

System pro testování žáků

Miroslav Hrončok, 3.E

Ročníková práce z předmětu IVT na

Gymnáziu Arabská ve školním roce 2008/2009

Obsah dokumentace

1	Zadání ročníkové práce	3
1.1	Objednávka	3
1.2	Resume in English	3
2	Analýza problému aplikací řešeného	4
2.1	Podstata testu	4
2.1.1	Typy otázek	4
2.1.1.1	Otázka s odpovědí ano, nebo ne	4
2.1.1.2	Otázka s číselnou odpovědí	4
2.1.1.3	Otázka s textovou odpovědí	5
2.1.1.4	Otázka s možnostmi	5
2.2	Podstata žáka	6
3	Návrh datových struktur	7
3.1	Jednotlivé tabulky a jejich položky	7
3.1.1	Tabulka témat	7
3.1.2	Tabulka testů	7
3.1.3	Tabulka otázek	7
3.1.4	Tabulka možných odpovědí	8
3.1.5	Tabulka uživatelů / žáků	8
3.1.6	Tabulka hodnocení	8
3.1.7	Tabulka nastavení	8
3.2	Relace mezi tabulkami	9
4	Návrh algoritmu práce aplikace	10
4.1	Vytvoření uživatele	10
4.2	Vytvoření testu	10
4.3	Vypracovávání testu	10
4.4	Vyhodnocení testu	10
4.4.1	Hodnocení otázky s možnostmi	10
4.4.2	Ukázka komentovaného kódu	11
5	Grafické vyjádření datového modelu úlohy	14
6	Grafické vyjádření algoritmu úlohy	15
7	Uživatelská příručka	15
7.1	Podmínky běhu aplikace	15
7.2	Definování přístupu do databáze	15
7.3	Vytvoření tabulek v databázi	15
8	Seznam použité literatury	15

9 Závěr	16
9.1 Klady aplikace	16
9.2 Zápory aplikace	16
9.3 Náměty na eventuální rozšíření	16
9.3.1 Statistika	16
9.3.2 Lepší systém přihlašování uživatelů	16
9.4 Postřehy	16
10 Oponentský posudek	17

1 Zadání ročníkové práce

Prvním bodem dokumentace k ročníkové práci má být smyšlený dopis, případně mail. V něm mám uvést stručné nastínění tématu práce, požadavky na vývojové nástroje pro tvorbu relační databázové aplikace, specifika, proč je nutno vytvořit aplikaci „na zakázku“ a nelze použít komerční software, možnosti využití aplikace. Součástí zadání bude resumé (respektive abstrakt) v jazyce anglickém.

1.1 Objednávka

From: "RNDr. Pavel Mára" <pavel.mara@zshornidolni.cz>
To: "Miroslav Hrončok" <churchyard@gmail.com>
Subject: System pro testování žáků
Date: Sun, 10 May 2009 20:34:19 +0200

Vážený pane Hrončoku,

Jakožto ředitel základní školy v Horní Dolní bych se na Vás chtěl obrátit s nabídkou práce. Dostali jsem na Vás doporučení od pana Romana Pokorného.

Naše škola má několik počítačových učeben a chceme veškerou výuku provozovat na nich. Hledáme řešení pro testování žáků z nabytých znalostí ve formě webové aplikace. Vyučující bude moci ve webovém prohlížeči vytvořit test a žáci pak takto vytvořený test budou moci vyplnit a odeslat, výsledky se však budou vyhodnocovat automaticky a tím ušetří čas učitelé, který jinak musí testy zdlouhavě pročitat a opravovat ručně.

