět ze skupiny čtyř slov, je pravděpodobnost náhodného uhádnutí 0,004.<sup>11</sup> Úlohy uspořádací jsou vlastně jen obdobou předchozího modelu. Jeden sloupec odpovědí je zadán a druhý je sice nevyjádřený, ale můžeme si pod ním představit škálu čísel od jedné až po číslo rovné počtu odpovědí. Pravděpodobnost náhodného uhodnutí je zde opět malá.<sup>12</sup>

---

<sup>10</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 21.

<sup>11</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 28.

<sup>12</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 29.



## Dichotomické úlohy

Tento typ úloh je více známý pod pojmem ano – ne otázky, nebo-li otázky se dvěma variantami odpovědí. Opět se jedná jen o obdobu klasických uzavřených úloh. Jejich největší nevýhodou je velké procento uhodnutí (50 %), proto bychom tento typ úloh měli používat omezeně a při jejich konstrukci dbát na přítomnost jen jedné základní myšlenky.<sup>13</sup>

## Konstrukce uzavřených úloh

Pokud chceme konstruovat uzavřené úlohy kvalitní, měli bychom dodržovat několik pravidel, která sníží pravděpodobnost náhodného uhodnutí. V první řadě by měly být všechny odpovědi gramaticky konzistentní se zadáním a navzájem podobné. Častou chybou bývá rovněž jiná délka správné odpovědi. Pokud chceme, aby interpretace úlohy byla co nejjasnější, určitě bychom se měli vyhnout zadání v negativní formě, dvojitým záporům či příliš složitým formulacím.

Pro úlohy přiřazovací je důležité si předem stanovit princip, podle kterého bude přiřazení provedeno. Maximální počet prvků v jednom sloupci by měl být deset. Položky v jednom sloupci seřadíme chronologicky pro lepší orientaci. Závěrem je určitě dobré si uvědomit, jakým způsobem má testovaný k odpovědi dojít. Pokud správná odpověď závisí na tom, že testovaný použije třeba vedlejší výpočet nebo si nakreslí pomocný graf, není použití uzavřených úloh zcela vhodné, protože tento materiál nebude mít učitel při hodnocení k dispozici.<sup>14</sup>

### 4.1.2.2 Otevřené úlohy

#### Úlohy doplňovací

Tyto úlohy můžeme nazvat mezičlánkem mezi uzavřenými a otevřenými úlohami. Na rozdíl od uzavřených úloh doplňovacích v tomto případě varianty možných odpovědí nevyplývají ze zadání úlohy.

Úlohy otevřené doplňovací je vhodné použít v případě, že upřednostňujeme

---

<sup>13</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 30.

<sup>14</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 27.

znalost pojmu před jeho rozpoznáním ve skupině několika dalších. Je tedy vhodné je v testu použít, pokud chceme zjistit aktivní ovládnutí termínů.<sup>15</sup>

### **Úlohy se stručnou odpovědí**

Typ úloh s vysokou frekvencí používání je vhodný všude tam, kde nám nestačí jednoslovná odpověď, ale na druhou stranu bychom rádi získali odpověď co nejstručnější. Krátkou formu odpovědi však nezískáváme tím, že v zadání exaktně uvedeme maximální možnou délku odpovědi, nýbrž vhodnou konstrukcí otázky. Použití tohoto typu úloh je vhodné, pokud zjišťujeme znalosti vzorců, konkrétních vztahů nebo definic.<sup>16</sup>

### **Úlohy se širokou odpovědí**

Na rozdíl od všech ostatních typů, o kterých jsme se doposud zmínili, tento typ je charakteristický tím, že je skórován subjektivně nikoli objektivně. To v praxi znamená, že dva učitelé při svém hodnocení nedojdou ke stejným výsledkům. Je to způsobeno především samotnou podstatou tohoto typu úloh. Jedná se totiž o úlohy s velmi široce pojatým zadáním, které umožňuje žákům individuální přístup k odpovědi. Na jednu stranu tedy umožníme žákům co nejlepší možnost organizace vlastních myšlenek, na stranu druhou se připravíme o možnost objektivního hodnocení. Úlohy se širokou odpovědí tedy používáme hlavně v případech, kdy chceme poznat žákovy myšlenkové pochody a jeho způsob myšlení. V ostatních případech bychom měli zvolit jiný typ úlohy.<sup>17</sup>

### **Konstrukce otevřených úloh**

Ačkoliv by se mohlo zdát, že konstrukce otevřených úloh je snadná, neměli bychom se nechat touto představou příliš zmást. Ve všech případech je velice

---

<sup>15</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 31.

<sup>16</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 32.

<sup>17</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 35-36.

důležitá jasně položená otázka. U testů doplňovacích bychom měli dbát na to, abychom konstrukcí úlohy nenapovídali. Zejména se jedná o velikost vynechaných míst pro doplnění, ta by měla být vždy stejně dlouhá. Dále je častou chybou vynechání příliš mnoha slov následujících za sebou, zkoušený pak ztratí smysl a význam celého sdělení. Je-li to tedy možné, vynecháváme v jedné větě vždy jen jedno slovo a to takové, které v ní má podstatnou úlohu. Nechtěně napovědět můžeme i špatnou gramatickou konstrukcí věty. V případě užití otevřených úloh s krátkou odpovědí je důležité si uvědomit, že některé úlohy se svojí podstatou přibližují úlohám uzavřeným, protože žák prakticky rozmýšlí jen několik možných variant odpovědí. Příkladem mohou být úlohy z fyziky, kdy žák přemýšlí nad pohybem tělesa na kladce. Směry pohybu jsou předem determinované a žák vlastně vybírá jen z několika variant. V takovém případě bychom v zadání měli zkoušeného vyzvat, aby stručně popsal svojí úvahu. U otázek s širokou odpovědí musíme dbát především na to, aby studenti měli dostatek času k jejich vypracování. Určitě bychom v testu takových úloh neměli mít mnoho. Ze zadání úlohy musí být jasně patrné, na jakou problematiku se má student soustředit. Pro skórování otevřených úloh obecně bychom měli mít připravenou vzorovou odpověď. Testy by se neměly opravovat jako anonymní. Postupujeme tak, že nejprve opravíme stejnou položku u všech testů. V neposlední řadě bychom měli studentům vedle získaných bodů poskytnout i slovní hodnocení jejich odpovědi z důvodu kvalitnější zpětné vazby.<sup>18</sup>

### 4.1.3 Tvorba didaktických testů

Je žádoucí, abychom při vytváření testů, samotné aplikaci testů i jejich evaluaci využívali statistického způsobu myšlení i svého vyjadřování. Dodáme tím zkoušce měřicího charakteru. To, že se během aplikace didaktických testů zaměřujeme na větší objem testovaných, je jejich výhodou i slabinou.

Výhody didaktických testů jsou zejména v tom, že při jejich odborné konstrukci zmenšíme možnost výskytu logických omylů i subjektivního přístupu učitele k jednotlivým žákům. Ačkoliv je subjektivní přístup ke studentům v dnešní době velice žádaný, tak právě používání didaktických testů je velice vhodným prostředkem pro poskytnutí objektivních údajů o úrovni studentů i o kvalitě jejich

---

<sup>18</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 36.

přípravy. Bohužel, didaktické testy však postrádají svůj smysl tam, kde učitel vyžaduje do testovaného obsahu promítnout svoje morální, estetická a další hlediska, která jsou závislá na osobnosti jedince.<sup>19</sup>

#### 4.1.4 Vlastnosti didaktického testu

Největší chybou je začínat vytváření testu návrhem testových úloh. Je důležité se nad testem zamyslet jako nad celkem a nezaměřovat se jen na jeho dílčí prvky. Jinak se nám může stát, že navrheme sice množství úloh, ty ale nebudou dohromady působit konzistentně a ve finále nám vznikne nekvalitní a nevyvážený test. Nekvalitní test se vyznačuje zejména tím, že nezkouší rovnoměrně celý obsah učiva a požaduje po zkoušených jen pouhé opakování zapamatovaných údajů. Právě v nekvalitních testech bývá úloh tohoto typu velké množství a využití potenciálu didaktických testů je tím zmařeno. Kvalitní test tedy sestavíme jedině tak, že před zbrklým navrhováním úloh upřednostníme důkladné promyšlení návrhu celého testu.<sup>20</sup>

Během plánování testu bychom měli dbát na čtyři základní kritéria, která má správný didaktický test splňovat. Jsou jimi objektivita, validita, reliabilita a ekonomičnost. Dodržení základních norem pro konstrukci a kontrolu didaktických testů nám pomůže sestavit kvalitní test, který nám poskytne dobrou zpětnou vazbu o studentských vědomostech i jejich připravenosti.<sup>21</sup>

Při plánování testu se musíme především zajímat o to, k jakému účelu daný test vytváříme. Test můžeme například použít pro zjištění vstupních znalostí studentů před probíráním látky či jako test výstupních kompetencí po probrání určité látky. Test může rovněž sloužit jako průběžný nebo-li informativní, kdy během probírání látky zjišťujeme, do jaké míry si ji žáci již stihli osvojit.<sup>22</sup>

Další věcí, kterou si musíme ujasnit, je obsah testu. Standardním postupem bývá učivo rozdělit na části jako například fakta, pojmy, definice a podobně. Pro každou část stanovíme počet testových úloh a k jednotlivým úlohám přiřadíme, jaké má být

---

<sup>19</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 5.

<sup>20</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 194.

<sup>21</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 6.

<sup>22</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 194.

dosaženo úrovně osvojení. K tomuto účelu můžeme použít třeba Bloomovu taxonomii výukových cílů.<sup>23</sup>

#### 4.1.4.1 Objektivita testu

Objektivitou zde máme na mysli především nezaujatost v průběhu testování a během následného hodnocení testu. Oproti ostatním druhům zkoušení nám dává používání didaktických testů jistotu určité spravedlnosti. Tento fakt byl potvrzen na základě několika experimentů a zejména oproti klasickému písemnému zkoušení, ve kterém různí učitelé dosahovali odlišných hodnocení, je didaktický test nezaujatý. Je totiž v zásadě nemožné, aby zkoušející bez toho, aby porušil základní pravidla pro testování, vyhodnotil test odlišně od jiného zkoušejícího.<sup>24</sup>

#### 4.1.4.2 Validita testu

Validním nebo-li platným testem můžeme nazvat takový test, který plní předem stanovené cíle a požadavky. Pokud srovnáme objektivitu testu s jeho validitou, zjistíme, že objektivita je vlastností nejzřejmější. Naproti tomu validita, ač není tak zřejmá, je vlastností nejdůležitější. Kromě validity plní veškeré ostatní charakteristiky spíše pomocnou funkci a mají význam jen v případech, „vedou-li k diagnosticky efektivnímu – validnímu – výsledku“.<sup>25</sup> To, že je didaktický test objektivní i reliabilní, nám však ještě nezaručuje jeho validitu. Příkladem nevalidního testu může být použití testu určeného pro žáky sedmé třídy pro testování žáků tříd devátých a nebo naopak. Test neplní účel, pro který byl vytvořen, a tak validitu ztrácí.<sup>26</sup>

#### 4.1.4.3 Reliabilita testu

Reliabilita nám říká, jak silným a přesným měřicím nástrojem se náš didaktický

---

<sup>23</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 195.

<sup>24</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 6.

<sup>25</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 7.

<sup>26</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 7.

test stal. Je to veličina, která nám vyjadřuje schopnost testu dosáhnout konzistentních výsledků během jeho užívání. Vysoká reliabilita testu vypovídá o tom, že jeho výsledek není ovlivněn náhodnými vlivy, náladou zkoušeného či jeho okolím. Reliabilita testu se prakticky skládá ze dvou možných druhů chyb. Vliv na tyto chyby má psychický, ale i fyzický stav zkoušeného a především samotná konstrukce testu, nejednoznačnost otázek, nedostatky během procesu testování či špatné hodnotící postupy.<sup>27</sup>

#### **4.1.4.4 Ekonomičnost testu**

Kvalitně připravený didaktický test šetří čas učitele i jeho studentů. Porovnáme-li například didaktický test s ústním zkoušením, dojdeme k závěru, že ústní zkoušení je časově o dost náročnější, nehledě na to, že se učitel může věnovat pouze jedinému studentovi. Didaktický test je proto účelné použít vždy, pokud nám má nabídnout stejně hodnotné výsledky jako ústní zkoušení.

Další výhodou testu je jeho jednoduchá korekce a konzistentní hodnocení dosažených výsledků. Prokázat přímou finanční úsporu při použití didaktických testů nelze už jen proto, že učitel musí testy minimálně vytisknout a ústní zkoušení se zpravidla obejde bez pomůcek. Navýšení výdajů by však nemělo učitele od používání testů odradit, neboť se vynaložení těchto prostředků projeví ve vyšší kvalitě a v rozsahu získaných údajů. Připravit správně konstruovaný test je pro učitele obtížnější než příprava jiných druhů zkoušení. Odměnou však učiteli bude, že žáci pochopí učivo hlouběji. Učitel může postupem času svoje testy vylepšovat a rozšiřovat jejich zásobu, to znamená, že z dlouhodobého hlediska nabízí používání testů značnou úsporu času.<sup>28</sup>

#### **4.1.5 Verifikace vlastností didaktického testu**

Chceme-li, aby byl námi vytvořený test spolehlivý a opravdu měřil ty vědomosti, pro které byl konstruován, je nezbytné ověřit jeho funkčnost na dostatečně velkém

---

<sup>27</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 7.

<sup>28</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 8.

vzorku testovaných jedinců. Až takovéto ověření nám poskytne kvalitní výsledky o skutečných vlastnostech testu. Během verifikace vlastností didaktického testu posuzujeme nejen kvalitu a vlastnosti jednotlivých úloh, ale i vlastnosti celého testu.<sup>29</sup>

#### 4.1.5.1 Rozbor úloh

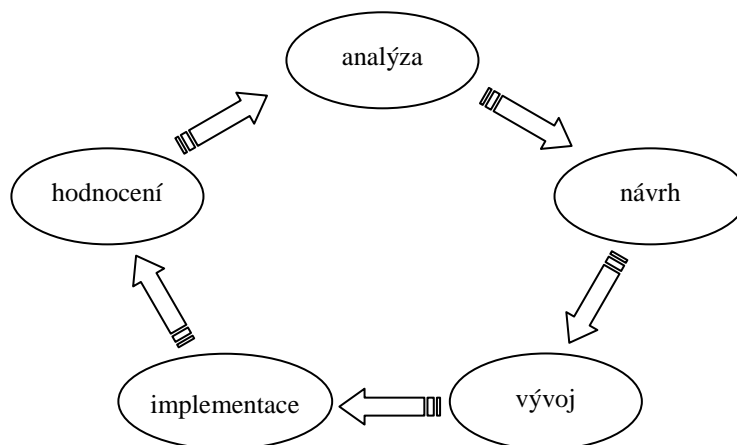
Rozbor úloh je důležitým krokem pro vytvoření kvalitních didaktických testů. Tento rozbor nebo-li analýzu je však možné uskutečnit až při získání dostatečného množství vypracovaných testů. Z toho plyne, že nelze vypracovat kvalitní didaktický test bez potřebné zpětné vazby a kontroly. Kvalitní úlohy tedy získáme až po jejich odzkoušení a následném odstranění jejich nežádoucích částí. Tento systém je uzavřený kruh a můžeme říct, že kvalita úlohy roste s počtem jejího nasazení v praxi a jejích korekcí. K účelu návrhu můžeme použít například modelu ADDIE. Model ADDIE, jehož název je akronymem slov analyse (analýza), design (návrh), development (vývoj), implementation (užití), evaluation (hodnocení), je při návrhu školení nebo kurzů využíván velmi často. Základním pilířem celého modelu je právě hodnocení. Každá z pěti částí modelu by na svém konci měla být zhodnocena.

Například, zda byl její návrh proveden dobře nebo jestli bylo dosaženo požadovaných cílů. Pro správnou funkci modelu je rovněž nezbytné, aby žádná z fází vývoje nebyla vynechána.<sup>30</sup>

---

<sup>29</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 195.

<sup>30</sup> STÁRKOVÁ, Dagmar. *Model ADDIE při vytváření koncepce výuky a jeho aplikace* [online]. 2012 [cit. 2013-04-13]. Dostupné z: <[http://it.pedf.cuni.cz/strstud/edutech/2012\\_Addie\\_Starkova/](http://it.pedf.cuni.cz/strstud/edutech/2012_Addie_Starkova/)>.



**Obrázek č. 1: Struktura modelu ADDIE**

Během samotného rozboru úloh se zaměříme především na jejich obtížnost, citlivost a dále také na analýzu vynechaných nebo nesprávných odpovědí. Tento rozbor pro nás bude stěžejním zdrojem informací o kvalitě připravovaného didaktického testu.

