



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

Bakalářská práce

# Online aplikace pro odhalování podvádění v soutěži Bobřík informatiky

Vypracoval: Martin Candra  
Vedoucí práce: Mgr. Václav Šimandl

České Budějovice 2015

**ZADÁNÍ BAKALÁŘSKÉ PRÁCE**  
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin CANDRA**  
Osobní číslo: **P120260**  
Studijní program: **B7507 Specializace v pedagogice**  
Studijní obor: **Informační technologie a e-learning**  
Název tématu: **Online aplikace pro odhalování podvádění v soutěži Bobřík informatiky**  
Zadávající katedra: **Katedra informatiky**

**Z á s a d y p r o v y p r a c o v á n í :**

Student vytvoří webovou aplikaci, která bude umožňovat odhalování podvádění tzv. úspěšných řešitelů v soutěži Bobřík informatiky. Aplikace bude vytvořena v programovacím jazyce PHP. Student analyzuje možné jevy, které mohou ukazovat na podvádění soutěžícího v soutěži (např. krátký čas soutěžení jedince oproti průměru, zasílání stejných odpovědí soutěžícími z jedné školy ve stejný čas), a na jejich základě navrhne a naprogramuje metody, jak podezřelé soutěžící odhalovat. Výsledkem činnosti každé z metod bude seznam podezřelých soutěžících, přičemž tyto seznamy aplikace dále porovná a číselně vyjádří, nakolik je podvádění u jednotlivých soutěžících pravděpodobné. Aplikace uloží podezřelé soutěžící, včetně informace o jejich přítomnosti v jednotlivých seznamech, do nové databázové tabulky, odkud je bude na vyžádání schopna načíst podle různých kritérií (např. podle soutěžní kategorie nebo školy).

Rozsah grafických prací: CD ROM

Rozsah pracovní zprávy: 40

Forma zpracování bakalářské práce: tištěná

Seznam odborné literatury:

CIZEK, G. J. *Detecting and Preventing Classroom Cheating*. Thousand Oaks: Corwin Press, 2003. 167 s. ISBN 0-7619-4655-1.

BLISS, T. J. *Statistical Methods to Detect Cheating on Tests: A Review of the Literature* [online]. 2012 [cit. 2013-12-06]. Dostupné z:

[http://www.ncbex.org/assets/media\\_files/Research/2012Bliss.pdf](http://www.ncbex.org/assets/media_files/Research/2012Bliss.pdf)

ROWE, N. C. *Cheating in Online Student Assessment: Beyond Plagiarism*. In: *Distance Learning Administration*. 2004, roč. 7, č. 2. Dostupné z:

<http://www.westga.edu/distance/ojdla/summer72/rowe72.html>

TOMCSÁNYI, P. a J. VANÍČEK. *Implementation of informatics contest Bebras in Czechia and Slovakia*. In: *Information and communication technology in education. Proceedings*. Ostrava: Ostravská univerzita, 2009. s. 214-218. ISBN 80-7368-459-4.

VANÍČEK, J. *Rozvoj inforatických kompetenci žáků pomocí soutěže: kauza Bobřík informatiky* [online]. In: *Sborník konference Počítač ve škole. Nové Město na Moravě: Gymnázium Vincence Makovského, 7.-9. 4. 2009* [cit. 2013-11-13]. Dostupné z: [http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/vanicek\\_pdf\\_43005.pdf](http://www.pocitacveskole.cz/system/files/uzivatel/9/clanky/vanicek_pdf_43005.pdf)

LECKY-THOMPSON, E. a S. D. NOWICKI. *PHP 6: programujeme profesionálně*. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 718 s. ISBN 978-80-251-3127-5.

BORONCZYK, T.. *PHP 6, MySQL, Apache: vytváříme webové aplikace*. Vyd. 1. Brno: Computer Press, 2009. ISBN 978-80-251-2767-4.

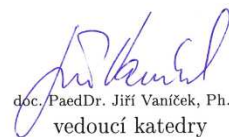
Vedoucí bakalářské práce: **Mgr. Václav Šimandl**  
Katedra informatiky

Datum zadání bakalářské práce: **27. března 2014**

Termín odevzdání bakalářské práce: **30. dubna 2015**



Mgr. Michal Vančura, Ph.D.  
děkan



doc. PaedDr. Jitka Vaníček, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 27. března 2014

## **Prohlášení**

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 - v úpravě vzniklé vypuštěním vyznačených částí archivovaných pedagogickou fakultou 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.