Konkrétní požadavky:

- možnost vytváření a správy uživatelů (žáků) – identifikace pomocí uživatelského jména
- centrální učitelské heslo (učitelé nemusí mít různé účty)
- řazení testů do předmětů, případně tématických okruhů
- různé typy otázek
 - otázka s možnostmi (libovolný počet odpovědí je správně)
 - otázka s číselnou odpovědí (z oboru reálných čísel)
 - otázka s textovou odpovědí
 - otázka na jednoznačné ano/ne
- bodové ohodnocení otázek
- uložení hodnocení konkrétních žáků u konkrétních testů ve formě procent z maximálního počtu dosažitelných bodů
- nemožnost úpravy testu po jeho uvolnění (kvůli fair-play)
- nemožnost udělat jeden test vícekrát za sebou (kvůli fair-play)
- statistika u žáka (jaké testy, kdy a za kolik procent)
- statistika u testu (nejlepší žáci)

Obracíme se na Vás, protože jsem na trhu nenalezli hotové řešení, které by splňovalo naše požadavky. Pokud Vás nabídka zaujala, zašlete prosím odpověď s cenovým ohodnocením a případnými dotazy.

Děkuji

RNDr. Pavel Mára, ředitel školy

1.2 Resume in English

I should create a web application for managing tests for students of an elementary school. The teacher should be able to create the test in web browser and students to take it and send the answers to the application. The application itself should count the rank of the answers. The statistics of taken tests should be also available.

2 Analýza problému aplikací řešeného

Obsahuje pravidla, podle kterých bude aplikace pracovat (např. pravidla pro jednotlivé položky a účel sbíraných dat, zdůvodnění v aplikaci použitých matematických a statistických metod, resp. funkcí).

2.1 Podstata testu

Na začátku jsem si potřeboval stanovit několik bodů, kterými se budu řídit.

- a) Test patří do právě jednoho tématu.
- b) Test je souborem jedné a více otázek různých typů.
- c) V závislosti na typu otázky, existují správné i nesprávné možnosti odpovědi, které je třeba evidovat.

2.1.1 Typy otázek

Vyvstala otázka ohledně typů otázek, proberu je tedy po jednom.

2.1.1.1 Otázka s odpovědí ano, nebo ne

Tento typ otázky je velice prostý. Žák rozhodne, zda je tvrzení pravdivé, nebo ne. Příkladem takové otázky může být třeba:

Je tučňák savec?

Zde je jasné, že tučňák savec buďto je, nebo není. Pokud bychom však kladli otázku na barvu tučňáků, s tímto typem otázky bychom si nevystačili.

Vzhledem k tomu, že odpovědi na tuto otázku může být pouze **ano**, nebo **ne**, není nutné v databázi ukládat možné odpovědi. Stačí pouze u otázky jednoznačně rozlišit dva typy. Otázka s odpovědí ano a otázka s odpovědí ne. Detailněji se tomu budu věnovat v kapitole n3.

2.1.1.2 Otázka s číselnou odpovědí

Další typ otázky. Tentokrát jde o otázku, na kterou lze odpovědět nějakou položkou z oboru reálných čísel. Příkladem může být třeba:

Vyjádřete x z rovnice $x - 5 = 3$.

Problém ovšem nastane, například u otázky:

Vyjádřete x z rovnice $2x - 4 = 2x - 6$.

Jak jistě víte, taková rovnice nemá řešení, respektive x představuje prázdnou množinu. Prázdná množina však nelze vyjádřit žádnou položkou z oboru reálných čísel. Jde tedy o otázku, která má relativně omezenou funkčnost.

V databázi je uchovávána jedině jedna možná odpověď, tedy ta správná.

2.1.1.3 Otázka s textovou odpovědí

Další z řady omezených otázek. Jedná se o otázku, na kterou lze odpovědět pouze textem. Příkladem je jednoduchá otázka:

Kdo napsal knihu Pán prstenů?

Pokud k dané otázce nenabízíme žákům žádné možnosti, jedná se otázku tohoto typu. Je jasné, že správnou odpovědí není ano, ne, ani reálné číslo.

Pokud nebudu brát v úvahu jiné knihy se stejným názvem, je správná odpověď logicky jen jedna. Pána prstenů napsal John Ronald Reuel Tolkien.

Tato úvaha je však chybná, protože správnou odpovědí může být i Tolkien, nebo J. R. R. Tolkien. Co s tím?