#### **4.1.5.2 Obtížnost úloh**

Znát obtížnost testových úloh je pro správné sestavení testu velice užitečné. V praxi je pak časté řazení úloh v testu podle jejich obtížnosti. Obtížnost úlohy určíme buď výpočtem indexu obtížnosti  $P$  a nebo výpočtem hodnoty obtížnosti  $Q$ . Index obtížnosti  $P$  nám udává procento testovaných osob ve skupině, které daný úkol zodpověděly správně, zatímco hodnota obtížnosti  $Q$  nám udává procento testovaných, kteří úlohu vypracovali špatně a nebo vůbec.

$$P = 100 \frac{n_s}{n}$$

V tomto vzorci pro výpočet indexu obtížnosti  $P$ , použijeme proměnnou  $n_s$  což je celkový počet testovaných osob ve skupině, které odpověděly správně a proměnnou  $n$ , která označuje celkový počet osob ve skupině.

$$Q = 100 \frac{n_s}{n}$$

Ve vzorci pro výpočet hodnoty obtížnosti  $Q$  zastupuje proměnná  $n_s$  množství testovaných, kteří zodpověděli úlohu špatně a nebo vůbec a proměnná  $n$  celkový



počet testovaných osob. Vysokou obtížnost úlohy poznáme podle vysokých hodnot obtížnosti  $Q$  nebo nízké míry indexu obtížnosti  $P$ . Zvláště obtížné úlohy dosahují hodnot obtížnosti vyšších než 80, naopak velmi snadné úlohy dosahují hodnot obtížnosti maximálně 20. U úloh, kde se hodnota obtížnosti blíží k číslu 100, bychom měli zvážit, zda je v testu vůbec použít. Oproti tomu jednu či dvě extrémně lehké úlohy můžeme použít z psychologických důvodů jako úvodní úlohy testu. Dodáme tím zkoušenému více sebejistoty.<sup>31</sup>

#### 4.1.5.3 Citlivost úloh

Citlivost úlohy je vlastnost, která nám určí, do jaké míry úloha rozlišuje mezi úspěšným a slabým studentem. Pro výpočet koeficientu citlivosti se uvádějí různé způsoby. Pro učitelské testy je nejčastějším způsobem výpočtu porovnání odpovědí nejsilnější a nejslabší třetiny respondentů.

$$d = \frac{N_H - N_D}{N_{DIF}}$$

Před začátkem výpočtu je důležité si studenty rozdělit na dvě stejně početné skupiny, přičemž jednu skupinu budou tvořit žáci s celkově lepším prospěchem a skupinu druhou žáci horší. Pro výpočet koeficientu citlivosti  $d$  je tedy potřeba znát  $N_H$ , což je počet testovaných s celkově lepšími výsledky, kteří tuto otázku vyřešili správně.  $N_D$  jsou ti respondenti s celkově horšími výsledky, kteří danou otázku rovněž zodpověděli správně. Poslední proměnná  $N_{DIF}$  je celkový počet studentů z jedné skupiny. Výsledkem výpočtu pak bude interval  $\langle -1, 1 \rangle$ , ve kterém se citlivost naší úlohy bude pohybovat. Praxí potvrzené výsledky naznačují, že ideální koeficient citlivosti je 0,3 a vyšší. Naopak zcela špatná citlivost je záporná, jelikož podle ní můžeme poznat, že otázku řešili lépe slabí žáci než ti prospěchově lepší. Problémem v tomto případě bývá špatná formulace nebo nepřesnost zadání. Prospěchově slabší žáci si nepřesnosti nevšimnou a budou řešit úlohu bez komplikací, naproti tomu silnější studenti budou nepřesnosti formulace věnovat příliš mnoho pozornosti a správnost jejich řešení tím bude ovlivněna.<sup>32</sup>

---

<sup>31</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 196.

<sup>32</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 87.

„To, že v položce chybuje stejný nebo větší počet žáků v testu úspěšných, než žáků neúspěšných, může mít různé důvody, které je třeba objasnit. Při dostatečné velikosti vzorku však představuje vždy varovný signál.“<sup>33</sup>

#### 4.1.5.4 Analýza nesprávných a vynechaných odpovědí

Během vyhodnocování testů může nastat situace, že v některých úlohách žáci až příliš často vynechávají odpovědi. Této skutečnosti bychom měli věnovat náležitou pozornost, protože kromě neznalosti daného učiva může vynechaná odpověď znamenat i její špatné pochopení nebo nedostatek potřebného času. Některé zdroje uvádějí, že bychom měli věnovat více pozornosti takovým otevřeným úlohám, jež vynechalo kolem 30 – 40 % studentů. V případě úloh uzavřených je toto číslo okolo 20 % studentů.<sup>34</sup>

Nejjednodušší rozbor nesprávných odpovědí je u úloh s výběrem správné odpovědi. Při kontrole těchto úloh nám stačí zjistit, zda jsou všechny nesprávné varianty pro studenty dostatečně lákavé. Poznáme to hlavně podle rovnoměrného vyplňování možných variant. Pokud existuje taková varianta, kterou vybírá jen malý počet testovaných nebo nikdo, je jisté, že daný distraktor (klamnou nabídku) musíme nahradit distraktorem jiným, popřípadě můžeme tento distraktor, z úlohy úplně odstranit.<sup>35</sup>

Pokud chceme dělat rozbor úloh s otevřenou odpovědí, měli bychom postupovat tak, že nejprve všechny chyby roztřídíme do kategorie základní a vedlejší. Chyby základní jsou ty, které jsou způsobeny opravdovou neznalostí dané látky, zato chyby vedlejší jsou zpravidla způsobeny různými nedbalostmi jako je přehlédnutí, numerická chyba, špatná čitelnost úlohy a jiné. Převaha vedlejších chyb nad hlavními může o testové úloze vypovídat, že její řešení je v podstatě věcí náhody. Pokud v úlohách převažují chyby vedlejší nad základními, měli bychom určitě zvážit jejich vymazání z testu.<sup>36</sup>

---

<sup>33</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 90.

<sup>34</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 197.

<sup>35</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 197.

<sup>36</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 197.

#### 4.1.5.5 Rozbor testu jako celku

Nesmíme opomenout, že testové úlohy nestojí samostatně a že jsou součástí jednoho celku. Proto musíme dbát na to, aby didaktický test jako celek plnil určitá kritéria. Těmito kritérii jsou validita a reliabilita didaktického testu.<sup>37</sup>

Validita didaktického testu je jeho nejpodstatnější charakteristikou. Pro zjištění validity testu neexistují žádné ověřené výpočty. Musíme se proto spolehnout na náš zdravý úsudek. Hlavní otázkou pro nás je, zda je test složen opravdu z podstatných prvků učiva a jestli odpovídá obsahu látky, která je probírána. Jinými slovy, jestli test opravdu zkouší to, co bylo naším záměrem. Test obsahově validní nám může pomoci posoudit jiný učitel, popřípadě odborník na danou tematiku.<sup>38</sup>

Reliabilita didaktického testu je mírou spolehlivosti testu. To v praxi znamená, že při opakovaném použití testu za stejných podmínek musíme dospět k totožným nebo téměř totožným výsledkům. Pro jasné určení reliability didaktického testu použijeme koeficient reliability. Ten může nabývat hodnot od 0 do 1. Při koeficientu rovném 0 je test naprosto nereliabilní, to znamená nespolehlivý a při koeficientu rovném 1 je naopak maximálně spolehlivý. Požadovaná hodnota koeficientu pro individuální testování je 0,80. Spolehlivost testem naměřených výsledků klesá spolu s koeficientem reliability.<sup>39</sup>

Reliabilita testu do značné míry závisí na počtu jeho úloh. Obecné pravidlo říká, že reliabilita testu narůstá s počtem úloh. Například u testů s počtem úloh menším než 10 bude hodnota koeficientu reliability maximálně 0,60. Pro získání konkrétní hodnoty reliability didaktického testu použijeme Kuderův-Richardsonův vzorec.

$$r_{kr} = \frac{k}{k-1} \left( 1 - \frac{\sum pq}{s^2} \right)$$

V tomto vzorci zastupuje proměnná  $k$  počet úloh v testu,  $p$  je podíl studentů, kteří zodpověděli danou otázku správně a celkový počet testovaných,  $q = (1 - p)$  a  $s$  je směrodatná odchylka všech studentských výsledků.<sup>40</sup>

---

<sup>37</sup> HRABAL, Vladimír, LUSTIGOVÁ, Zdena a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha, 1992, s. 6.

<sup>38</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 198.

<sup>39</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 198.

<sup>40</sup> CHRÁSKA, Miroslav. *Metody pedagogického výzkumu*. Havlíčkův Brod, 2007, s. 199.

#### 4.1.5.6 Hodnocení výkonu studenta percentilovou škálou

Pro dostatečné ohodnocení studentova výkonu nám nestačí znát pouze počet dosažených bodů v testu. Výkon studenta bychom měli porovnat s výkony ostatních testovaných. Teprve poté získáme komplexní přehled o tom, jakých výkonů studenti skutečně dosáhli. Toto srovnání neslouží jen učitel, nýbrž i samotným studentům, kteří v rámci přirozené lidské soutěživosti mohou do budoucna na svůj výkon klást vyšší nároky. Při tomto srovnávání získá učitel jasnou představu o studentově výkonu. Pro porovnání testovaných výkonů využijeme metodu standardizace pomocí percentilové škály. Tato škála nám udává počet studentů v procentech, kteří dosáhli horších výsledků. Můžeme tak stanovit relativní pořadí studenta ve skupině testovaných. Pro výpočet použijeme následujícího vzorce.

$$PR = 100 \frac{n_k - \frac{n_i}{2}}{n}$$

V tomto vzorci nám  $PR$  označuje percentilové pořadí daného studenta, proměnná  $n_k$  zastupuje kumulativní četnost zjišťovaného výsledku,  $n_i$  vyjadřuje četnost výsledku a  $n$  je celkový počet studentů v testovaném vzorku.

## 5 Současný stav problematiky

Webové stránky zabývající se výukou se staly populárním prostředkem doplnění klasické výuky. Hlavním důvodem je jejich interaktivita a možnost okamžité zpětné vazby směrem ke studentovi. Dle mého názoru tak mohou webové stránky studentovi nabídnout zdokonalení zejména z hlediska zapamatování si pojmů a jejich následné aplikace. Student si přečte výklad dané problematiky a poté si své nově nabyté vědomosti může okamžitě prověřit v testu, který bývá na kvalitních výukových stránkách k teoretickému výkladu přiložen. Po vyhodnocení testu vidí student okamžitě své výsledky. Úskalím této zpětné vazby u některých webů je, že nenabízejí vysvětlení jednotlivých odpovědí a tak student pouze pozná, zda odpověděl správně nebo ne. Dobré výukové aplikace tak rozpoznáme dle kvality zpětné vazby, kterou poskytují.

Výukové weby můžeme rozdělit na několik typů. Převážnou část tvoří weby encyklopedického typu, to znamená weby, na kterých se uživatel pouze seznámí s problematikou. Na takovýchto stránkách nalezneme strukturovaný text obsahující výklad, poznámky, obrázky, hypertext či otázky k výkladu. Jedná se však většinou o statické stránky bez možnosti otestování nabytých vědomostí. Dalším často se objevujícím typem jsou weby, které jsou pravým opakem těch encyklopedických a nabízejí pouze testy. Tyto weby se nezajímají o výklad teorie, v nejlepším případě nabízejí seznam hypertextů odkazující na jiné stránky, kde se výklad nachází. Tento způsob není úplně vhodný, jelikož validita odkazů nebývá kontrolována a ty pak odkazují na již neexistující weby. Výhodou takovýchto webů je poskytnutí zpětné vazby. Nejlepší variantou je tak kombinace dvou předchozích. Na českém internetu je takovou zářivou ukázkou webový portál [www.helpforenglish.cz](http://www.helpforenglish.cz), který nabízí vše, co jen může student anglického jazyka od výuky na internetu požadovat. Portál umožňuje registraci uživatele a uchovávání výsledků jeho testů. Nalezneme zde přehlednou databázi článků o gramatických jevech, reáliích, výslovnosti a mnohem více dalších kategorií. A v neposlední řadě je téměř ke každému článku k dispozici test s velice kvalitně připraveným komentářem ke každé odpovědi.

Webových portálů, jako je výše zmiňovaný, existuje celá řada. Chtěl jsem jen na příkladu demonstrovat, co by měl výukový web uživateli poskytnout. Výukové weby mají před sebou jistě slibnou budoucnost, je však na uživatelích, aby rozhodli, které

weby obstojí v konkurenci a budou hojně užívány a zdokonalovány a nebo které naopak zaniknou kvůli své strnulosti a absenci zpětné vazby.

Pokusil jsem se provést průzkum mezi učiteli španělštiny na základních školách, abych zjistil jejich názor na současný stav španělských výukových webů a získal vstupní podklady pro případný výzkum. Vzhledem k tomu, že je španělština v současnosti na základních školách pouze volitelným jazykem, podařilo se mi získat informace pouze od 6 učitelů. Dospěl jsem k názoru, že takto nízký počet by jistě nebyl pro relevantní výzkum dostačující. Nicméně připomínky oslovených učitelů mi byly při návrhu aplikace dobrou inspirací.

## **5.1 Analýza českých výukových webů španělského jazyka**

Další část práce věnuji analýze několika českých webů, které se zabývají výukou španělštiny. Rád bych popsal jejich klady a zápory a na základě tohoto vyhodnocení pak postupoval při vytváření vlastní webové stránky. Jednotlivé weby budu vždy hodnotit ze tří pohledů a to podle množství, kvality a dostupnosti studijních materiálů, dále podle toho, do jaké míry je uživatelům umožněno podílet se na rozšiřování obsahu webové stránky a v neposlední řadě se chci zaměřit na grafické zpracování. Analyzoval jsem šest webových stránek, které se dostaly na přední pozice ve vyhledávači po zadání klíčových slov španělština, online a zdarma. Klíčové slovo zdarma bylo použito úmyslně z důvodu filtrace placených kurzů a jiných placených služeb. Na začátku analýzy vždy uvedu několik technických údajů o doméně a poté bude následovat vlastní analýza podle kritérií, která jsem popsal výše.

### **5.1.1 [www.spanelstina-online.cz](http://www.spanelstina-online.cz)**

koordinátor projektu:	Petra Vildová
poslední aktualizace:	2010
software:	<a href="http://www.webnode.cz">www.webnode.cz</a>

Podle autorů tohoto webu se jedná o stránky nabízející studijní materiály, testy, informace o zkouškách, odkazy a možnost komunikovat pomocí veřejného fóra.

Stránky mne na první pohled zaujaly kvalitně zpracovaným a přehledným layoutem. Po prohlédnutí webu jsem však zjistil, že studijních materiálů ani testů se zde nenachází mnoho a jejich kvalita a zpracování rovněž nejsou oslnivé. Výsledek testování je zobrazen pouze ve vyskakovacím javascriptovém okně jako upozornění, že uživatel dostal x bodů z maxima. Zpětná vazba je sice rychlá, ale velmi stručná, bez náležitých vysvětlení či komentářů. Po důkladnějším prohlédnutí webu jsem usoudil, že stránka slouží převážně k prezentaci jazykových kurzů a jako rozcestník. Vítanou položkou v menu pro mne byl seznam doporučených učebnic pro samostudium či souhrn základních španělských frází. Nicméně datum poslední aktualizace 2010 není moc dobrou vizitkou o aktuálnosti webu. Testů i článků je k dispozici celkem osm, to ovšem není nijak převratné číslo a jelikož se jedná o stránky, na kterých uživatelé nemohou přispívat a rozšiřovat tak obsah, řekl bych, že je to číslo dosti nedostačující. V mém hodnocení tedy web [www.spanelstina-online.cz](http://www.spanelstina-online.cz) obstál jen z hlediska dobrého grafického zpracování. Z hlediska rozsahu poskytovaných studijních materiálů i z hlediska možnosti uživatelů podílet se na rozšiřování obsahu stránek bych tento web hodnotil nedostatečnou známkou.

### **5.1.2 [www.spanelstina-zdarma.cz](http://www.spanelstina-zdarma.cz)**

provozovatel:	Máριο Vondruška
autor:	Mgr. Marek Vít (Vitware s.r.o.)
poslední aktualizace:	2013
software:	redakční systém od Vitware s.r.o

Tmavě šedá stránka, která na první pohled barvami trochu odradí, je ve skutečnosti jedním z velice kvalitních zdrojů učebních materiálů pro studenty španělštiny.

Rozhraní použité na webových stránkách je od společnosti Vitware s.r.o. Mgr. Marka Víta a Petra Hudíka. Společnost se věnuje vývoji aplikací pro výuku angličtiny a spravuje web [www.helpforenglish.cz](http://www.helpforenglish.cz), což je velice obsáhlá databáze anglické gramatiky a slovní zásoby.

Stránka [www.spanelstina-zdarma.cz](http://www.spanelstina-zdarma.cz) nám tak nabízí veškeré funkce svého mateřského webu. Výukové materiály jsou seřazeny do několika kategorií a stránka

má i vlastní obrázkový slovník. K většině článků existuje i příslušný didaktický test, který po vyplnění nabídne uživateli velice kvalitní zpětnou vazbu. U každé položky barevně zvýrazní dobré a špatné odpovědi a nabídne komentář. Na konci testu si uživatel může prohlédnout svojí procentuální úspěšnost a porovnat jí s průměrnou úspěšností ostatních uživatelů. Rozdělení kategorií je velice intuitivní a mimo gramatiky, slovní zásoby a testů zde nalezneme i kategorie jako výslovnost, četba, maturitní témata nebo reálie. V neposlední řadě je zde připravený kurz pro začátečníky, sestavený z článků od uživatelů. Tato webová stránka do značné míry naplňuje moji představu o přehledném a kvalitně zpracovaném výukovém webu se zaměřením na didaktické testy.

Nalezl jsem však pár věcí, které mi přece jen na tomto webu chybí, je to především datové úložiště. Je velice pravděpodobné, že učitel, který bude přidávat články, je bude také chtít využít při svých hodinách a přijde mi velmi užitečné mu umožnit přiložit k článku nějaké další materiály ke stažení. Této problematice se budu věnovat později v kapitole 6.6.3 Správa souborů. Web [www.spanelstina-zdarma.cz](http://www.spanelstina-zdarma.cz) by jistě posunulo, kdyby umožnil uživatelům vytvářet skupiny, ve kterých by mohli sdílet soubory nebo komunikovat. Závěrem musím určitě poznamenat, že tento web a jeho zpracování mi bylo velkou inspirací.