V Českých Budějovicích 26. 6. 2015

---

Martin Candra

## **Abstrakt**

Cílem této bakalářské práce je v programovacím jazyce PHP vytvořit webovou aplikaci, která umožní pověřeným osobám zjednodušené odhalení podvádějících soutěžících v soutěži Bobřík informatiky.

Pro vytvoření této aplikace bylo potřeba analyzovat jevy, které mohou ukazovat na podvádění soutěžících. Na základě těchto konkrétních jevů byly vytvořeny metody, které vyhledávají a ukládají seznamy soutěžících, kteří by mohli podvádět. Dále se tyto seznamy porovnávají a na základě údajů v těchto seznamech se číselně vyhodnotí, jak je pravděpodobné, že účastník soutěže podváděl. Podezřelí soutěžící jsou uloženi do databázové tabulky spolu s údaji, ve kterých seznamech se soutěžící nachází. Z této databázové tabulky si může pověřená osoba moci načíst úplný seznam podezřelých účastníků nebo část seznamu podle zadaných kritérií jako například kategorie, rok, škola atd.

## **Klíčová slova**

Bobřík informatiky, odhalování podvádění, webová aplikace, PHP

## **Abstract**

The aim of this bachelor's work is to create a web application using PHP language. It will enable authorized persons to detect cheating individuals in the competition Bebras contest.

It is needed to analyze phenomenons which can point to cheating by contestants. I will create methods based on these phenomenons which will search and save lists of contestants who are suspected of cheating. Further more these lists will be making comparisons among themselves and they will numerically evaluate the probability of cheating of contestant based on the data. Suspicious-looking contestants will be saved into database table together with data in which lists the suspect is placed. Authorized person will be able to load complete list of suspicious participants or its part according to entered criterion from this database.

## **Keywords**

Bebras contest, detection of cheating, web application, PHP

## **Poděkování**

Tímto bych chtěl velice poděkovat vedoucímu mé bakalářské práce Mgr. Václavu Šimandlovi. za přínosné rady, připomínky, ochotu a odborné vedení při zpracovávání mé bakalářské práce.

## Obsah

1	Úvod.....	10
2	Bobřík informatiky .....	11
2.1	Kategorie .....	11
2.2	Test.....	11
2.3	Koordinátor .....	12
2.4	Ukládání dat .....	12
3	Možnosti podvádění.....	14
3.1	Tradiční podvádění.....	14
3.1.1	Získávání, poskytování a vyžadování informací .....	14
3.1.2	Nepovolené materiály, plagiátorství.....	14
3.1.3	Ovlivňování průběhu a hodnocení zkoušky.....	15
3.2	Elektronické podvádění.....	15
3.2.1	Získávání, poskytování a vyžadování informací .....	15
3.2.2	Nepovolené materiály, plagiátorství.....	16
3.2.3	Ovlivňování průběhu a hodnocení zkoušky.....	16
4	Softwarové odhalování podvádění.....	18
4.1	Zahrnuté způsoby podvádění .....	18
4.1.1	Samostatné rizikové jevy .....	19
4.1.2	Hromadné rizikové jevy.....	19
4.2	Návrh implementace odhalování rizikových jevů.....	20
4.3	Vyhodnocení pravděpodobnosti podvádění.....	22
4.4	Vytvoření samotné aplikace .....	22
4.5	Nastavení aplikace .....	23
4.5.1	Nastavení rozsahů .....	23
4.5.2	Nastavení vah.....	25
4.5.3	Spojení s databází .....	25
4.5.4	Nastavení tabulek.....	26
4.6	Spuštění programu .....	26
4.6.1	Webový prohlížeč .....	26
4.6.2	Bash .....	27
4.7	Doba běhu aplikace .....	28
5	Zobrazovací stránka.....	30
5.1	Vytvoření nové stránky .....	30



5.1.1	Vytvoření pohledu .....	30
5.1.2	Zobrazení dat z vytvořeného pohledu.....	33
5.2	Zobrazované sloupce dat.....	34
5.3	Filtrování dat .....	35
6	Testování.....	37
6.1	Simulace.....	37
6.1.1	Samostatné identifikátory .....	37
6.1.2	Hromadné identifikátory.....	38
6.1.2.1	Krátký průměrný čas školy.....	38
6.1.2.2	Lepší kategorie na škole .....	39
6.1.2.3	Zlepšující se výsledky .....	40
6.2	Porovnání výsledků ostré soutěže .....	41
6.2.1	Ročník 2013 .....	41
6.2.2	Ročník 2014.....	44
7	Závěr .....	46
	Seznam použité literatury a zdrojů .....	47
	Seznam obrázků.....	49
	Seznam grafů .....	50
	Seznam tabulek .....	51

# 1 Úvod

Tato bakalářská práce je zaměřena na odhalování podvádějících soutěžících v informatické soutěži Bobřík informatiky. Cílem bakalářské práce bylo vytvořit online aplikaci, která by dokázala odhalit soutěžící, kteří při skládání soutěžního testu podváděli.