Relativně zajímavým způsobem zpracování této otázky by bylo, kdyby učitel místo správné odpovědi zadal regulární výraz, kterému musí žákova odpověď vyhovovat, aby byla vyhodnocena správně. Nepředpokládám však, že učitel literatury bude ovládat regulární výrazy.

Další možností, mnohem méně elegantní, je uvést několik akceptovatelných odpovědí. Například tři příklady zmiňované výše. Jenže zrovna u této otázky je možností strašně moc, různé pořadí jmen, různé kombinace zkratk a nezkratk. Akceptujeme vynechání jednoho jména? V tom případě také různé kombinace vynechaných křestních jmen.

Nejlepším řešením v poměru cena / výkon je nabádat žáky ke konkrétně znějící odpovědi.

Kdo napsal knihu Pán prstenů? Uvedte prosím jen příjmení a to včetně diakritiky.

Všimněte si, že tímto způsobem lze pomalejší žáky zapeklitě zmást.

V aplikaci jsem proto zvolil metodu více možných správných odpovědí (pokud to je nevyhnutelné) v kombinaci s nabádáním učitele. V databázi proto uchovávám všechny akceptovatelné správné odpovědi.

2.1.1.4 Otázka s možnostmi

Naprosto základní otázka, oblíbená u všech studentů (mohou si tipnout) i kantorů (rychlý způsob vyhodnocování správnosti). Otázka má určitý počet možností¹, z toho libovolný počet je správně.

Příkladem takové otázky je tato:

Které z těchto zvířat má čtyři končetiny?

- vlk
- včela
- pes
- pavouk
- žížala

¹ Otázka s jednou možností je logicky totožná s otázkou na ano / ne.

- klokan

Je celkem jasné, že správné odpovědi jsou vlk, pes a klokan. Způsobem vyhodnocování odpovědí žáků a jejich bodováním se budu zabývat v kapitole 4. U tohoto typu otázky je potřeba uchovávat v databázi dobré i špatné možnosti a jednoznačně je odlišit.

2.2 Podstata žáka

- a) Učitel smí přidávat a mazat žáky, měnit jim hesla, ale nesmí mazat nebo měnit jejich výsledky.
- b) Žák má nárok na vykonání testu pouze jednou.
- c) Žák si může změnit heslo, ale nesmí ovlivňovat výsledky testů jinak, než jejich vypracováním.
- d) Uchovává se pouze hodnocení testu, tedy nikoli konkrétní odpovědi každého žáka.

3 Návrh datových struktur

V této části uvedu popis datového modelu aplikace a to jednotlivých datových tabulek a jejich položek, stanovení primárních klíčových položek a relací mezi tabulkami. Grafické vyjádření naprosto nelogicky uvedu až v kapitole 5.

3.1 Jednotlivé tabulky a jejich položky

Databáze této aplikace obsahuje celkem sedm tabulek, v této části se na ně podrobně podíváme. Primární klíčové položky jsou označeny **tučným písmem**. Neklíčové, ale unikátní položky jsou označeny *kurzívou*. Vezmu tabulky popořadě podle logiky věci.

3.1.1 Tabulka témat

sloupec	typ	význam
id	int(10)	
<i>name</i>	varchar(255)	název tématu

Tabulka 1: topics

3.1.2 Tabulka testů

sloupec	typ	význam
id	int(10)	
topic_id	int(10)	téma testu
<i>name</i>	varchar(255)	název testu
datetime	int(10) ²	datum vytvoření

Tabulka 2: tests

3.1.3 Tabulka otázek

sloupec	typ	význam
id	int(10)	
test_id	int(10)	v jakém je testu
type	int(10)	typ otázky ³
content	text	znění otázky
points	int(10)	bodové ohodnocení

Tabulka 3: tasks

² S datem a časem raději zacházím jako s číslem, jedná se o POSIX notaci

³ 1 pro otázku s odpovědí ano, 2 pro otázku s odpovědí ne, 3 pro číselnou, 4 pro textovou, 5 pro otázku s možnostmi