### **5.1.3 [www.jazyky-online.info/spanelstina](http://www.jazyky-online.info/spanelstina)**

provozovatel:	Vysoké učení technické v Brně
správce projektu:	Ing. Zdeněk Brož, Ph.D.
poslední aktualizace:	2013
partneři projektu:	Online-languages.info Jazyky-online.info WoordenBoek.eu KoDi.cz

Velice dobře zpracovaná stránka, spadající do licence Creativer Commons, pro výuku hned několika světových jazyků.

Projekt je pod záštitou [www.dicts.info](http://www.dicts.info). To je soubor GNU online slovníků, v rámci kterého si můžeme stáhnout příslušnou databázi slovíček zdarma. Obsah



nalezneme opět přehledně rozdělený do kategorií. Namátkou zmíním jen některé příklady: španělská gramatika, obrázkový slovník, nejpoužívanější fráze, aplikace pro mobilní telefony a mnohé další. Kategorie jsou plněny buď konkrétním obsahem od webmastera nebo obsahují pouze odkazy na již existující stránky věnující se stejné problematice. Stránka je velice přehledná a nabízí spoustu materiálů k nastudování. Co ovšem na stránce nenajdeme, jsou didaktické testy a ani možnost se zaregistrovat jako uživatel a spravovat studijní materiály. Je možné, aby po domluvě s provozovatelem stránky uživatel také přispíval svými články. Po emailové konverzaci s panem Brožem jsem zjistil, že pokud má kdokoliv zájem přispívat svými články, je mu to umožněno. Článek připraví v textovém editoru a pošle emailem. Správce pak článek zkontroluje a umístí na web. Přestože je tedy stránka spíše statická a uživatel nemá k dispozici rozhraní, kterým by mohl přímo přidávat a editovat články, obsahuje celkem velké množství učiva a podpůrných materiálů, jako jsou poslechová cvičení nebo videa.

#### **5.1.4 www.estudiante.cz**

provozovatel:	Petr Stříbrný
poslední aktualizace:	2012
software:	Wordpress 3.4.1

Tato stránka sice nepřekypuje výukovými materiály, zato nabízí spoustu užitečných praktických rad pro studenty španělštiny.

Web byl vytvořen roku 2009 a jedná se spíše o studentský blog, který informuje studenty o probíhajících akcích, shromážděních, koncertech atd. Nalezneme zde ale příjemné překvapení v podobě několika výukových aplikací a slovníků pro mobilní telefony. Uživatelé se sice nemohou registrovat, ale je jim umožněno většinu příspěvků okomentovat. Jak jsem již zmínil výše, stránka nenabízí velké množství materiálů a didaktickým testům se nevěnuje téměř vůbec. Její největší přínos vidím v možnosti stažení výukových aplikací.

### **5.1.5 jazyk.okhelp.cz/spanelstina**

provozovatel: Josef Kolísek  
poslední aktualizace: 2013

Na začátek je důležité poznamenat, že stránka [www.okhelp.cz](http://www.okhelp.cz) primárně plní funkci katalogu.

Grafické zpracování je dosti zvláštní a stránka působí docela zmateně a neuspořádaně. Nicméně pod nelibivou doménou se nachází celkem dobře zpracované materiály pro výuku španělštiny. Články jsou doplněny didaktickými testy a student může po absolvování testu shlédnout vyčerpávající statistiku svého výkonu a srovnat svoje výsledky s ostatními studenty. Stránka ale bohužel nenabízí možnost registrace, student si své výsledky může prohlédnout jen po vyplnění testů. Na první pohled je tedy jasné, co mi na webu chybí. Je to možnost uživatelů se na obsahu webu aktivně podílet. To tato stránka bohužel neumožňuje a i když je možné, že po domluvě s provozovatelem bude umožněno přispívat svými články a testy, není to přesně to, co bych od stránky věnující se výuce jazyků očekával.

### **5.1.6 www.spanelstina-on-line.cz**

provozovatel: GDI s.r.o  
poslední aktualizace: 2013

Webová stránka s celkem přívětivou doménou a příjemným grafickým zpracováním je jen velkým rozcestníkem a reklamním serverem.

Stránka nabízí vyčerpávající seznam institucí, které se věnují překladům ze španělštiny do ostatních jazyků samozřejmě včetně češtiny. Vedle katalogu překladatelských institucí zde nalezneme i seznam tlumočnicků či jazykových škol a kurzů. Stránka rovněž slouží k propagaci španělských učebnic a ostatních materiálů, které si návštěvník může zakoupit v e-shopu. Bohužel z hlediska poskytování výukových materiálů a didaktických testů tato stránka není zajímavá.

### 5.1.7 Shrnutí

Z předchozí analýzy vyplývá, že na českém webu si neplacené webové stránky, které se věnují výuce španělštiny, teprve hledají své místo. Avšak projekty [www.spanelstina-zdarma.cz](http://www.spanelstina-zdarma.cz) a [www.jazyky-online.cz](http://www.jazyky-online.cz) v sobě skrývají vysoký potenciál a mezi ostatními analyzovanými weby jasně vyčnívají jak zpracováním tak i množstvím nabízených materiálů. Web [www.spanelstina-zdarma.cz](http://www.spanelstina-zdarma.cz) navíc nabízí opravdu dobré zpracování didaktických testů a sestavený kurz španělštiny pro začátečníky. Do analýzy jsem úmyslně nezahrnul stránky komerčních vzdělávacích institucí. Dále jsem se také nezabýval analýzou interních školících systémů, jelikož se nejedná o weby přístupné veřejnosti. Z prostudovaných webových stránek jsem stanovil závěr. Na českém internetu chybí taková stránka, která by uživatelům umožnila plně se zapojit do tvorby jejího obsahu a poskytla by nejen možnost prezentovat články či příspěvky ve fórech, ale rovněž by umožnila vytvářet didaktické testy s kvalitní zpětnou vazbou, či také sdílet rozšiřující materiály jako videa, obrázky nebo hudbu. Učitelům by tak umožnila získávat statistické informace o svých testech, na jejichž základě by je mohli zdokonalovat. Studentům by poskytla možnost otestovat a doplnit svoje vědomosti.

## 6 Praktická část

Aplikace [www.ensename.cz](http://www.ensename.cz) je naprogramována v jazyce PHP. Aktuální verze PHP na webovém serveru Apache je 5.3.10. Pro ukládání dat je použita relační databáze MySQL verze 5.1.61. Pro finální výstup je použit jazyk XHTML transitional a CSS2. Pro lepší přístupnost a možnost technické podpory je webhosting zřízen u společnosti [pipni.cz](http://pipni.cz) v tarifu PROFI. Webová stránka má podle serveru <http://www.ping.eu> průměrnou odezvu 14.43 ms. Aplikace je programovaná procedurálně a funkčně je rozdělena na několik částí, podrobnému popisu těchto částí se budu věnovat v kapitolách 6.2 Struktura aplikace a 6.3 Struktura databáze.

### 6.1 Použité technologie

Aplikace byla naprogramována pod operačním systémem Ubuntu 11.10 v programu gedit. Ladění proběhlo v Operačním systému Ubuntu 11.10, dále v operačním systému Windows XP SP2 a v operačním systému Windows 7. Nejlepšího a nejrychlejšího zobrazení jsem dosáhl v prohlížeči Chrome a Firefox. O něco pomaleji, ale s dobře zobrazenými CSS prvky, se aplikace načítá v prohlížečích Opera a Safari. V prohlížeči Internet Explorer se stránka načítá s dobrou rychlostí, zato některé prvky CSS se zobrazují podstatně odlišně než u předchozích uvedených prohlížečů.

Přestože se tato práce nezabývá popisem programovacích jazyků, je podle mě vhodné na úvod před vlastním popisem aplikace napsat pár slov o jednotlivých použitých technologiích. Těmito technologiemi jsou HTML, CSS, PHP a MySQL.

### 6.1.1 HTML

Jazyk HTML vznikl v roce 1991, od této doby se jeho užití i možnosti značně rozšířily. V současnosti je jedním z nejpoužívanějších jazyků pro vytváření prezentací na internetu. Hlavní charakteristikou HTML je používání značek, takzvaných tagů, od tohoto pojmenování se užívá název značkový jazyk. V principu jde při užívání jazyka o zápis obsahu mezi dva párové tagy, které prohlížeči sdělí, jaký formát pro zobrazení daného obsahu má zvolit.

Příklad užití HTML tagů pro zobrazení textu a hypertextového odkazu na webové stránce:

```
<p> Buenos días! <br />
<a href="http://www.ensename.cz/">odkaz</a>
</p>
```

Na tomto příkladu vidíme názorně princip zapouzdřování tagů. Je velice důležité kontrolovat správný sled tagů, aby nedošlo k jejich křížení. Webový prohlížeč by poté mohl zdrojový kód přeložit v lepším případě s chybou, v horším vůbec. Použitý příklad vypíše v odstavci text *Buenos Días!*, odřádkuje a vypíše text, který je zároveň hypertextovým odkazem vedoucím na stránku <http://www.ensename.cz>. Vedle tagů párových existují i tagy nepárové, jejichž užití se od předchozího typu liší pouze absencí ukončovacího tagu.

Příklad syntaxe nepárového tagu:

```

```

Nepárový tag `<img>` se používá pro zobrazení obrázku na webové stránce. V našem případě zobrazí obrázek `picture.jpg`. Při použití nepárových tagů musíme dbát na ukončení tagu použitím zpětného lomítka.

Základní struktura HTML dokumentu musí vždy obsahovat informaci o použitém typu HTML a DTD, nebo-li základní informace pro prohlížeče, podle kterých vybírají jazykovou sadu a příslušné HTML elementy.

### 6.1.2 CSS

Zkratka CSS je akronymem anglických slov *Cascading Style Sheets*, česky kaskádové styly. CSS je technologií navrženou za účelem oddělení obsahu webových stránek od jeho grafického zobrazení. Používá se zejména s technologiemi HTML,

XHTML nebo XML. V současnosti existují tři verze, a sice CSS1, CSS2 a CSS3. Ve své podstatě CSS přiřazuje vlastnosti zobrazení HTML elementům. Vlastnosti můžeme přiřazovat buď globálně všem elementům určitého typu a nebo pomocí návěští jen konkrétním elementům. V praxi je poté možno vlastnosti vpisovat buď přímo do HTML kódu k patřičným elementům, což je v rozporu s myšlenkou oddělení kódu od obsahu webu, nebo do oblasti hlavičky dokumentu HTML, kdy sice oddělíme obsah od informací o zobrazení, ale stále máme data v jednom souboru. Nejlepší variantou je tedy použít externí CSS soubor, do kterého informace o zobrazení daných elementů vložíme.

Použití takzvaných inline stylů, nebo-li vložení CSS kódu přímo do zdrojového kódu HTML k patřičnému elementu:

```
<p style="color:red;">ahoj</p>
```

V tomto příkladu přiřadíme elementu `</p>` (odstavci) pomocí inline stylu barvu písma. Text *ahoj* se nám tedy zobrazí červeně. Jak jsem již psal výše, toto užití není vhodné právě proto, že nám neumožňuje oddělit grafické vlastnosti elementu od jeho obsahu.

Další možností použití CSS je jeho vyčlenění do oblasti pod hlavičkou HTML kódu:

```
</head>
<style>
p {color:red;}
</style>
<body>
```

Úmyslně jsem do příkladu zahrnul i ukončovací element hlavičky `</head>` a počáteční element těla dokumentů, `<body>` aby bylo patrné, kde je definice CSS stylu v dokumentu zapsána. Příklad CSS je totožný s předchozím. Poslední a zároveň nejpraktičtější užití CSS je jeho zapsání do externího souboru. Tento soubor bude obsahovat soupis vlastností elementů jako v předchozím případě (bez použití elementů `<style></style>`) a bude uložen s příponou `css`.

Následující dva příklady ilustrují způsob připojení externího CSS souboru do HTML dokumentu:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

```
@import url(style.css) screen;
```

Pomocí těchto příkazů tedy připojíme externí soubor `style.css` k našemu HTML dokumentu. Druhý z příkazů však není dostatečně podporován ve starších verzích prohlížeče Internet Explorer.<sup>41</sup> Užití kaskádových stylů umožní vývojáři širokou škálu možností. Od nastavení zobrazení pro konkrétní zařízení (počítač, mobilní telefon, tiskárna), přes užití vrstev až po nastavení dynamických efektů, jakým je třeba změna barvy textu po najetí myši.

### 6.1.3 PHP

Zkratka PHP původně pochází z anglické fráze Personal Home Page. Technologie byla vytvořena roku 1994 Rasmusem Lerdorfem za účelem sledování návštěvnosti webových stránek. Postupem času se však užitečnost a nárůst možností použití zvětšovaly a prosadil se název PHP: Hypertext Preprocessor. Hlavní myšlenkou PHP scriptovacího jazyka je vykonat určitou reakci na událost, kterou vyvolá uživatel. Důležité je především to, že technologie PHP je nezávislá na platformě a je spuštěná na serveru. Všechny operace se tedy dějí na straně serveru nikoliv na straně klienta. Nezávislost na platformě nám umožňuje pracovat s technologií PHP pod různými operačními systémy.<sup>42</sup>

Příklad klasického použití PHP pro ověření vstupu z HTML formuláře:

```
If (is_numeric($_POST['phone'])) $phone=$_POST['phone'];
```

Příkaz `If` přiřadí hodnotu proměnné `phone` z globálního pole `$_POST` proměnné `$phone` za podmínky, že funkce `is_numeric()` rozhodne, že vložená proměnná je číslem. PHP obsahuje širokou škálu příkazů pro ověření datového typu či obsahu proměnné. Je proto velice vhodným prostředkem pro kontrolu a zadávání dat do databáze.

### 6.1.4 MYSQL

Aplikace původně vyvinutá společností MySQL AB je relační databází používající jazyk SQL. V současnosti jí vlastní Sun Microsystems, dceřinná

---

<sup>41</sup> PEXA, Petr. *Jazyky XHTML CSS DHTML WML*, České Budějovice, 2006, s. 87.

<sup>42</sup> ULLMAN, Larry. *PHP a MySQL*, Brno, 2004, s. 14.

společnost firmy Oracle. Relační databáze je systém tabulek uchovávající číselná nebo textová data či binární soubory. Pomocí příkazů jazyka SQL pak můžeme s daty v databázi pracovat, to znamená vytvářet, mazat a editovat tabulky a do nich pod určitým klíčem vkládat obsah. MySQL umí každé položce z tabulky přiřadit její index a při vložení nové položky přiřadí index o jednu větší než poslední, nebo-li inkrementuje hodnotu indexu položky. Jednotlivým sloupcům v tabulce pak můžeme nastavit datový typ a tím zajistit lepší a efektivnější ukládání dat.<sup>43</sup>

Příklad užití SQL příkazu pro vložení dat do databáze:

```
INSERT INTO users (name, surname, email) VALUES ('Vít',  
'Filipovský', 'witas@volny.cz');
```

Příkaz `INSERT INTO` vloží do sloupců `name`, `surname` a `email`, tabulky s názvem `users` hodnoty `Vít`, `Filipovský` a `witas@volny.cz`.

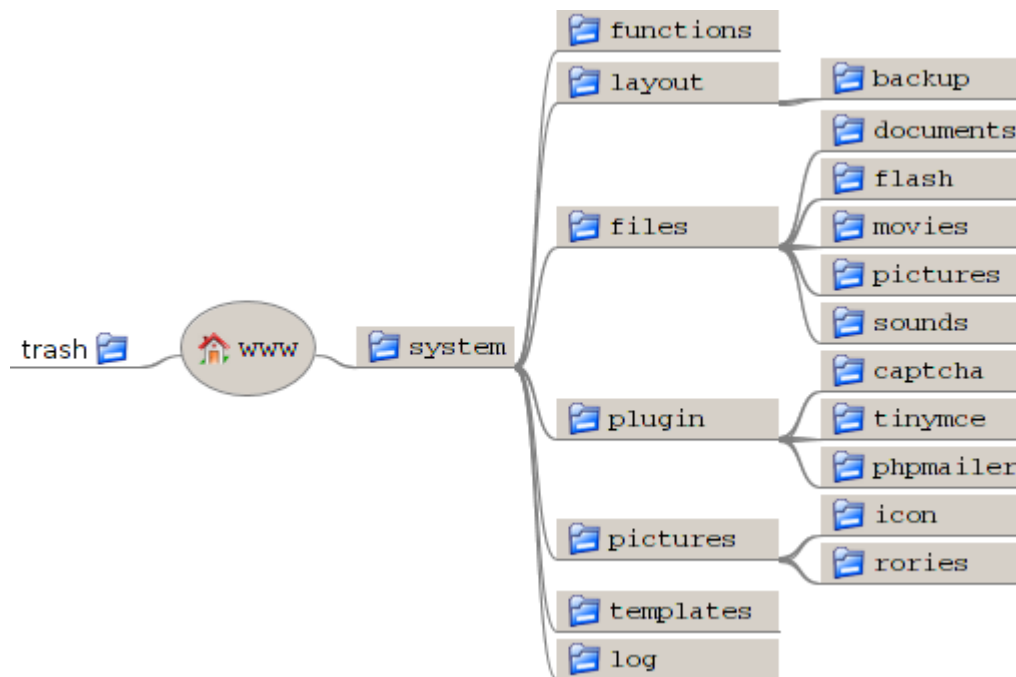
## 6.2 Struktura aplikace

Aplikace `ensename.cz` má předem připravenou strukturu adresářů pro snazší orientaci a také pro snazší nastavení oprávnění pomocí `.htaccess`. V této kapitole se chci pro větší přehlednost věnovat pouze struktuře adresářů. Strukturu nejlépe ilustruje Obrázek č. 2. Jádru programu je umístěno v adresáři systému a v adresáři `functions`. Jádrem jsou myšleny prvky aplikace, na kterých je zbytek přímo závislý, jsou to zejména inicializační soubory a soubory obsahující elementární funkce pro práci s databází.