3.1.4 Tabulka možných odpovědí

sloupec	typ	význam
id	int(10)	
task_id	int(10)	k jaké je otázce
truth	char(1) ⁴	správnost odpovědi
content	text	znění odpovědi

Tabulka 4: answers

3.1.5 Tabulka uživatelů / žáků

sloupec	typ	význam
id	int(10)	
username	varchar(100)	uživatelské jméno
password	varchar(40)	heslo

Tabulka 5: users

3.1.6 Tabulka hodnocení

sloupec	typ	význam
id	int(10)	
user_id	int(10)	uživatel
test_id	int(10)	test
datetime	int(10)	čas odevzdání testu
result_value	int(10)	dosažené hodnocení v %

Tabulka 6: results

3.1.7 Tabulka nastavení

Tato tabulka obsahuje název webu, heslo učitele a maximální počet možností k otázce. Položka *conf_name* obsahuje název vlastnosti, tedy například *rootpassword* a položka *conf_value* pak samotnou hodnotu, tedy například konkrétní heslo učitele.

sloupec	typ	význam
conf_name	varchar(50)	název položky
conf_value	varchar(255)	hodnota

Tabulka 7: config

4 Nahrazení typu Boolean. 0 pro FALSE a 1 pro TRUE.

3.2 Relace mezi tabulkami

Relace mezi tabulkami pochopíte z názvů položek, nebo z grafického návrhu v kapitole 5.

4 Návrh algoritmu práce aplikace

V této části uvedu popis reakcí, které se vykonají na základě některých uživatelských akcí. Uvedu výpis úseku komentovaného zdrojového kódu.

4.1 Vytvoření uživatele

Formulář pro vytvoření uživatele obsahuje čtyři položky. Položku pro uživatelské jméno, dvě položky pro heslo (pro ověření) a položku pro učitelské heslo.

Pokud uživatel se stejným jménem neexistuje, hesla se shodují a učitelské heslo je správně, uživatel je úspěšně přidán do databáze. K dávkovému přidání uživatelů je lepší použít samotné SQL.

4.2 Vytvoření testu

Vytvoření testu je o něco komplikovanější procedura. V první fázi učitel definuje téma testu. Buď výběrem ze stávajících, nebo zadáním nového tématu. V případě zadání nového tématu se kontroluje, zdali již neexistuje. V takovém případě je použito již existující téma. Definuje také název testu a typ jeho první otázky. Test se uloží do databáze a učitel postupně vytváří otázky, které se také ukládají do databáze postupně. Spolu s nimi i případné záznamy o možných odpovědích.

Ve chvíli, kdy je test vytvářen, jej uživatelé mohou začít řešit nedokončený. Nemělo by to však nastat, protože učitel zadá žákům, kdy mají test vypracovat.

4.3 Vypracování testu

Žák test vypracovává celý nejednou, nikoli postupně po otázkách. Je to proto, aby si mohl vybrat, v jakém pořadí je bude řešit.

Vidí zadání otázky a prostředky k vyjádření odpovědi - vstupní pole, zaškrtačací políčka, atp. Za textem otázky následuje její bodové ohodnocení.

Žák zvolí podle svých vědomostí odpovědi a odešle formulář na server, tam probíhá zpracování a vyhodnocení. Hodnocení se pak uloží do databáze.

4.4 Vyhodnocení testu

Nejdůležitější částí aplikace je vyhodnocování testu. Hodnocení otázek všech typů, kromě posledního, funguje na bázi všechno, nebo nic. V případě otázky z možnostmi to je ale jinak.

4.4.1 Hodnocení otázky s možnostmi

Pokud žák odpověděl absolutně správně, tedy zaškrtnal pouze správné odpovědi, případně nezaškrtnl žádnou, pokud byly všechny chybné, dostane plný počet bodů a hotovo.

Pokud nezaškrtnl nic, ale alespoň jedna odpověď byla správná, nedostane žádné body a má smůlu, měl se lépe učit.

Mimo tyto krajní případy se stává, že žák nezaškrtně všechny správné odpovědi a/nebo některé zaškrtně navíc. V takové případě nastupuje takzvané *Hrončokovo hodnocení otázek s téměř libovolným počtem správných odpovědí*. Zatím tato procedura nebyla patentována jen díky tomu, že to u nás naštěstí nejde.

1. Nejdříve se rozdělí počet bodů za otázku mezi správné odpovědi. Tedy maximální počet dosažených bodů se vydělí počtem správných odpovědí. Nulou nedělíme, je to ošetřené v prvním odstavci této kapitoly.
2. Odečteme od zaškrtnutých správných odpovědí ty nesprávně zaškrtnuté (tedy odpovědi, které nejsou pravdivé, ale žák je tak označil). Pokud dostaneme nekladné číslo, za otázku žák body nedostane. Body z jiných otázek se ale odčítat nebudou.
3. Kladným číslem vynásobíme číslo, které jsme dostali v prvním bodě. Zaokrouhlíme nahoru na celé číslo.

4.4.2 Ukázka komentovaného kódu

```
function spocitej_vysledek($data) {
    $test_id = $data['testid'];
    $db = mysql_query("SELECT * FROM " . TASKS_TABLE . " WHERE test_id = '$test_id'
ORDER BY id");
    $points = 0;
    $maxpoints = 0;
    while ($row = mysql_fetch_array($db)) {
        $output .= '<fieldset>' . "\n\r";
        $output .= '<legend>' . task_content($row['id']) . ' (' .
pointy(task_points($row['id'])) . ') </legend>' . "\n\r";
        $maxpoints = $maxpoints + task_points($row['id']);
        if ($row['type'] == 1) { // jde o yes otazku
            if ($data['yesorno' . $row['id']] == "yes") {
                $youranswer = 'Ano';
                $points = $points + task_points($row['id']);
            } else {
                $youranswer = 'Ne';
            }
            $output .= '<p>' . "\n\r";
            $output .= ' <span class="trueanswer"><strong>Správná
odpověď:</strong> Ano<br /></span>' . "\n\r";
            $output .= ' <span class="youranswer"><strong>Tvoje odpověď:</strong>
' . $youranswer . '</span>' . "\n\r";
            $output .= '</p>' . "\n\r";
        } elseif ($row['type'] == 2) { // jde o no otazku
            if ($data['yesorno' . $row['id']] == "yes") {
                $youranswer = 'Ano';
            } else {
                $youranswer = 'Ne';
                $points = $points + task_points($row['id']);
            }
            $output .= '<p>' . "\n\r";
            $output .= ' <span class="trueanswer"><strong>Správná
odpověď:</strong> Ne<br /></span>' . "\n\r";
            $output .= ' <span class="youranswer"><strong>Tvoje odpověď:</strong>
' . $youranswer . '</span>' . "\n\r";
            $output .= '</p>' . "\n\r";
        } elseif ($row['type'] == 3) { // jde o cislenou odpoved
            $task_id = $row['id'];
```

```

        $db2 = mysql_query("SELECT content FROM " . ANSWERS_TABLE . " WHERE
task_id = '$task_id' LIMIT 1");
        $subrow = mysql_fetch_array($db2);
        $output .= '<p>' . "\n\r";
        $output .= ' <span class="trueanswer"><strong>Správná
odpověď:</strong> ' . $subrow['content'] . '<br /></span>' . "\n\r";
        $output .= ' <span class="youranswer"><strong>Tvoje odpověď:</strong>
' . $data['answer' . $row['id']] . '</span>' . "\n\r";
        $output .= '</p>' . "\n\r";
        if ($data['answer' . $row['id']] == $subrow['content']) {
            $points = $points + task_points($row['id']);
        }
    } elseif ($row['type'] == 4) { // jde o textovou odpoved
        $task_id = $row['id'];
        $db2 = mysql_query("SELECT content FROM " . ANSWERS_TABLE . " WHERE
task_id = '$task_id' ORDER BY id");
        $output .= '<p>' . "\n\r";
        $truth = FALSE;
        while ($subrow = mysql_fetch_array($db2)) {
            $output .= ' <span class="trueanswer"><strong>Správná
odpověď:</strong> ' . $subrow['content'] . '<br /></span>' . "\n\r";
            if ($data['answer' . $row['id']] == $subrow['content']) {
                $truth = TRUE;
            }
        }
        $output .= ' <span class="youranswer"><strong>Tvoje odpověď:</strong>
' . $data['answer' . $row['id']] . '</span>' . "\n\r";
        $output .= '</p>' . "\n\r";
        if ($truth) {
            $points = $points + task_points($row['id']);
        }
    } else { // jde o moznosti, nebo o utok mimozemstanu
        $goods = 0; // kolik zvolil spravne dobrych odpovedi
        $bads = 0; // kolik zak zvolil jako dobre, ale byli spatne
        $truths = 0; // kolik bylo spravnych moznosti
        $output .= '<p>' . "\n\r";
        $output .= ' <span class="answers_title">Odpovědi:<br /></span>' .
"\n\r";
        $task_id = $row['id'];
        $db2 = mysql_query("SELECT * FROM " . ANSWERS_TABLE . " WHERE task_id
= '$task_id' ORDER BY id");
        while ($subrow = mysql_fetch_array($db2)) {
            $output .= ' <span class="" . $optclass . "'>' .
$subrow['content'] . '<br /></span>' . "\n\r";
            if ($subrow['truth']) {
                ++$truths;
                if ($data['opt' . $subrow['id']]) { ++$goods; }
            } else {
                if ($data['opt' . $subrow['id']]) { ++$bads; }
            }
        }
        $output .= '</p>' . "\n\r";
        if ($truths == $goods && $bads == 0) {
            $points = $points + task_points($row['id']);
            $output .= '<p>Dostáváš plný počet bodů za tuto otázku.</p>' .
"\n\r";
        } elseif ($truths == 0) {
            $output .= '<p>Nedostáváš nic za tuto otázku.</p>' . "\n\r";
        } else {
            $tajl = task_points($row['id'])/$truths; // jeden dilek je
celkovy pocet bodu deleno poctem spravnych odpovedi
            $thispoints = $goods*$tajl - $bads*$tajl; // za spravnou
+dilek, za odpoved mimo spravne -dilek
            if ($thispoints < 0) { $thispoints = 0; } // nebudeme mu
uzirat body z jinych otazek
            $thispoints = round($thispoints);