---

<sup>43</sup> ULLMAN, Larry. *PHP a MySQL*, Brno, 2004, s. 16.





Obrázek č. 2: Adresářová struktura aplikace ensename.cz

### 6.2.1 Adresáře a jejich obsah

Členění aplikace do jednotlivých adresářů je v první řadě důležité pro přehlednost celé aplikace. Struktura je, podle mého názoru, intuitivní a urychluje orientaci ve zdrojovém kódu. Další důvod, který mě vedl k rozčlenění aplikace do složek, je určitě bezpečnost. V takto rozčleněné aplikaci mohu snadněji nastavit oprávnění jednotlivým částem aplikace a tak zamezit nežádoucímu přístupu do složek obsahujících citlivá data, zároveň mohu zachovat přístup do složek, které obsahují data veřejná.

Kořenový adresář `www` obsahuje složky `system` a `trash` a soubory `index.php` a `screen.css`. Složka `trash` je poměrně nepodstatná a obsahuje jen zkušební soubory, které zachovávám pro případ, že bych potřeboval nahlédnout do staršího zdrojového kódu.

Složka `system` je vlastně kořenovým adresářem pro zbytek aplikace. Obsahuje složky `functions`, `layout`, `files`, `plugin`, `pictures`, `templates` a `log`. Dále zahrnuje soubory `db.php`, `functions.php`, `check.php`, `init.php`, `lang.php`, `script.php`, `terminal.php`.

Functions je velice důležitou složkou obsahující soubory se základními funkcemi pro načítání dat z databáze `db_func.php`, pro formátování stránky `format_func.php` a pro uživatelské operace `usr_func.php`.

Složka `layout` obsahuje kromě složky `backup`, ve které jsou zálohovány pracovní verze několika souborů, také soubory s jednotlivými funkcemi, aplikace, jakými jsou například správa účtu, zobrazení menu, vytváření testů a tak dále. Aplikace načítá veškeré soubory PHP, které tato složka obsahuje. Aplikaci lze tedy rozšířit o další funkci jednoduchým nahráním souboru s funkcí do složky a jejím zavedením v souboru `index.php`. Tomuto řešení se budu podrobněji věnovat v kapitole 6.4.1 Inicializační funkce.

Files je složka sloužící pro uchovávání souborů nahraných uživateli. Pro lepší třídění obsahuje podsložky `documents`, `flash`, `movies`, `pictures` a `sounds`, do kterých jsou soubory od uživatelů po nahrání automaticky tříděny.

Složka `plugin` obsahuje dodatečná použitá rozšíření. Podle názvů složek poznáme, o která rozšíření se jedná. Jsou to `captcha` nebo-li rozšíření použité při registraci nového uživatele, dále `tinymce`, což je rozšíření pro klasická textová pole a `phpmailer`, vylepšená verze funkce pro odesílání formulářů mailem. Více o rozšířeních bude uvedeno v kapitole 6.4 Funkce.

Templates je složkou obsahující soubory šablon `meta.php` a `tinymce.php`. Tyto soubory jsou vyčleněny zvlášť kvůli větší přehlednosti aplikace a také kvůli snazšímu nastavení zabezpečení pomocí `.htaccess`. Složka `log` obsahuje pouze soubor `log.txt` a je vytvořena prakticky ze stejného důvodu jako složka předchozí.

## **6.2.2 Rozdělení a funkce souborů**

V předchozí kapitole jsem nastínil hierarchii adresářové struktury a obsah jednotlivých složek. V této kapitole bych rád rozvedl úlohu jednotlivých PHP souborů bez detailních popisů jednotlivých funkcí, které pro přehlednost analyzuji až v kapitole 6.4 Funkce.

### **6.2.2.1 Kořenový adresář `www`**

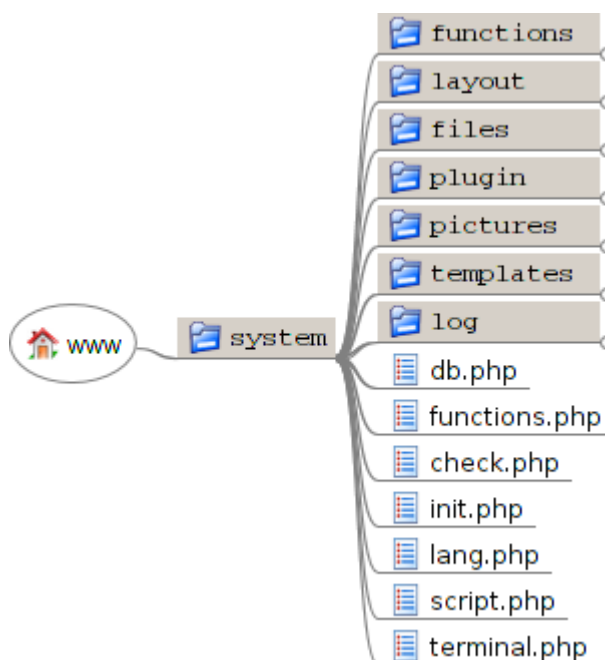
Základní soubor obsahující algoritmy pro inicializaci všech funkcí `index.php` je rozdělen na pět částí. První část se zaměřuje na načtení inicializačních souborů, základních a rozšiřujících funkcí, včetně správy jazykových mutací. Druhá část

kontroluje vstupy uživatele a provádí základní přesměrování na jednotlivé stránky aplikace. Třetí část se stará o vypisování zásobníku upozornění a chybových hlášek. Část čtvrtá obsahuje šablonu pro konečný výstup v podobě HTML kódu, pátá část algoritmus pro ověření přihlášení uživatele.

Dále se v kořenovém adresáři nachází soubor screen.css obsahující informace o grafickém rozložení a formátování celé aplikace.

### 6.2.2.2 Adresář system

Ponoříme-li se o úroveň níže do adresářové struktury, ocitneme se v adresáři system. Zde nalezneme celkem sedm souborů obsahujících bazální algoritmy pro funkci celé aplikace. Strukturu ilustruji na Obrázku č.3.



Obrázek č. 3: Obsah adresáře systém

První ze souborů db.php se stará o připojení databáze. Připojí inicializační soubor a na základě proměnných v něm uložených načte příslušnou databázi. Funkce je tvořena jen několika řádky, proto by mohla být zajisté umístěna do souboru k ostatním funkcím. Její správná funkčnost je však pro celý systém natolik zásadní, že jsem jí raději vyčlenil do samostatného souboru, který je rovněž i samostatně načítán.

Soubor functions.php obsahuje především příkaz pro připojení souborů ve složce

functions viz složka functions v kapitole 6.2.1 Adresáře a jejich obsah. Mimo to obsahuje ještě dodatečné funkce, které nebylo možno rozčlenit na databázové, uživatelské a formátovací, například funkce logování, funkce pro změnu jazykového nastavení, funkce pro generování náhodného kódu při zapomenutí hesla a tak dále.

Další soubor v adresáři se jmenuje příznačně check.php. Je to soubor obsahující kontrolní funkce pro dobrý chod celého systému. Nalezneme zde například funkce pro kontrolu globální proměnné `$_POST` na přítomnost nepovolených znaků, funkci pro kontrolu správně zadané hodnoty formulářového pole nebo funkci pro kontrolu relace.

Soubor init.php je základním inicializačním prvkem aplikace. Obsahuje hodnoty, které používá soubor db.php pro načtení databáze a přepínače pro základní nastavení aplikace. Pro lepší ilustraci uvádím část zdrojového kódu.

```
$_SESSION['init']=array(  
/*nastavení hlavičky*/  
'webname'=>"Ensename.cz",  
'description'=>"stránky pro výuku španělštiny, página para enseñar  
español, page for learning spanish",  
'keywords'=>"španělština, español, spanish, zdarma, libre, free,  
achievement test, online testy",  
'bottom'=>"Programed by witasfilipovsky@gmail.com",  
/*nastavení webu*/  
'log'=>"off", //zakázat funkci log  
'im3'=>"on", //povolit funkci terminál pro ladění aplikace  
'alert'=>"on", //povolit funkci upozorňování  
'uploadscript'=>"off", //zakázat uploadování scriptových souborů  
'javascript'=>"on", //zapnout javascript  
);
```

Soubor init.php vlastně obsahuje pouze dvě pole s uloženými proměnnými. První pole zde uvedeno není a obsahuje proměnné pro načtení databáze. Uvedená část zdrojového kódu nastavuje stránce title neboli její název, description česky popis stránky a keywords, což jsou klíčová slova pro vyhledávače. Dále si můžeme všimnout přepínačů pro zapnutí funkcí logování, ladění, zobrazování upozornění, javascriptu a povolení uploadovat scriptové soubory do uživatelských adresářů.

Základní slovník aplikace je umístěn v souboru lang.php. Strukturu uložených dat opět ilustruji příkladem zdrojového kódu.

```
$_SESSION['titles']=array(
'LOGIN' => array('Login','Usuario','Login'),
'ACCOUNT' => array('Účet','Cuenta','Account'),
'PASS' => array('Heslo','Contraseña','Password'),
```

Struktura pole je vytvořena tak, abychom se pomocí statické proměnné a indexu dostali pro požadovaný text určité jazykové mutace. Index jazykové mutace nastaví uživatel přepnutím a podle tohoto indexu se vybírá příslušná proměnná z pole.

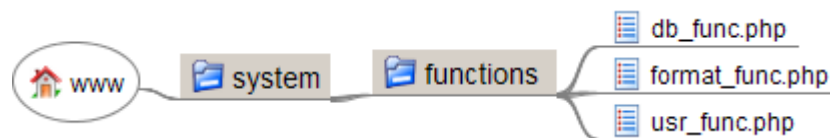
Soubor lang.php obsahuje celkem čtyři vícerozměrná pole. V jednom poli jsou uloženy všechny chybové hlášky. V poli druhém jsou uloženy všechny krátké texty a popisky formulářů či jejich tlačítek. Třetí pole obsahuje všechny dlouhé texty jako je například znění emailu pro obnovení hesla. Poslední pole obsahuje texty, které se zobrazují v javascriptových oknech, kontrolujících obsah formulářů.

Dalším souborem je script.php. Tento soubor obsahuje javascript, který kontroluje správnost vstupů v textových polích. Správnost vstupů je kontrolována především pomocí algoritmů v PHP. Nicméně dvojí kontrola formulářových polí je bezesporu dobrým prostředkem pro zajištění lepší kontroly vkládaných dat. Kontrola pomocí javascriptu navíc probíhá na straně počítače a ne serveru. Pokud tedy uživatel políčko formuláře vůbec nevyplní, nebude požadavek na server odeslán.

Posledním souborem v hlavním adresáři je terminal.php. Tento soubor obsahuje množství ladících mechanismů pro snazší tvorbu webu. Tyto mechanismy se používají pomocí globální proměnné `$_GET`, do které se data ukládají pomocí adresního řádku. Soubor terminal.php tedy kontroluje obsah adresního řádku a jakmile je v něm přítomna proměnná `im3` obsahující hodnotu, vykoná patřičnou funkci. Mezi tyto funkce patří například odlogování uživatele, výpis globálního pole `$_SESSION`, zobrazení uživatelského jména a další.

### 6.2.2.3 Složka functions

Hlouběji v adresářové struktuře se nachází složka functions, která obsahuje tři soubory se základními funkcemi aplikace. Jsou to `db_func.php`, `format_func.php`, `usr_func.php`.



Obrázek č. 4: Obsah adresáře functions

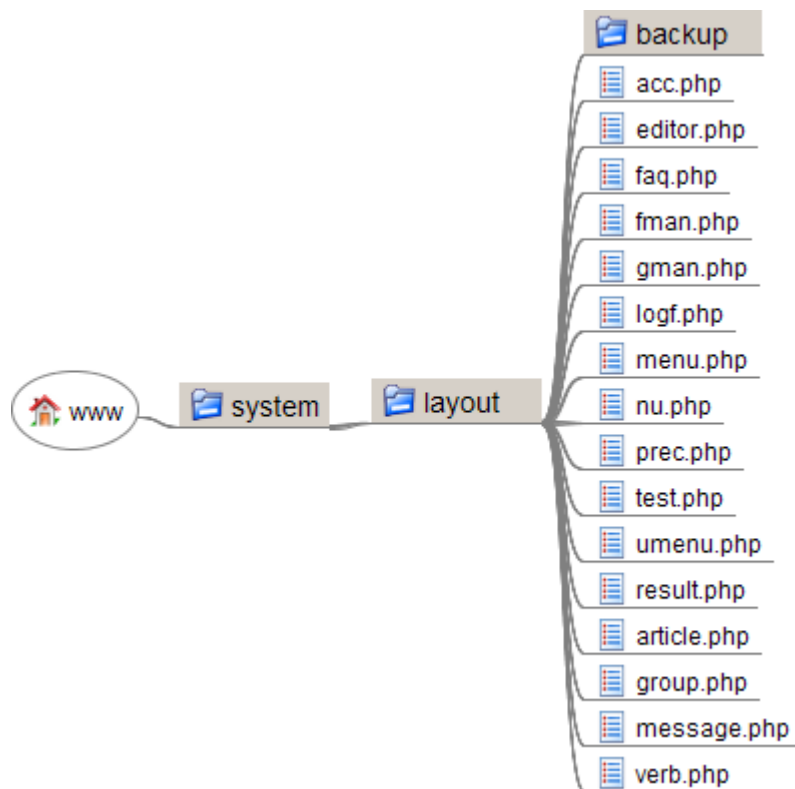
Soubor `db_func.php` tvoří základní funkce pro práci s databází, jako jsou načtení položky z databáze, smazání položky nebo update položky v databázi. Mimo to obsahuje ještě konkrétní databázové funkce, které se v aplikaci často používají.

Další soubor v této složce nese název `format_func.php` a je tvořen funkcemi, které zajišťují formátování obsahu webu. Významnou část aplikace tvoří funkce pro vytvoření formuláře. Jelikož je celý web víceméně založen na práci s formuláři, je zde připravena zásoba funkcí pro tvoření veškerých typů formulářových prvků. Dále existuje funkce zajišťující převod pole na tabulku či umožňující sestavit hypertextový odkaz.

Třetí a poslední soubor ve složce obsahující uživatelské funkce se jmenuje `usr_func.php`. Tento soubor obsahuje veškeré funkce spravující uživatelskou činnost. Mezi hlavní patří zejména funkce pro vytvoření uživatele, změny jeho údajů, přihlášení, odhlášení, zjištění existence uživatele a další.

#### 6.2.2.4 Adresář layout

Na stejné adresářové úrovni jako složka `functions` leží složka `layout`. Tato složka obsahuje soubory s funkcemi zajišťujícími uživatelské vstupy a zobrazení stránky.



Obrázek č. 5: Obsah adresáře layout

Soubor `acc.php` obsahuje znění stránky sloužící ke správě uživatelského účtu. Zdrojový kód, který soubor obsahuje, můžeme rozdělit na dvě části. První část je tvořena algoritmy, které kontrolují, zda byl přijat vstup od uživatele a následně ho vyhodnotí. Část druhá vytvoří HTML kód, který se v závislosti na uživatelském vstupu vypíše.

Soubor `editor.php` obsahuje rozhraní pro tvorbu didaktických testů. Zdrojový kód se opět skládá z několika částí, jejichž funkcionality jsou na sobě navzájem závislé. Editor totiž umožňuje dynamicky měnit strukturu editovaných položek. Různý počet zobrazovaných prvků stránky je tedy přímo závislý na uživatelském vstupu. V praxi to znamená, že uživatel nastaví vlastnosti celého testu a podle nich si dále generuje předem neurčený počet formulářů. Uživatelem generovaná data se ukládají do globální proměnné `$_SESSION`, kde se aktualizují a poté se uloží do databáze.

Nápověda k ovládní programu a vysvětlení případných nejasností je obsahem souboru `faq.php`. Respektive soubor `faq.php` obsahuje jednoduchý algoritmus pro načítání textu z databáze a ten zde vypisuje.

Soubor `fman.php` je tvořen zdrojovým kódem, který se stará o správu souborů.

Tvoří ho několik částí. Hlavní část se stará o upload souboru na server a o zapsání parametrů nahrávaného souboru do databáze. Kontroluje rovněž, zda nejsou nahrávány na server soubory, které by mohly narušit chod aplikace. V praxi tato funkce způsobí zablokování uploadu souborů s konkrétními koncovkami, jako jsou například php, js, html, css a další. Fman.php obsahuje rovněž algoritmy pro nastavení práv přístupu k souborům ostatním uživatelům a skupinám či algoritmy pro mazání souborů ze serveru.

Důležitá součást, avšak pro běžné uživatele nedostupná, je kód souboru gman.php. Zde jsou uloženy algoritmy pro nastavování vztahů mezi konkrétními položkami v databázi. Administrátor tak může z jednoho místa vytvářet nové skupiny, ale i nastavovat oprávnění uživatelům nebo skupinám, popřípadě přiřazovat uživatele do pracovních skupin.

Soubor logf.php je tvořen zdrojovým kódem formuláře pro přihlašování uživatelů. Je rozdělen na část, která vyhodnocuje vstupní data uživatele a na část, která je zobrazuje na obrazovku.

Velice důležitou součástí je rovněž soubor menu.php, který ovládá zobrazení levého uživatelského menu. V tomto souboru jsou uloženy algoritmy pro načítání a zařazování vytvořených testů a článků do určitých kategorií.

Nového uživatele lze vytvořit pomocí funkce umístěné v souboru nu.php. Jako již v předchozích případech, i zde je kód rozdělen na dvě části. Část, která zpracovává data od uživatele a část, která generuje HTML kód. Dále je zde ještě část obsahující kód pluginu CAPTCHA pro kontrolu, zda registrovaný je skutečně člověk nebo se jedná o internetového robota. Při zadání správných údajů jsou data odeslána do funkce umístěné v souboru usr\_func.php, kde jsou zpracována a je vytvořen nový uživatel.

Pro vygenerování a zaslání nového hesla se používají funkce umístěné v souboru prec.php. Tato stránka vypíše jednoduché textové pole, do kterého uživatel vloží svoje uživatelské jméno. Funkce zkontroluje, jestli uživatel existuje a pokud ano, je vygenerováno nové heslo, které se automaticky zašle na jeho emailovou adresu.

Soubor test.php, respektive zdrojový kód nacházející se v tomto souboru zajišťuje všechny funkce týkající se vypisování a kontroly vyplněných testů. Pokud si tedy uživatel přeje vyplnit didaktický test, je to právě soubor test.php, který je zvolen pro



práci s databází. Údaje o testu jsou nejprve načteny do globálního pole `$_SESSION` a poté je z nich vygenerován test. Po vyplnění testu se odpovědi uloží rovněž do globálního pole a funkce pro porovnávání výsledků porovná odpovědi uživatele se správnými odpověďmi načtenými z databáze. Následuje vyhodnocení a zapsání výsledků uživatele do databáze.

`Umenu.php` obsahuje jednoduchý algoritmus pro vypisování sekundárního menu nebo-li menu, které se zobrazí při otevření uživatelského účtu. Podle skupiny, do které uživatel patří, jsou vybrány položky, jenž bude menu obsahovat a ty se následně zobrazí.

Veškeré statistické informace jsou zobrazovány pomocí funkcí ze souboru `result.php`. Soubor je rozdělen na část, která vypisuje informace pro učitele a část, která vypisuje informace pro studenty. Podle uživatelské skupiny je rozhodnuto, která část je spuštěna. Učitelům se zobrazí statistické údaje, jako jsou například počet vyplnění testu, počet správných nebo špatných odpovědí či údaje jako reliabilita. Studentům se zobrazí jejich osobní výsledky, hodnocení a průměry ze všech testů.

O editaci a vypisování článků se stará soubor `article.php`. Zdrojový kód je rozdělen na tři části, přičemž jedna se opět stará pouze o výpis stránky. Zbylé dvě části se starají o editaci starých a tvorbu nových článků a o vyhodnocování uživatelských vstupů. Již editor pro tvorbu článků je sestaven tak, aby uživatele vedl k logickému rozčlenění článku.

Jelikož článek je vlastně distančním textem, je rozhraní pro vkládání uspořádáno tak, aby článek při vyplnění všech polí splňoval základní kritéria pro strukturu tohoto typu textu. Správně vytvořený článek potom bude mít část úvodní, motivační a aktivizační, expoziční a fixační.<sup>44</sup> Je jasné, že ne každý článek bude kvalitním distančním textem, ale to závisí do veliké míry na autorovi. Administrátor aplikace pak má možnost naprosto nevhodné články z databáze odstranit a autora těchto příspěvků upozornit na jejich neadekvátnost.

Všechny materiály, které autor vytvoří, je možné propojit odkazy pomocí algoritmů uložených v souboru `group.php`. Každý test, článek i soubor je opatřen vlastním id, proto můžeme pomocí jednoduché relační tabulky nastavit vzájemný vztah těchto objektů. Uživatel použije jednoduchého rozhraní, které tvoří tři select

---

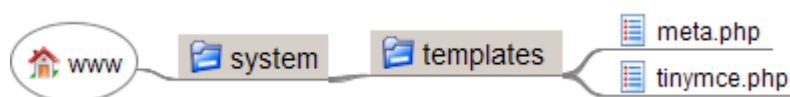
<sup>44</sup> KOPECKÝ, Kamil. *E-learning (nejen) pro pedagogy*, Olomouc, 2006, s. 45.

boxy. V jednom zvolí typ vazby test -> článek, test -> soubor nebo článek-> soubor a při zobrazení testu nebo článku jsou studentovi automaticky nabídnuty odkazy na tyto další související materiály.

Uživatelé si také mohou zasílat jednoduché zprávy pomocí funkce uložené v souboru message.php. Výstup stránky tvoří pouze seznam doručených zpráv a formulář pro zaslání zprávy nové. Zpráva může být zaslána libovolnému uživateli, který ji může zobrazit nebo smazat.

O databázi španělských slovesných tvarů se stará funkce ze souboru verbs.php. Jedná se opět o rozhraní, které umožňuje výpis a editaci příslušných položek v databázi. Oprávnění editovat slovesa mají pouze učitelé, prohlížet je mohou všichni uživatelé.

#### 6.2.2.5 Složka templates



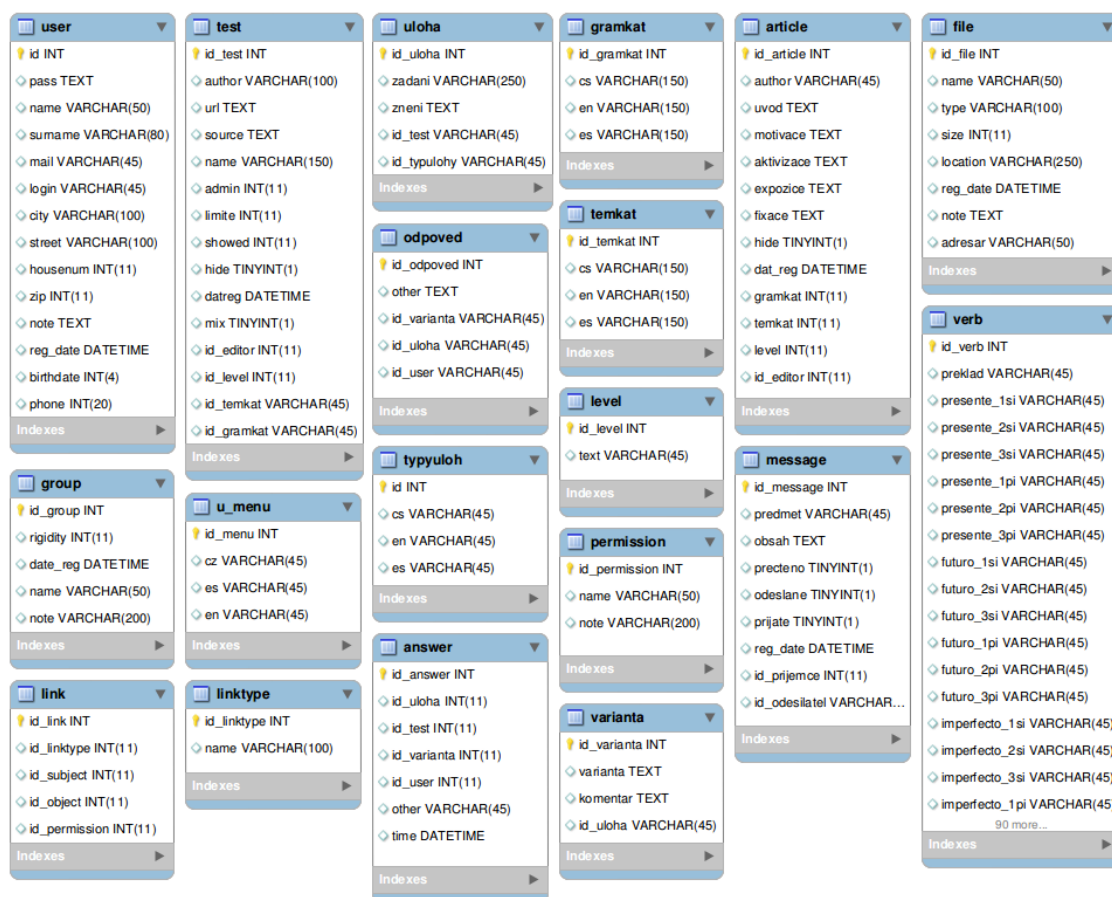
Obrázek č. 6: Obsah složky templates

Šablona meta.php obsahuje text hlavičky HTML dokumentu, ve kterém jsou příslušné údaje nahrazeny proměnnými. Tyto proměnné jsou definovány v souboru init.php. Tato hlavička je tedy jednotná pro všechny zobrazené stránky. Z podstaty programu je tedy možné dynamicky měnit vlastnosti hlavičky jako je název stránky (title), její popis,(description) nebo klíčová slova (keywords). Zároveň je možno zapínat a vypínat používání javascriptu.

Soubor tinymce.php je šablona pro plugin TinyMCE, která pomocí javascriptu rozšiřuje možnosti klasického textového pole a přidává do něj ovládací prvky podobné klasickým textovým editorům. V této šabloně je definováno, jaké ovládací prvky se zobrazí.

## 6.3 Struktura databáze

Databáze je zatím složena z osmnácti tabulek. Píši zatím, protože v budoucnu se počet tabulek může v závislosti na rozšiřování aplikace měnit.

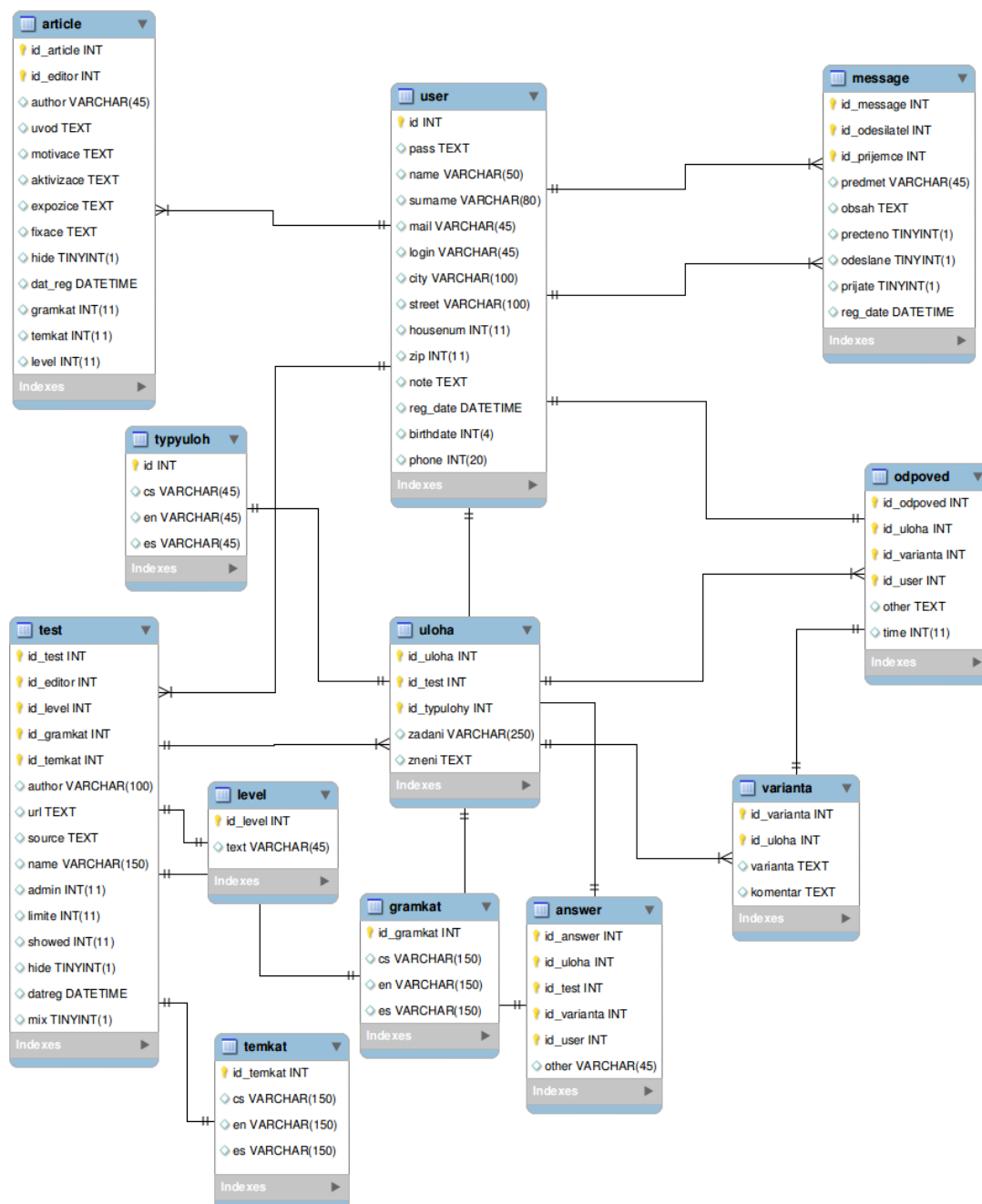


Obrázek č. 7: Přehled všech tabulek v databázi

Základní tabulkou je tabulka user, která uchovává informace o uživatelích. V tabulce file jsou uchovávány informace o nahraných souborech. Informace o odeslaných zprávách uchovává tabulka message. Nejrozměrnější tabulkou je tabulka s názvem verb, který obsahuje gramatické tvary španělských sloves. Články uveřejňované na webu jsou uloženy v tabulce article.

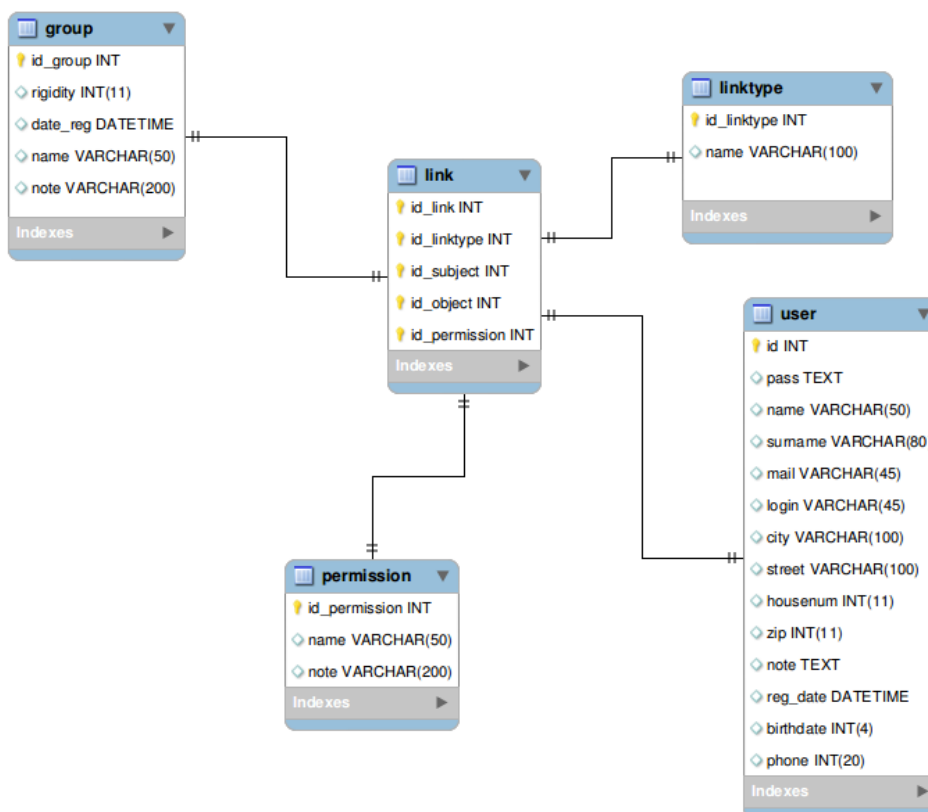
Uchovávání dat didaktických testů má složitější strukturu. Základní tabulkou je Test. Ta obsahuje bazální definice a nastavení testu. K této tabulce je připojena tabulka s názvem Uloha vztahem 1:n. Každá úloha náleží jen jednomu testu. V tabulce Uloha je uložené zadání a znění úlohy. K této tabulce jsou dále připojeny vztahem 1:1 tabulky Gramkat, Temkat, Typyuloh a Level. Tyto tabulky obsahují

názvy kategorií, podle kterých probíhá třídění daného testu. Mimo tyto je k tabulce Uloha připojena tabulka Varianty vztahem 1:n. Každá úloha může mít libovolný počet variant odpovědi. Ke každé variantě existuje právě jedna správná odpověď, mezi tabulkou Varianta a Odpověď je tedy vztah 1:1. Tabulka Odpovědi je rovněž připojena k tabulce Uloha, abychom mohli zobrazovat odpovědi k úlohám bez nutnosti načítat data z tabulky Varianty. Poslední tabulka, která souvisí s tabulkou Testy, je tabulka Answer, která obsahuje odpovědi uživatelů na jednotlivé otázky.



Obrázek č. 8: Struktura vazebních tabulek potřebných ke správě testů

Vedle tabulek sloužících pro uchovávání dat testů a článků jsou zde také tabulky, které se starají o nastavení sdílení a oprávnění uživatelů a skupin. Zde figuruje tabulka Linktype, která obsahuje id a názvy všech možných kombinací vztahů mezi ostatními zvolenými tabulkami. Dále zde máme tabulku Group, která obsahuje id a názvy pracovních skupin a tabulku Permission, která obsahuje id a názvy typů oprávnění. Všechny tyto tabulky spojuje tabulka Link. V této tabulce nastavíme vztahem 1:1 id typu vazby z tabulky Linktype. Dále zde opět vztahem 1:1 nastavíme ve sloupci id\_object id prvku z tabulky nadřazené a do sloupce id\_subject nastavíme identifikátor prvku z tabulky podřazené. Nakonec do sloupce id\_permission vložíme id oprávnění. Tímto způsobem můžeme v jedné relační tabulce uchovávat údaje o podřazenosti různých prvků. Například uživatel je podřazen skupině, soubor je podřazen skupině nebo uživateli. Tabulka Link nám tedy udržuje informace o typu vztahu mezi jednotlivými objekty.