```

```

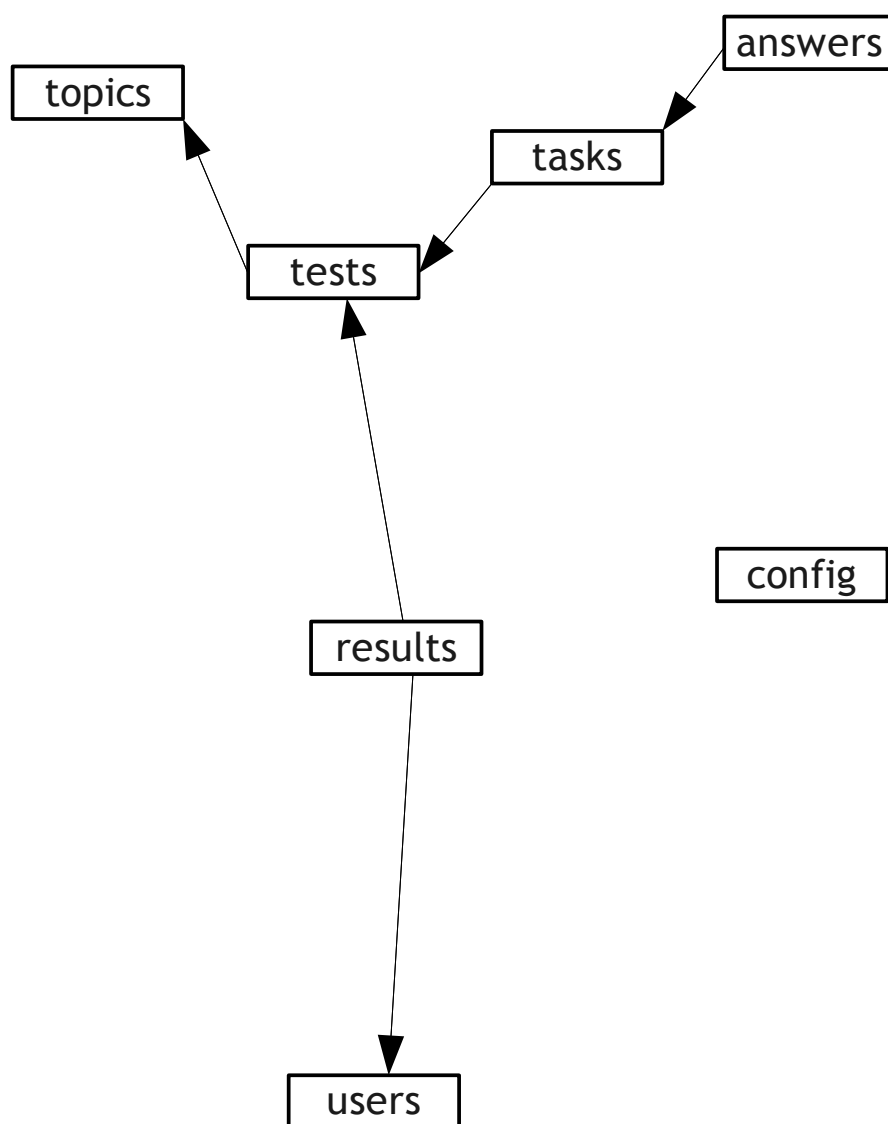
        $points = $points + $thispoints;
        $output .= '<p>Dostáváš ' . pointy($thispoints) . ' za tuto
otázku.</p>' . "\n\r";
    }
}
$output .= '</fieldset>' . "\n\r\n\r";
}
$result_value = round(($points/$maxpoints)*100); // procenta
$output .= '<p>Dostáváš ' . $result_value . '% z tohoto testu.</p>' . "\n\r";
$user_id = get_userid($data['username']);
$test_id = $data['testid'];
$timenow = mktime();

$db = mysql_query("INSERT INTO " . RESULTS_TABLE . " VALUES(0, '$user_id',
'$test_id', '$timenow', '$result_value')");

return $output;
}