Obrázek č. 9: Přiřazení uživatele ke skupině a nastavení oprávnění

V poslední řadě zde máme tabulku uchovávající informace o slovesných tvarech s názvem `Verb` a tabulku `U_menu`, ve které jsou uloženy jednotlivé prvky uživatelského menu.

## 6.4 Funkce

Celá aplikace využívá zejména parametrických funkcí. Ať funkce pro práci s databází nebo funkce pro formátování stránky, vždy se skládá z hlavního algoritmu a parametrem jsou dodány potřebné proměnné. Funkce pak buď vrátí výslednou proměnnou nebo sama vykoná volání funkce další.

### 6.4.1 Inicializační

Tato funkce připojuje veškeré soubory nacházející se ve složce `layout`. Aplikace ji obsahuje proto, aby umožnila snazší implementaci rozšiřujících funkcí jako je například posílání zpráv nebo nahrávání souborů. Při potřebě dalšího rozšíření pak stačí vytvořit soubor typu PHP obsahující funkci a nahrát ho do složky `layout`. Aplikace soubor sama připojí a funkce může být zavolána.

- Nejprve uložíme do globální proměnné přesnou adresu k souboru `index.php`.

```
$_SESSION['directory']= getcwd();
```

- Poté načteme do pole všechny soubory z adresáře `layout`.

```
function layout(){
    $layout = scandir($_SESSION['directory']."/system/layout/");
    foreach ($layout as $l){
        $u=explode(".", $l); //rozděl na jméno a příponu
        if ($u[1]=="php"){ //pokud přípona souboru je php
            $result[]=$l;}}
    return $result; } //vrátí pole s názvy všech php souborů
```

- Nyní připojíme všechny soubory pomocí funkce `require`.

```
$layout=layout();
foreach ($layout as $l){
require ($_SESSION['directory']."/system/layout/".$l);
}
```

Všechny soubory s příponou php uložené v adresáři layout jsou nyní připojeny a funkce, které obsahují, jsou k dispozici. Volání může probíhat pomocí přepínače s využitím globální proměnné `$_GET['page']`. V následujícím příkladu načítáme do proměnné `$content` výstup funkce `n_u()`, která je uložena v souboru `nu.php`.

```
if ($_GET['page']=='new-user') $content=n_u();
```

Data do proměnné `$_GET['page']` vložíme pomocí adresního řádku. Adresa pak bude mít následující tvar: `http://www.ensename.cz/index.php?page=new-user`. Tímto způsobem je řešeno veškeré přesměrovávání mezi jednotlivými stránkami.

## 6.4.2 Databázové

Pro databázové operace je rovněž připraveno několik funkcí, které se jako v předchozím případě skládají z vlastního algoritmu pro operaci s databází a z parametrů, které obsahují název tabulky, sloupce či klauzule s podmínkami. Funkce pro práci z databází jsou celkem čtyři. `Db_insert_data()` pro vložení nové položky do databáze, `db_update_data()` pro změnu existujících hodnot v databázi, `db_load_data()` pro načtení dat z databáze a `db_delete_data()` pro smazání dat. Podrobněji zde rozeberu pouze funkci pro vložení dat do databáze, jelikož algoritmy použité ve funkcích ostatních jsou velice podobné a liší se zejména jen v použití jiného SQL příkazu.

```
function db_insert_data($table, $value, $column){
    $query= "INSERT INTO `".$table."` ( ".$column.")
    VALUES ( ".$value.")";
    $result = @mysql_query($query);
    query_check($result, $query);
    $return=array($query,mysql_insert_id());
return $return;
}
```

Data jsou do funkce vložena pomocí parametrů `$table`, `$value` a `$column`. Parametry `$table` a `$column` jsou vždy typu string, Parametr `$value` může být typu string nebo integer. Po vykonání dotazu je provedení zkontrolováno funkcí `query_check()`, která v případě nevykonání ohlásí chybu a vrátí znění. Přestože funkce data vkládá, má i návratovou hodnotu, kterou je id vložené položky.

### 6.4.3 Uživatelské

Uživatelských funkcí je několik a zajišťují nebo kontrolují základní uživatelské vstupy. Nejdůležitějšími jsou funkce pro vytvoření nového uživatele `new_user()`, funkce pro zalogování `login()` a funkce pro odlogování `logout()`. Dále se zde nachází funkce pro zaslání emailu `send_mail()`, funkce pro vytvoření pracovní skupiny `create_group()`, funkce pro zjištění pracovní skupiny `load_group()` a funkce pro kontrolu existence uživatele `login_exist()`.

Funkce `new_user()` má za úkol zkontrolovat, jestli jsou vyplněna povinná formulářová pole a zda obsahují hodnotu ve správném tvaru. U emailu je kontrolováno, jestli zadaná hodnota obsahuje znak zavináče. Rovněž jsou porovnána obě zadaná hesla. U číselných hodnot jako je poštovní směrovací číslo nebo telefon je pomocí funkce `is_numeric()` kontrolováno, zda je zadaná hodnota skutečně číslem. Po zkontrolování formuláře jsou data vložena do databáze a uživatel je přiřazen do primární pracovní skupiny, kterou je buď skupina učitelé nebo skupina žáci. Do souboru `log.txt` je přidán záznam o provedené registraci a stránka je přesměrována.

Přihlášení uživatele pomocí funkce `login()` probíhá podobně jako jeho registrace. Funkce nejprve zkontroluje, jestli formulářová pole obsahují povolené znaky a poté zjistí, zda zadané uživatelské jméno existuje v databázi. Pokud uživatel existuje, jsou porovnány hashe obou hesel. Při shodě funkce uloží jméno uživatele do globální proměnné `$_SESSION['login']` a stránka se přesměruje.

Funkce `logout()` má za úkol ukončit `session` a vynulovat všechny globální proměnné. Pomocí příkazu `unset` je vymazána globální proměnná `$_SESSION['login']` a příkazů `session_unset()` a `session_destroy()` je ukončena celá relace. Do souboru `log.txt` je zapsána informace o odhlášení.

Pro zasílání emailu je implementován `phpmailer` verze 2.0.4 a funkce `send_mail()` pouze nastaví proměnné, které `phpmailer` vyžaduje. `Send_mail()` je tedy parametrická funkce, do které se pomocí parametru zadají hodnoty pro odesílatele, příjemce, předmět a znění emailové zprávy. Funkce `send_mail()` pak pouze předá tyto proměnné. Pro `phpmailer` jsem se rozhodl, protože jeho použití je vázáno licenci GNU a jeho spolehlivost a možnosti daleko přesahují použití klasického PHP příkazu `mail()`.

Funkce `create_group()` umožňuje vytvořit novou pracovní skupinu, do které se



mohou přidávat uživatele. Členové skupiny pak mezi sebou mohou snáze sdílet soubory a komunikovat. Tato funkce je vlastně nadstavbou funkce `db_insert_data()` pro vkládání dat do databáze. Již však obsahuje přednastavené hodnoty a jedním povinným parametrem předáváme pouze název skupiny k vytvoření.

Abychom zjistili všechny skupiny, do kterých uživatel patří, použijeme funkci `load_group()`. Tato funkce je nadstavbou funkce `db_load_data()`, kdy parametrem předáme název uživatele a funkce z databáze zjistí, kterých skupin je uživatel členem a vrátí nám seznam jejich id.

`Login_exist()` je jednoduchá funkce, která vrací hodnoty `TRUE` nebo `FALSE`. Pomocí parametru funkcí předáme přihlašovací jméno uživatele a funkce zjistí, jestli tento uživatel existuje v databázi a podle toho nám vrátí hodnotu. Tato funkce je volána především během registrace nového uživatele.

## 6.4.4 Formátovací

Generování výsledného HTML je řešeno rovněž pomocí parametrických funkcí. Volaná funkce použije zadané parametry k sestavení výsledného HTML kódu a tento kód vrátí v podobě proměnné typu `string`. Formátovací funkce bychom mohli rozdělit na funkce jednoduché a funkce složité. I když je toto rozdělení trochu zavádějící, pomůže nám lépe se orientovat. Funkce jednoduché buď neobsahují mnoho parametrů a zpravidla jen použijí proměnnou s obsahem a obklopí jí formátovacími tagy nebo jsou to funkce, které jsou součástí funkcí složitějších. Mezi tyto funkce patří `href()`, `form()`, `form_textarea()`, `select_check()`, `rows()` a `table()`. Funkce složité obsahují buď parametr, který rozhoduje o výsledné podobě HTML kódu a nebo obsahují složitější algoritmus. Mezi tyto funkce patří `form_input()`, `insert_div()`, `array_to_table()` a `form_select()`.

### 6.4.4.1 Funkce jednoduché

Hlavní princip fungování funkcí ilustruji na zdrojovém kódu funkce `href()`:

```
function href($href, $title, $text, $more=NULL) {
    $result = '<a href="'. $href. '" title="'. $title. '"
    ' . $more. '>' . $text. '</a>';
    return $result;}

```

Základním kamenem jsou parametry obsahující hodnoty, které bude funkce používat ke generování výsledného HTML kódu. V příkladu vidíme základ kódu `<a href="" title=""></a>`, do kterého tyto parametry dosadíme a získáme tím výsledný kód. Parametr `$href` nese hodnotu hypertextového odkazu. Parametr `$title` obsahuje popisek odkazu. Další parametr `$text` je tvořen textem, který bude zobrazen a parametr `$more` umožňuje vložit rozšiřující obsah, jako je třeba javascript nebo třída pro použití kaskádových stylů.

Funkce `form()` pracuje s parametry `$data`, `$action`, `$method`, `$name`, `$class` a `$more`. Parametr `$data` je tvořen obsahem formuláře, který je předem vygenerován. Další parametr `$action` obsahuje název souboru, do kterého budou data formuláře odeslána. Parametr `$method` může nabývat hodnoty `post` nebo `get` podle toho, jakou metodu chceme zvolit. Jméno nebo-li atribut `name` a zároveň atribut `id` definujeme pomocí parametru `$name`, kdy se oba atributy nastaví na stejnou hodnotu. Poslední dva parametry jsou nepovinné a to `$class` a `$more`. Kde hodnota parametru `$class` nastaví název třídy pro kaskádové styly a parametr `$more`, použijeme k připojení dalšího kódu jako v případě funkce `href()`.

Funkce `form_textarea()` má pouze tři parametry. Jsou jimi `$name`, `$data` a nepovinný parametr `$class`. Tyto parametry plní stejné funkce jako v předchozím případě. Jméno čili `name` atribut formuláře a zároveň atribut `id` je nastaven na hodnotu parametru `$name`. Druhý parametr s názvem `$data` tvoří textový obsah formuláře a parametr `$class` opět nastavuje třídu pro kaskádové styly.

Funkce `rows()` a `table()` slouží jako pomocné pro funkci `array_to_table()`. Můžeme je používat i odděleně, kdy funkce `rows()` obsahují parametry `$data`, `$class` a `$type`. Tyto funkce slouží k obklopení textu z parametru `$data` tagy a pro tvoření sloupců a řádků tabulky. Funkce podle obsahu parametru `$type` použije buď dvojici tagů `<td></td>` nebo `<tr></tr>` a vytvoří HTML kód, ve kterém bude mezi tyto tagy umístěn text z parametru `$data`. Nepovinným parametrem `$class` můžeme opět nastavit název kaskádového stylu. Funkce `table()` má podobnou úlohu jako `rows()` a to takovou, že obklopí text z parametru `$data` tagy `<table></table>` a vrátí výsledný kód. Opět je zde možnost nastavení kaskádového stylu parametrem `$class` a vepsání rozšiřujícího kódu parametrem `$more`.

Poslední z „jednoduchých“ funkcí je funkce `select_check()`, což je pomocná

funkce, která umožňuje při vkládání formuláře typu select vybrat výchozí hodnotu ze seznamu možností. Tato funkce má dva parametry. Jedním je pole obsahující všechny položky seznamu a jejich indexy a druhým je index položky, která má být opatřena atributem `selected="selected"`. Výstup funkce tvoří vícerozměrné pole obsahující indexy, hodnoty položek seznamu, případně název třídy pro kaskádové styly a u jedné položky navíc informaci o tom, jestli byla vybrána. Pro lepší ilustraci uvádím tabulku, která symbolizuje vstup a výstup funkce. Předpokládejme, že chceme vybrat hodnotu zelená s indexem 2.

Dvě tříprvková pole jako parametr funkce.			Dvě čtyřprvková pole jako výstup funkce.			
index	Hodnota	class	index	hodnota	class	přidaný sloupec
1	Červená	NULL	1	červená	NULL	
2	Zelená	NULL	2	zelená	NULL	selected="selected"

Tabulka č. 2: Porovnání vstupního parametru s výstupem funkce `select_check()`

#### 6.4.4.2 Funkce složité

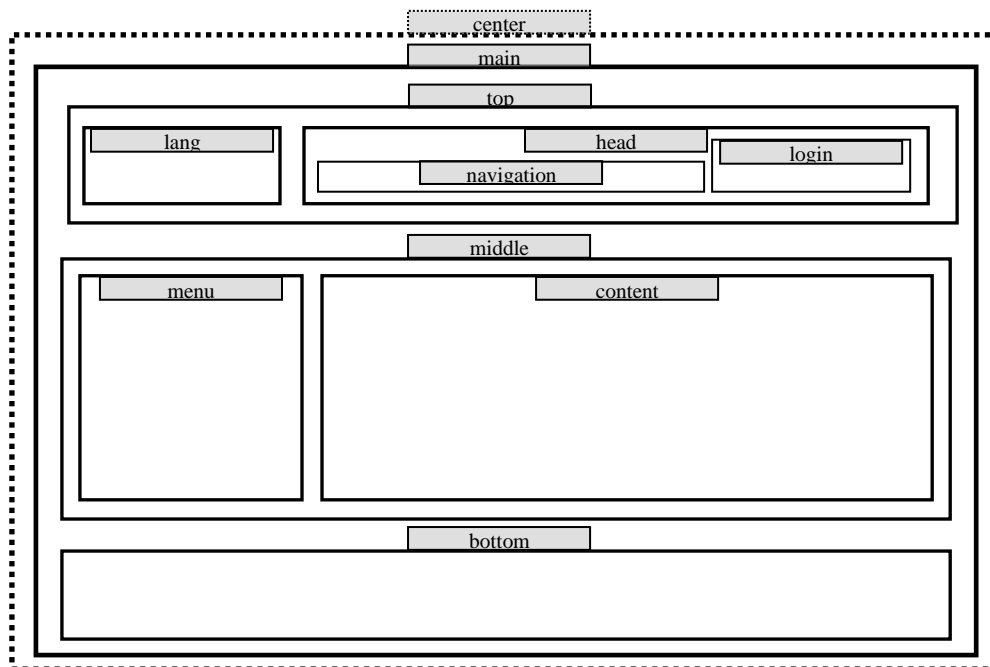
Jednou z hlavních formátovacích funkcí celé aplikace je funkce `insert_div()`. Tato funkce je v podstatě skoro totožná s funkcí `rows()`. Počet povinných i nepovinných parametrů je stejný a to dva. Povinné parametry jsou `$name`, pro nastavení atributů `name` a `id` a parametr `$data` nesoucí obsah, který bude obklopen příslušnými HTML tagy. Pokud bychom nastavili hodnoty jen těchto dvou povinných parametrů, funkce obklopí obsah parametru `$data` tagy `<div></div>` a nastaví atribut `id` a `name` roven hodnotě parametru `$name`. Můžeme ale využít nepovinných parametrů ke změně obklopujícího tagu. Funkce obsahuje přepínač, který může změnit typ tagu na `<span>` nebo `<p>`. Druhý nepovinný parametr nám umožní nastavit název třídy kaskádového stylu. Funkce `insert_div()` je využita hlavně k nastavení základní struktury aplikace v souboru `index.php`. Pro lepší představu uvádím zdrojový kód sloužící k vypsání základní struktury stránky a obrázek ilustrující strukturu `divů`, která bude tímto vnořováním funkce `insert_div()` vytvořena.

```
echo
insert_div("center",
insert_div("main",
insert_div("top",
insert_div("lang",insert_div("vlajky",lang(),"vlajky"),"lang").
```

```

insert_div("head",
    insert_div("navigation","", "navigation").
    insert_div("login",page_login(), "login"),
    "head"),
"top").
insert_div("middle",
    insert_div("menu",main_menu(), "menu").
    insert_div("content",$content, "content"),
    "middle").
insert_div("bottom",$_SESSION['init']['bottom'], "bottom"),
"main"),
"center");

```



**Obrázek č. 10: Grafické znázornění struktury webu**

Parametr s obsahem divu může být definován staticky jako je tomu v případě divu bottom, kdy je pouze zobrazen text patky uložený v globální proměnné `$_SESSION['init']['bottom']` a nebo pomocí proměnné jako je tomu v případě divu content. Soubor `index.php` obsahuje přepínač, který v závislosti na vstupu uživatele mění obsah proměnné `$content`.

Velice důležitou formátovací funkcí je `form_input()`. Tato funkce zajišťuje generování formulářových polí. Má povinné parametry `$type`, `$name`, `$value` a nepovinné parametry `$condition` a `$class`. Parametrem `$type` předáváme

funkci informaci o tom, jaký typ formulářového pole má být vytvořen. Nabývá tedy hodnot text, select, checkbox, radio, submit nebo password. Parametrem `$name` nastavíme atribut `name` a `id` formulářového pole a parametr `$value` obsahuje zobrazenou hodnotu. Pouze v případě, kdy chceme vytvořit formulářové pole typu select, obsahuje parametr `$value` vícerozměrné pole, kterým nastavíme jednotlivé hodnoty tagů `<option></option>`. V případě generování pole typu select je volaná funkce `form_select()`, která obsahuje algoritmus pro vytvoření seznamu s vybranou hodnotou. Všechna ostatní formulářová pole jsou generována funkcí `form_input()`. Podrobněji se zaměřím na funkci `form_input()`.

```
function form_input($type,$name,$value,$condition=NULL,$class=NULL) {
if ($value==FALSE) {$value_print='';}
    else{$value_print=' value="'. $value. '" ';}
if ($condition==NULL) $condition="";
if ($condition == "checked") $condition='checked="checked"';
if ($class==NULL){$class_print="";}
    else{$class_print='class="'. $class. '"';};
if ($type=="textarea") $result=form_textarea($name,$value,$class);}
elseif ($type=="select"){
$result=form_select($name,$value,$condition,$class);}
    else{ $result = '<input '. $class_print. ' id="'. $name. '"
type="'. $type. '" name="'. $name. '" '. $value_print. $condition. ' />';
}return $result;}
```

Ve zdrojovém kódu můžeme vidět, jakým způsobem probíhá nastavování jednotlivých proměnných, ze kterých je následně sestaven výsledný zdrojový kód formátu HTML. Abychom v případě použití formulářových polí typu checkbox nebo radio nemuseli při potřebě položku zaškrtnout vypisovat celou syntax `checked="checked"`, je funkce `form_input()` opatřena podmínkou, která v případě že parametr `$condition` obsahuje text „checked“, dosadí tuto syntax do parametru. Zde jsem demonstroval názorný příklad použití parametru `$condition`. Parametr `$class` opět slouží k dosažení názvu třídy pro kaskádové styly. V případě, že parametr `$type` obsahuje hodnotu `textarea`, jsou všechny parametry předány funkci `form_textarea()`, jelikož tag `<textarea>` používá odlišnou syntax pro výpis hodnoty a ne klasický atribut `value`. Pokud je parametr `$type` nastaven na hodnotu `select`, je volána funkce `form_select()`, která zaručí správné vygenerování seznamu hodnot. Konkrétní volání funkce `form_input()` může vypadat takto:

```
$a=form_input("text", "jmeno", FALSE ,NULL,
"prihlasovaci_formular");
```

Toto volání vygeneruje následující HTML kód a přiřadí ho proměnné `$a`:

```
<input class="prihlasovaci_formular" id="jmeno" type="text"
name="jmeno" />
```

Parametr `$value` můžeme nastavit na `$_POST['jmeno']`, tím docílíme toho, že pokud již byl formulář obsahující toto pole jednou odeslán s nějakou hodnotou, bude tato hodnota uložena do atributu `value`.

Funkce `array_to_table()` zajišťuje převedení vícerozměrného pole na tabulku. Její schopnosti sice nepokrývají veškeré možnosti práce s tabulkami, při formátování formulářových polí je však velmi užitečná. Funkce má čtyři parametry, přičemž jediný povinný parametr je parametr `$data`. Tímto parametrem se funkci předá pole s hodnotami. Tři nepovinné parametry se starají o formátování jednotlivých buněk, řádků a celé tabulky. Pokud chceme formátovat jednotlivé buňky, musí mít pole stejné rozměry jako pole obsahující vstupní data. Pokud chceme formátovat jednotlivé řádky, musí být počet prvků tohoto pole roven počtu řádků tabulky, kterou chceme vytvořit. V opačném případě funkce vyhodí hlášku o nekonzistentnosti dat. Formátování tabulky lépe vysvětlím pomocí následujícího příkladu:

```
$radky=array("prvni", "druhy", "treti");
$data=array(
array("jméno", "Vít"),
array("příjmení", "Filiposký"),
array("bydliště", "Rožmitál"));
$a=array_to_table($data,NULL,$radky, "udaje");
```

Pomocí tohoto PHP kódu vygenerujeme následující HTML syntax:

```
<table class="udaje">
<tr class="prvni"><td>jméno</td><td>Vít</td></tr>
<tr class="druhy"><td>příjmení</td><td>Filiposký</td></tr>
<tr class="treti"><td>bydliště</td><td>Rožmitál</td></tr>
</table>
```

## 6.4.5 Kontrolní

Kontrolních funkcí na rozdíl od funkcí předchozích není mnoho. Mezi ně patří funkce pro kontrolu proměnné `$_POST` a funkce pro kontrolu validity hodnot zadaných do formulářových polí.

Kontrolu pole `$_POST` zajišťuje funkce `form_characters()`, která je volána vždy, když pole obsahuje nějakou hodnotu, vyjma podmínky, kdy je pomocí pole `$_POST` odeslán soubor. Funkce provede veškeré doporučené postupy pro ošetření případného vložení škodlivého kódu do formulářového pole typu `text` nebo `textarea`. Tyto postupy jsem nastudoval z knihy *PHP a MySQL* od Larryho Ullmana. Jedná se především o ošetření vložení problematických znaků. Funkce `form_characters()` je sestavena přesně podle návodu z knihy.

```
if (count($_POST)>"0"){
    if (isset($_POST['upload_file'])){
    }else{
        foreach ($_POST as $post){
            form_characters($post);
        }
    }
}
function form_characters($data){
    if( ini_get ('magic_quotes_gpc')) {
        if (is_string($data)) {
            $data = stripslashes($data);
            $data = mysql_real_escape_string($data);
        }
    }
    return $data;
}
```

Nejprve jsou odstraněna všechna dvojité zpětná lomítka a poté opětovně vložena před problematické znaky. Tím je zajištěno, že proměnná nebude obsahovat dvojité lomítko, kterým by bylo možné zakomentovat část kódu a zároveň je zaručeno, že problematické znaky jako je znak větší, menší nebo uvozovky, budou opatřeny zpětnými lomítky a znemožní případnou injekci škodlivého PHP, javascriptu nebo SQL kódu.

## 6.5 Zabezpečení

Velice důležitou součástí celého systému je zabezpečení dat a hesel uživatelů. Přestože uživatelé budou nejspíše ve velké míře využívat možnost prezentace svých

článků, testů nebo souborů, je důležité zajistit, aby nebyla prezentována ta data, která si uživatel prezentovat nepřeje. Uživatel může například chtít na serveru uchovávat soubory důležité pro výuku, které před určitým datem nesmí být zveřejněny.

### 6.5.1 Kódování hesel

Nejvyšší důležitost má vhodné uchování hesla, které by neměl znát ani sám administrátor systému, jelikož lidé často používají k různým účtům stejná hesla a jejich kódováním se v případě vniknutí cizí osoby do databáze zamezí jejich zneužití.

Šifrování hesel v systému ensemame.cz probíhá pomocí hashovacího algoritmu MD5, který je přímo podporován jako funkce PHP. V praxi to znamená, že uživatelské heslo je podle algoritmu změněno na takzvaný hash nebo-li otisk. V databázi se pak uchovává pouze hash a nikoliv heslo samotné.

### 6.5.2 Omezení přístupu pomocí .htaccess

Velikým bezpečnostním rizikem bývá nekontrolované prohlížení složek pomocí webového prohlížeče. Adresářovou strukturu průměrně inteligentní uživatel rozpozná třeba při bližším prozkoumání adresního řádku nebo editací zdrojového kódu webu. Zvědavému uživateli pak stačí pouze adresu zkopírovat do adresní řádky a již si prohlíží obsah složek. Tomuto jevu lze zabránit užitím .htaccess. Soubor .htaccess může mít několik funkcí – přesměrovává stránku, nastavuje chybové stránky, umí kešovat soubory, nastavovat výchozí soubor a další. Nás zajímá hlavně funkce odepření přístupu do adresáře. Vytvoříme soubor s názvem .htaccess a vložíme do něj následující kód:

```
deny from all
allow from 90.182.22.42
```

První řádka zamezí všem přístup do složky pomocí webového prohlížeče. Druhá řádka nastaví IP adresu, která má tento přístup povolen, v našem případě je to veřejná adresa mého počítače. Soubor .htaccess pak jednoduše zkopírujeme do složky, kterou nechceme veřejnosti zpřístupnit.



## 6.6 Ovládání

Při návrhu a vlastní realizaci aplikace jsem se snažil, aby ovládání bylo co možná nejintuitivnější. Veškeré odkazy a grafická tlačítka jsem opatřil popiskami pro snazší pochopení jejich funkcí. Základní rozhraní sloužící pro čtení článků a vyplňování testů je složeno jen z vysunovacího menu rozděleného na kategorie a vlastního obsahu. Při návrhu grafické úpravy jsem se snažil o co nejkontrastnější kombinaci barev z důvodu dobré čitelnosti webu.

### 6.6.1 Vkládání testů

Rozhraní pro vkládání nových testů má jednoduché vysunovací menu, jehož karty se dají pro snazší vyplnění kliknutím na název zafixovat. Menu obsahuje celkem čtyři karty. Při změně vlastností na těchto kartách je vždy nutné změnu potvrdit klávesou enter nebo tlačítkem umístěným v dolní části karty.

První karta slouží k nastavení základních vlastností testu, kterými jsou název, autor, citace, url a zdroj. Formulářová položka název je z této karty jediná povinná. Všechny ostatní jsou volitelné a slouží hlavně k uvedení zdroje při použití testu, který byl převzatý z učebnice nebo z jiného webu.

Úlohou druhé karty je nastavení třídících parametrů testu. Učitel může nastavit gramatickou či tématickou kategorii. Může rovněž vytvořit téma nové nebo označit obtížnost testu podle standardizovaného označení A1-C2. Čím přesněji tyto vlastnosti nastaví, tím lépe se bude test zařazovat do kategorií.

Třetí karta je důležitá při nastavení vlastností, které bezprostředně souvisí s procesem testování. Těmito vlastnostmi jsou časový limit, maximální počet možností, počet zobrazovaných otázek, možnost míchat úlohy a skrýt test. Účel těchto vlastností si nyní podrobněji rozebereme. Vlastnost časový limit je z názvu vypovídající. Tímto parametrem nastaví učitel časový limit v minutách, za který má být test vykonán. Při nenastavení hodnoty je test časově neomezen. Po uplynutí časového limitu je test automaticky vyhodnocen. Žákovi jsou zobrazeny dva výsledky. Jeden, ve kterém jsou nevyplněné odpovědi brány jako špatné a druhý, do kterého se nevyplněné odpovědi nezapočítávají. Maximální počet možností je číslo určující maximální počet možných odpovědí u jedné úlohy. Tento počet je nastaven na deset, ale učitel ho může měnit. Atribut počet zobrazovaných otázek určuje, kolik úloh z testu bude žákovi zobrazeno. Větší smysl této vlastnosti dodá zaškrtnutí

atributu míchat úlohy. Pořadí úloh je při tomto nastavení při každém zobrazení testu jiné a navíc, pokud je počet zobrazovaných úloh menší než počet celkový, je test při každém zobrazení jedinečný. Počet možných kombinací můžeme vypočítat pomocí následujícího vzorce:

$$\binom{n}{k} = \frac{n!}{(n-k)! \cdot k!}$$

V tomto vzorci je  $n$  počet všech úloh a  $k$  je počet úloh, které vybíráme. Pokud tedy vytvoříme test o deseti úlohách a maximální počet zobrazených úloh bude sedm a samozřejmě políčko míchat úlohy bude zaškrtnuté, dostáváme 120 možných způsobů zobrazení tohoto testu. Pokud políčko míchat úlohy nezaškrtneme, bude vybráno prvních sedm úloh z testu. Poslední atribut skrýt test slouží k ukrytí testu před veřejností, třeba pro případ, kdy test není dokončen a učitel si ho nepřeje zobrazovat v nabídce.

Čtvrtá karta slouží k vložení formuláře pro testovou úlohu, vymazání všech formulářů a uložení testu do databáze. Při vložení nového formuláře učitel vyplní nejprve pole zadání. Poté dle své úvahy vytvoří vlastní jádro testu v poli znění. Nastaví počet variant a vybere ze čtyř možností typů testů, kterými jsou otevřené úlohy, seřazovací úlohy, spojovací úlohy a výběr z možností. Problematice návrhu a sestavování didaktických testů se podrobně věnuji v kapitole 4.1.3 Tvorba didaktických testů. Zde jen připomenu, že při vytváření testů s otevřenou odpovědí je důležité, aby byla vyžadována odpověď co nejstručnější, nejlépe jednoslovná. Program před kontrolou shody odpovědí převádí hodnoty na malá písmena. Další práce v editoru je celkem intuitivní. Důležité je, aby učitel při dokončení změn každého formuláře tyto změny potvrdil buď stiskem klávesy enter a nebo tlačítkem potvrzení.

## 6.6.2 Vkládání článků

Rozhraní pro vkládání článků je podstatně jednodušší než rozhraní předchozí. Obsahuje o poznání méně formulářových polí a jeho vyplnění je proto snazší. Formulář obsahuje pole úvod, motivace, expozice a fixace, přičemž je kontrolováno vyplnění všech polí, ale jen při nevyplnění pole úvod a expozice nedojde k uložení článku do databáze. Učitel má díky WYSIWYG editoru možnost doplnit článek

o obrázky, zvuky či videa, které buď může nalinkovat z jiné internetové stránky, nebo přímo ze svých zdrojů nahraných na webu [www.ensename.cz](http://www.ensename.cz).

### 6.6.3 Správa souborů

Aplikace [www.ensename.cz](http://www.ensename.cz) je vybavena rozhraním pro nahrávání, stahování a sdílení souborů. Opět se jedná o intuitivní jednoduché prostředí, které se skládá z několika formulářových polí a čtyř tlačítek.

Pro zobrazení souborů slouží velké formulářové pole typu `select`. Pro toto řešení jsem se rozhodl, protože tento způsob při užití atributu `multiple` umožňuje snadnou manipulaci s více položkami najednou. Mimo to při stisknutí písmena na klávesnici se výběr automaticky přesune na položku začínající tímto písmenem a při stisknutí klávesové zkratky `Ctrl + a` se provede výběr všech položek.

Druhé formulářové pole slouží k filtraci podle typu souborů. Aplikace třídí soubory na obrázky, zvuky, dokumenty, videa a flash animace. Uživatel má pak možnost zobrazit pouze určité typy souborů.

Další formulářové pole je opět typu `select` a umožňuje sdílet soubory mezi ostatními skupinami. Sdílení probíhá tak, že uživatel vybere konkrétní soubory, které chce sdílet, poté vybere skupinu a v poslední řadě vybere z dalšího formulářového pole oprávnění, které dané skupině k souboru přiřadí. V celé aplikaci lze nastavit tři typy oprávnění a to úplné řízení, čtení a zápis a čtení bez zápisu. Oprávnění lze rovněž odebrat. Pro souborový manažer jsou nastavení oprávnění pro čtení a zápis a čtení bez zápisu totožná, jelikož v současné době není zavedeno žádné rozšíření, které by přímou editaci souborů na serveru umožňovalo. Úplné řízení tedy umožní skupině sdílený soubor prohlížet, stahovat, sdílet a mazat. Čtení a zápis a čtení bez zápisu umožní soubor prohlížet, stahovat a dál sdílet. Odebrání sdílení zruší veškerá oprávnění skupiny.

Poslední formulářové pole typu `select` umožňuje sdílet soubory s konkrétními uživateli. Nastavení sdílení probíhá stejně jako v předchozím případě. Uživatel vybere soubory, které chce sdílet s jiným uživatelem. Poté vybere jeho jméno a nastaví oprávnění, která chce danému uživateli přiřadit.

Vedle formulářových polí typu `select` se na stránce nachází čtveřice tlačítek

sloužící ke správě souborů, respektive k jejich nahrávání, stahování, mazání a zobrazení detailu. Nahrávání souborů je řešeno pomocí formulářového pole typu file. Toto pole má nastaven atribut `multiple`, což umožňuje nahrání více souborů najednou. Maximální počet souborů je nastaven na patnáct a maximální velikost jednoho souboru je nastavena na 5Mb. Uživatel může tedy jednorázově nahrát maximálně 75Mb dat. Případné zvýšené nároky uživatelů budou řešeny individuálně. Stahovat více souborů najednou je rovněž možné. Uživatel označí soubory pro stažení a stiskne tlačítko stáhnout. Pro každý označený soubor se vytvoří vyskakovací okno, které stahování zahájí. U tohoto systému může nastat problém, pokud má uživatel v prohlížeči nastavené blokování vyskakovacích oken. V tomto případě nelze stáhnout více souborů najednou a uživatel si musí zobrazit detaily souboru a zde kliknout na odkaz, který zahájí stahování. Tlačítko smazat odstraní soubor ze serveru a zároveň odstraní všechny zápisy v databázi, ve kterých je tento soubor uložen.

#### **6.6.4 Seskupování studijních materiálů**

Kromě nastavování oprávnění může učitel rovněž nastavit vazby mezi jednotlivými testy, články a soubory nebo vytvořit vlastní skupinu a přidat do ní studenty. V praxi tento proces probíhá tak, že učitel nejprve vybere ze seznamu typ vazby. Vazeb existuje několik typů a to vazba typu článek - článek, článek - test, test - test, soubor - test a soubor - článek. Těmito vazbami učitel nastaví, které materiály se budou zobrazovat při prohlížení daného testu nebo článku. Mimo to může učitel vytvořit novou pracovní skupinu, jíž se stane automaticky správcem. Do této skupiny může přidávat další uživatele. Tento systém je užitečný, chce-li učitel sdílet konkrétní materiály pouze s určitými uživateli.