```

5 Grafické vyjádření datového modelu úlohy



6 Grafické vyjádření algoritmu úlohy

Úloha bohužel nemá žádný algoritmus, který by se dal znázornit graficky. Je sledem přímých reakcí na přímé akce uživatele.

7 Uživatelská příručka

Aplikace obsahuje interní systém nápovědy, který zajistí to, že žádná příručka není potřeba. V této kapitole tedy alespoň popíšu popis instalace.

7.1 Podmínky běhu aplikace

Ke spuštění aplikace je potřeba webový server s interpretem jazyka PHP řady 4 a vyšší a databázový server MySQL řady 4 a vyšší. Na webový server je pak potřeba nahrát všechny přiložené soubory.

Testování probíhalo na PHP verzi 4.4.8 a MySQL 4.0.27. Je vysoce pravděpodobné, že aplikace poběží i na novějších verzích, není to však otestováno ani potvrzeno.

7.2 Definování přístupu do databáze

V souboru *dbconfig.php* je před instalací potřeba definovat adresu databázového serveru, název databáze a přístupové údaje. Je to jednoduché, položky jsou komentovány.

7.3 Vytvoření tabulek v databázi

Pro vytvoření tabulek v databázi lze použít soubor dotazů jazyka SQL *install.sql*, který můžete provést například pomocí aplikace phpMyAdmin. Před provedením dotazů lze upravit v souboru potřebné údaje, jako je například učitelské heslo (výchozí heslo je *root*), nebo titulek celé aplikace.

8 Seznam použité literatury

K tvorbě aplikace jsem použil jen své vlastní znalosti a rady přátel.

9 Závěr

Vlastní zhodnocení práce. Zde sepíšu klady a zápory aplikace a náměty na její eventuální rozšíření, nejasnosti a problémy, se kterými jsem se setkal při tvorbě aplikace, možné změny v řešení a další postřehy, ke kterým jsem došel v průběhu vývoje aplikace.

9.1 Klady aplikace

Hlavním kladem aplikace je její celková jednoduchost a nerobustnost. Nechodím s pověstným kanónem na vrabce, takže je aplikace malá, rychlá a flexibilní.

Za další výhodu může být v našem prostředí považována čeština, kterážto je jazykem, kterým aplikace komunikuje s uživatelem.

Jakožto zastánce otevřeného a svobodného řešení také zdůrazňuji mezi výhodami fakt, že tento program je svobodný software a můžete jej dále distribuovat a/nebo upravovat pod podmínkami licence GNU General Public License⁵ vydané organizací Free Software Foundation, verze licence 2 nebo vyšší (dle vaší volby).

9.2 Zápory aplikace

Vzhledem k faktu, že aplikace nebyla nikdy nasazena v ostrém provozu, mezi zápory mohu zařadit její neozkoušenost ze stránky bezpečnosti.

9.3 Náměty na eventuální rozšíření

9.3.1 Statistika

Aplikaci chybí komplexní statistika, která by vhodnými dotazy na databázi vytvářela seznamy nejlepších žáků, nejobtížnějších testů a podobně.

9.3.2 Lepší systém přihlašování uživatelů

V současné době není přihlašování žáků řešeno příliš elegantně, respektive k žádnému přihlášení vůbec nedochází, lepší by bylo, aby se žák mohl přihlásit a poté po aplikaci „brouzdat“.

9.4 Postřehy

Aplikaci jsem psal ve dvou fázích s celkem dlouhou přestávkou. V takové chvíli jsem si uvědomil, že komentáře jsou cennější nežli sůl i zlato, protože i ve vlastním nekomentovaném zdrojovém kódu je těžké se vyznat.

Proto všem doporučuji: Komentovat, komentovat, komentovat...

⁵ <http://www.gnu.org/copyleft/gpl.html>

10 Oponentský posudek

Předvedl jsem své dílo, včetně rozpracované dokumentace, třetí (nezávislé a ne zcela laické) osobě a požádal jsem ji o písemné zhodnocení aplikace hlavně z pohledu uživatele.

10.1 Představení autora posudku

Posudek pro mě sestavil Milan Drozd, můj spolupracovník z projektu *Neverhood.Etomite.cz*. Je to student z Dubnice nad Váhem. Posudek jsem přeložil do češtiny.

10.2 Posudek

Aplikace, kterou mi Miro představil, je jednoduchá na ovládání. Již před přečtením dokumentace jsem přesně věděl jak s ní zacházet, co kde udělat.

Její rozhraní je jednoduché a bez zbytečností. Vytvořil jsem několik testů, sám jsem je poté zkusil vyplnit. Okamžité ohodnocení testu v procentech je velice přívětivé.

Během testování jsem bohužel narazil na drobnou logickou chybu. Zatímco jsem jakožto učitel tvořil test, mohl jsem ho jako žák vyplňovat, ačkoli měl zatím pouze jednu otázku, jednoduše jsem tak mohl získat 100% a učitel na to nemusel přijít.

Tento problém by šlo řešit položkou v databázi, která by určovala, jestli je test hotov, tedy jestli jej mohou žáci začít vyplňovat.

Kromě toho mě dost otravovala neustálá nutnost opakovaně zadávat učitelské heslo a nemožnost se pomocí přihlášení přepnout do učitelského režimu.

Dokumentace je velmi pěkně strukturovaná, dá se v ní dobře orientovat. Při čtení jsem pochopil, jak co funguje, někdy jsem se i pobavil.