## **7 Zhodnocení výsledků**

### **7.1 Využití**

Vzhledem k povaze aplikace je pravděpodobné, že má potenciál nalézt své uplatnění mezi žáky a učiteli všech typů škol. Používání aplikace není vázáno na žádnou instituci, tudíž jí mohou učitelé i studenti využívat během celého studia. Studium nejen jazyka je celoživotní proces a dokončení jednoho cyklu neznamena, že studenti nemohou nadále zlepšovat své vědomosti. V tomto směru jim může webová stránka [www.ensename.cz](http://www.ensename.cz) poskytnout dobrý prostor pro archivaci studijních materiálů a jejich rozšiřování za pomoci dalších uživatelů. Ensename.cz má potenciál stát se pomocníkem a průvodcem během studia španělského jazyka, ne strukturovaným vzdělávacím programem. Vše závisí jen na množství uživatelů, ochotných rozšiřovat obsah a stránku aktivně využívat a na mém budoucím úsilí web propagovat a tyto uživatele aktivně oslovovat.

### **7.2 Možnosti**

Aplikace je ze své podstaty navržena tak, aby mohla být do budoucna rozšířena o různé funkce, jejichž absence se používáním odhalí. Aplikace může být rozšířena o počet multimediálních zařízení, na kterých se její obsah bude zobrazovat. Mohou být rovněž přidány funkce, které umožní generování a ukládání dat v různých formátech nebo může být web rozšířen o další jazykovou mutaci. Funkcí, které by nějakým způsobem ulehčily studentům i učitelům proces výuky, se jistě najde mnoho, ale rád bych jejich případné doplňování řešil až po konzultaci s uživateli, pro něž budou tyto funkce sloužit především.

Aplikace může být rovněž upravena a použita k výuce úplně jiného jazyka, popřípadě k výuce úplně jiného předmětu. Nedělám si iluze, že by se aplikace mohla ujmout v nějaké komerční vzdělávací instituci jako oficiální komunikační systém, její možnosti jsou limitovány způsobem a kvalitou naprogramování. Přestože věřím, že jsem při jejím vývoji odvedl kvalitní práci, nemohu zajisté konkurovat komplexním vzdělávacím systémům typu moodle.

## 8 Závěr

V současnosti zaznamenává španělský jazyk značný zájem nejen mezi Evropany. Dostává se mezi volitelné jazyky na víceletých gymnáziích, zaměřují se na něj jazykové školy a na vysokých školách je již nabízen jako hlavní studijní obor. Studovat španělštinu tedy není v České republice v současnosti již nic exotického a tomu by měly odpovídat i možnosti a způsoby, jakými se studenti a učitelé mohou dostat ke studijním materiálům. Existuje zajisté spousta kvalitních učebnic a vzdělávacích institucí, možnosti hispanistů na českém internetu jsou však v současnosti značně omezené. Weby si nekonkurují ani navzájem nespolupracují a je velice málo těch, ve kterých by studenti a učitelé mohli bezplatně vyhledávat informace a studijní materiály. Mezi nejzajímavější projekty, které se snaží českému internetu nabídnout takovéto služby, patří weby [www.spandelstina-zdarma.cz](http://www.spandelstina-zdarma.cz) a [www.jazyky-online.info/spandelstina](http://www.jazyky-online.info/spandelstina). Tyto weby sice neumožňují přímou participaci uživatelů na tvorbě obsahu, je však možné kontaktovat jejich správce a ty v případě zájmu oslovit s nabídkou spolupráce. V takovém případě zájemce vypracovává svoje příspěvky v podobě textových souborů, které pak administrátorům zašle, ti je prohlédnou, opraví a umístí na web. Zájemce může také získat oprávnění a příspěvky vkládat sám rovnou do systému. Tyto informace jsem čerpal z emailové korespondence s administrátory těchto dvou zmiňovaných webů.

Pokud uživatelé mají zájem přidávat na web kvalitní obsah, je důležité je upozornit na to, že vkládání článků, testů i souborů má svá pravidla a jen dodržováním těchto pravidel se podaří vytvořit dobré distanční texty a validní a reliabilní didaktické testy. Tvorba didaktických testů podléhá pravidlům, která nám při dodržování zajistí, že vytvoříme takový didaktický test, který bude pro žáky atraktivní a zároveň jim pomůže procvičit jejich dovednosti a nabídne kvalitní zpětnou vazbu nejen jim, ale dokonce i učitelům. Zpětná vazba je základním pilířem každé výuky, a proto je důležité dbát na to, aby všechny jevy v didaktických testech byly řádně objasněny. Jen tak se může student poučit a svůj výkon zlepšovat. Rovněž je důležité tuto vazbu poskytnout i učiteli. Pokud bude mít učitel k dispozici přehled výsledků studentů, kteří absolvovali jím vytvořený test nebo komentáře k článkům, které vytvořil, snáze v nich vyhledá a opraví případné nejasnosti nebo chyby a tímto způsobem může svoje výukové materiály postupně zdokonalovat.

Přes veškerá rizika jsem se rozhodl naprogramovat aplikaci, jejímž hlavním kritériem bude umožňovat uživatelům přímou participaci na vytváření uveřejňovaného obsahu. Jsem si vědom nebezpečí, které s sebou přináší velká volnost uživatelů na webových portálech a je jasné, že se nevyhnu mazání neadekvátních příspěvků, dokumentů či článků. Na druhou stranu je možné, že registrovaní uživatelé začnou dbát na kvalitu svojí prezentace na tomto kanálu a tím pádem zvýší kvalitu celého webu. Je pravděpodobné, že s rostoucím počtem uživatelů vzroste i počet učitelů, kteří budou chtít tyto uživatele zaujmout svými články. Záleží tedy převážně na počtu a odhodlání prvních uživatelů, jakým způsobem a jakou rychlostí se stránka bude dále vyvíjet. Z mého hlediska bych rád nabídl každému zájemci možnost se v tomto projektu realizovat, jelikož i středně pokročilý student může napsat kvalitní materiál pro výuku studentů na nižší úrovni. Samotné psaní distančního textu nebo vytváření testu je beze sporu prospěšné i pro rekapitulaci učitelových znalostí a dovedností.

Vývoj i naprogramování celé aplikace pro mne byly velikou výzvou a s odstupem času jsem rád, že jsem si zvolil právě toto téma. Jednak mi jeho zvolení umožnilo spojit mé dva velké zájmy (programování a španělštinu) a co víc, do značné míry mi během vývoje a realizace celé práce rozšířilo vědomosti v obou oblastech. Také mne velice potěšil kontakt s učiteli a zjištění jejich názorů. Pevně věřím, že tato práce nevyjde naplano a že se mi podaří pro tento prázdný korpus sehnat kvalitní a zapálené tutory, kteří nebudou motivováni finančním ohodnocením ale stejným zápalem, jaký motivoval mě při tvorbě tohoto projektu.

## 9 Seznam použitých zdrojů

- [1] HRABAL, Vladimír, Zdena LUSTIGOVÁ a Ludmila VALENTOVÁ. *Testy a testování ve škole*. Praha: Pedagogická fakulta UK, 1992. 100 s. ISBN 0862 - 156x.
- [2] CHRÁSKA, Miroslav. *Metody pedagogického výzkumu: Základy kvantitativního výzkumu*. Praha: Grada, 2007. 272 s. ISBN 978-80-247-1369-4.
- [3] KOPECKÝ, Kamil. *E-learning: nejen pro pedagogy*. Olomouc: Hanex, 2006. 125 s. ISBN 80-85783-50-9.
- [4] PEXA, Petr. *Jazyky XHTML, CSS, DHTML, WML: Kompletní referenční příručka pro tvorbu webu a WAPu*. České Budějovice: Kopp, 2006. 207 s. ISBN 80-7232-286-9.
- [5] STÁRKOVÁ, Dagmar. *Model ADDIE při vytváření koncepce výuky a jeho aplikace* [online]. 2012 [cit. 2013-04-13]. Dostupné z: <[http://it.pedf.cuni.cz/strstud/edutech/2012\\_Addie\\_Starkova/](http://it.pedf.cuni.cz/strstud/edutech/2012_Addie_Starkova/)>.
- [6] ULMAN, Larry. *PHP a MySQL: Názorný průvodce tvorbou dynamických WWW stránek*. Brno: Computer Press, 2004. 534 s. ISBN 80-251-0063-4.
- [7] *Centrum pro zjišťování výsledků vzdělávání* [online]. 2010 [cit. 2013-04-13]. Dostupné z: <<http://www.ceremat.cz/didakticke-testy-1404034141.html>>.
- [8] *Druhy a příklady didaktických testů*. Infogram: portál pro podporu informační gramotnosti [online]. 2013 [cit. 2013-04-13]. Dostupné z: <<http://www.infogram.cz/article.do?articleId=2149>>.



## 10 Seznam tabulek, obrázků a příloh

### 10.1 Seznam tabulek

Tabulka č. 1: Kategorizace didaktických testů.....	14
Tabulka č. 2: Porovnání vstupního parametru s výstupem funkce select_check() .....	59

### 10.2 Seznam obrázků

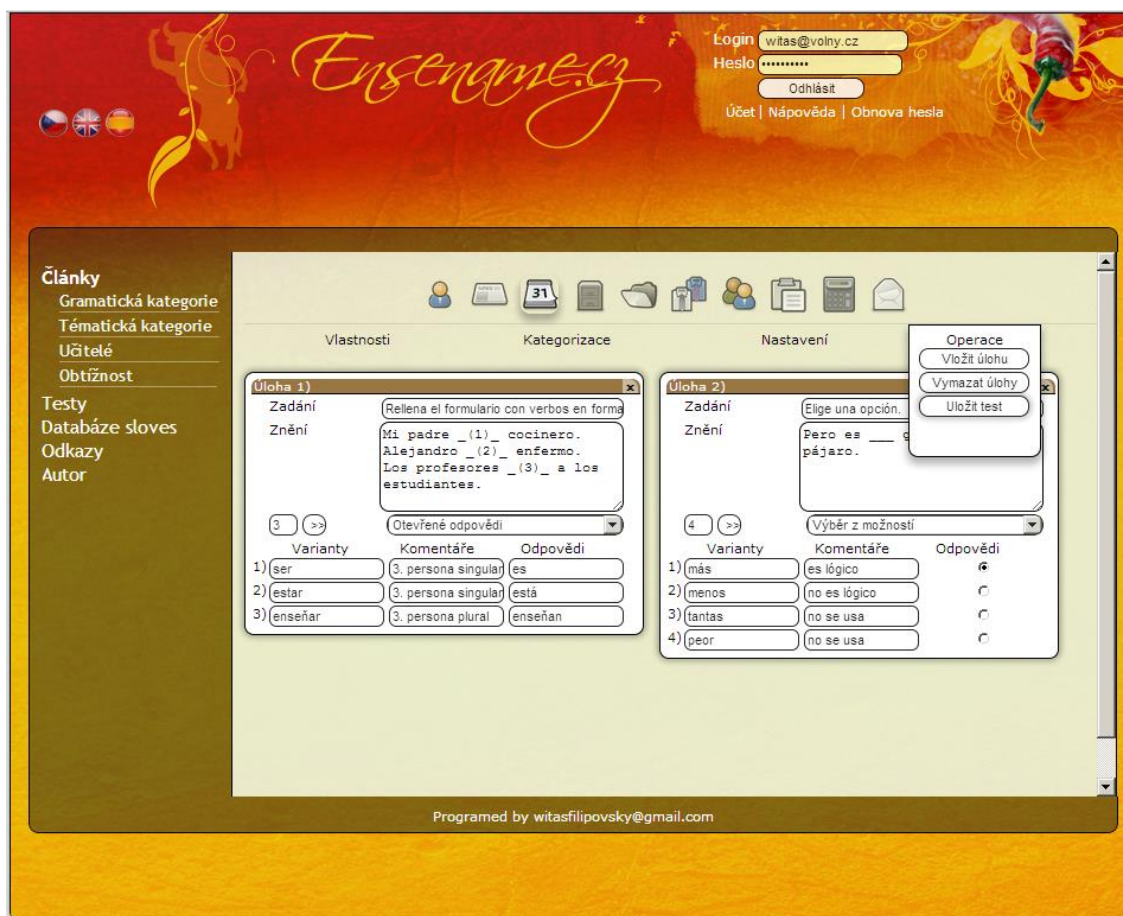
Obrázek č. 1: Struktura modelu ADDIE .....	24
Obrázek č. 2: Adresářová struktura aplikace ensoname.cz .....	41
Obrázek č. 3: Obsah adresáře systém.....	43
Obrázek č. 4: Obsah adresáře functions .....	46
Obrázek č. 5: Obsah adresáře layout.....	47
Obrázek č. 6: Obsah složky templates .....	50
Obrázek č. 7: Přehled všech tabulek v databázi.....	51
Obrázek č. 8: Struktura vazebních tabulek potřebných ke správě testů.....	52
Obrázek č. 9: Přiřazení uživatele ke skupině a nastavení oprávnění .....	53
Obrázek č. 10: Grafické znázornění struktury webu.....	60

### 10.3 Seznam příloh

Příloha I: Náhled rozhraní pro tvorbu didaktických testů.....	74
Příloha II: Náhled rozhraní pro správu souborů.....	75
Příloha III: Náhled testovacího rozhraní .....	75
Příloha IV: Náhled rozhraní pro správu účtu .....	76

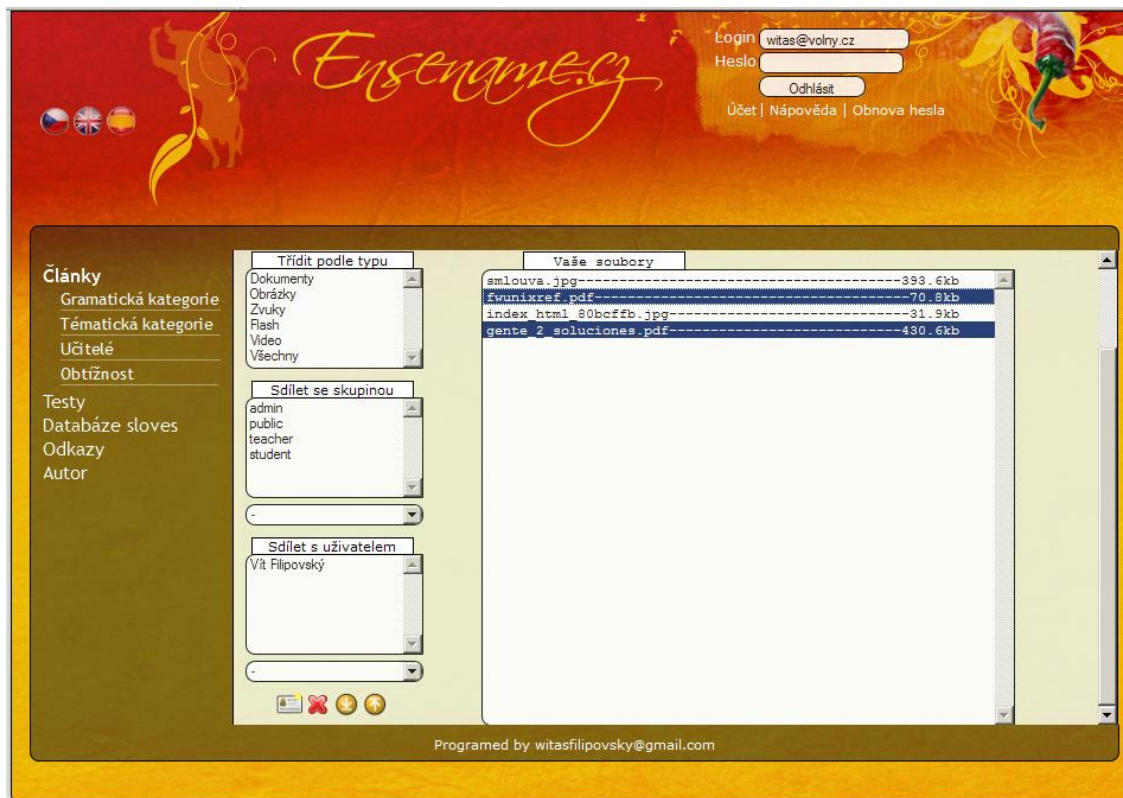
# 11 Přílohy

## Příloha I: Náhled rozhraní pro tvorbu didaktických testů<sup>45</sup>



<sup>45</sup> Vlastní

## Příloha II: Náhled rozhraní pro správu souborů<sup>46</sup>



## Příloha III: Náhled testovacího rozhraní<sup>47</sup>



<sup>46</sup> Vlastní

<sup>47</sup> Vlastní

#### Příloha IV: Náhled rozhraní pro správu účtu<sup>48</sup>

Engname.cz

Login: witas@volny.cz  
Heslo: .....  
Odhlásit  
Účet | Návod | Obnova hesla

**Články**  
Gramatická kategorie  
Tématická kategorie  
Učitelé  
Obtížnost  
Testy  
Databáze sloves  
Odkazy  
Autor

Jméno: Vít  
Příjmení: Filipovský  
Login: witas@volny.cz  
Datum registrace: 2012-09-23 15:19:51  
Email: witas@volny.cz  
Staré heslo:   
Nové heslo:   
Znovu:   
Rok narození: 1987  
Město: Rožmitál pod Třemšínem  
Ulice: Palackého  
Čp: 68  
Psč: 26242  
Telefon: 605266701  
Poznámka:

Aktualizovat   Obnovit

Programed by witasfilipovsky@gmail.com

<sup>48</sup> Vlastní