Nejprve bylo potřeba seznámit se s pravidly a fungováním soutěže. Dále bylo třeba provést analýzu způsobů, jakými může soutěžící v této soutěži podvádět. Z těchto způsobů byly vybrány ty, které půjdou odhalovat zpětně z databáze soutěže. K těmto způsobům byly navrženy jejich softwarová řešení a podle tohoto návrhu byly následně naprogramovány jednotlivé metody a tyto metody byly zahrnuty do aplikace.

V druhé fázi byl do administračního prostředí testu Bobříka informatiky vytvořen modul pro zobrazování výstupů aplikace. Díky tomuto modulu je možné zobrazovat, řadit, filtrovat a exportovat jednotlivé podezřelé soutěžící, kteří byli označeni aplikací.

Dále je pro správce popsáno, jak aplikaci napojit na databázi, jak ji spouštět a jak dlouho bude zpracovávání dat aplikací přibližně trvat. Také je uvedeno, kam na server je potřeba umístit jednotlivé soubory modulu a aplikace.

## 2 Bobřík informatiky

Bobřík informatiky je soutěž, do které se zapojují střední a základní školy druhého stupně již od roku 2008. Jedná se o soutěž, kde každý žák soutěží sám za sebe. Soutěží se formou online testu, který je zaměřený na, jak už z názvu vyplývá, informatické znalosti. Soutěž není jednostranně zaměřena na programování ani na uživatelský přístup. Tato soutěž patří k mezinárodní skupině a je sdružená pod názvem Bebras s informatickými soutěžemi v desíctce evropských států. [1]

### 2.1 Kategorie

Soutěžící jsou rozděleni do kategorií podle toho, jaký ročník školy navštěvují. Kategorie Mini je určena pro čtvrté a páté ročníky základních škol. V této kategorii test obsahuje 12 otázek a čas na jeho splnění je třicet minut.

V kategorii Benjamin soutěží šesté a sedmé ročníky základních škol (u osmiletého gymnázia jsou to prima a sekunda). Další kategorií je Kadet, které se účastní osmé a deváté ročníky základních škol (na osmiletém gymnáziu tercie a kvarta). Kategorie Junior je pro první a druhé ročníky středních škol (kvinta a sexta na osmiletém gymnáziu). Poslední kategorií je Senior. Tato kategorie je určena pro třetí a čtvrté ročníky středních škol (septima a oktáva na osmiletém gymnáziu). U těchto čtyř kategorií test obsahuje 15 otázek, na které mají soutěžící 40 minut. [2]

### 2.2 Test

Test probíhá na školních počítačích formou online testu s využitím webového prohlížeče na stránkách soutěže ([www.ibobr.cz](http://www.ibobr.cz)). Nejdříve se musí účastník registrovat. K registraci je potřeba tajný registrační kód, který studentům sdělí školní koordinátor nebo dozorující učitel. Registrací student získá přístup k samotnému testu. Soutěžící odpovídá na jednotlivé otázky, které jsou seřazeny náhodně. Při zadání chybné odpovědi se odečítají body, proto je k dispozici možnost na otázku neodpovídat, což umožňuje soutěžícímu předejít ztrátě bodů. Ke každé otázce se lze vrátit později a upravit její odpověď. Test je možné kdykoliv předčasně ukončit. Není povolena žádná nápověda. Při plnění testu musí být ve třídě dozor, který kontroluje, zda spolu soutěžící nespolupracují nebo zda nemají nějaké nepovolené materiály. Není možné, aby někdo plnil test bez řádného dozoru. V testovacím dnu není povoleno jednomu a témuž studentovi absolvovat test vícekrát. [2] Na školách testování probíhá po skupinách

(většinou po třídách) z důvodu menší kapacity počítačových učeben. To znamená, že soutěžící neplní test hromadně, ale postupně v průběhu dne. [13]

Z výsledků testů se vyhlásují takzvaní úspěšní řešitelé. To jsou soutěžící, kteří dosáhli minimálního bodového zisku v dané kategorii. [13]

## 2.3 Koordinátor

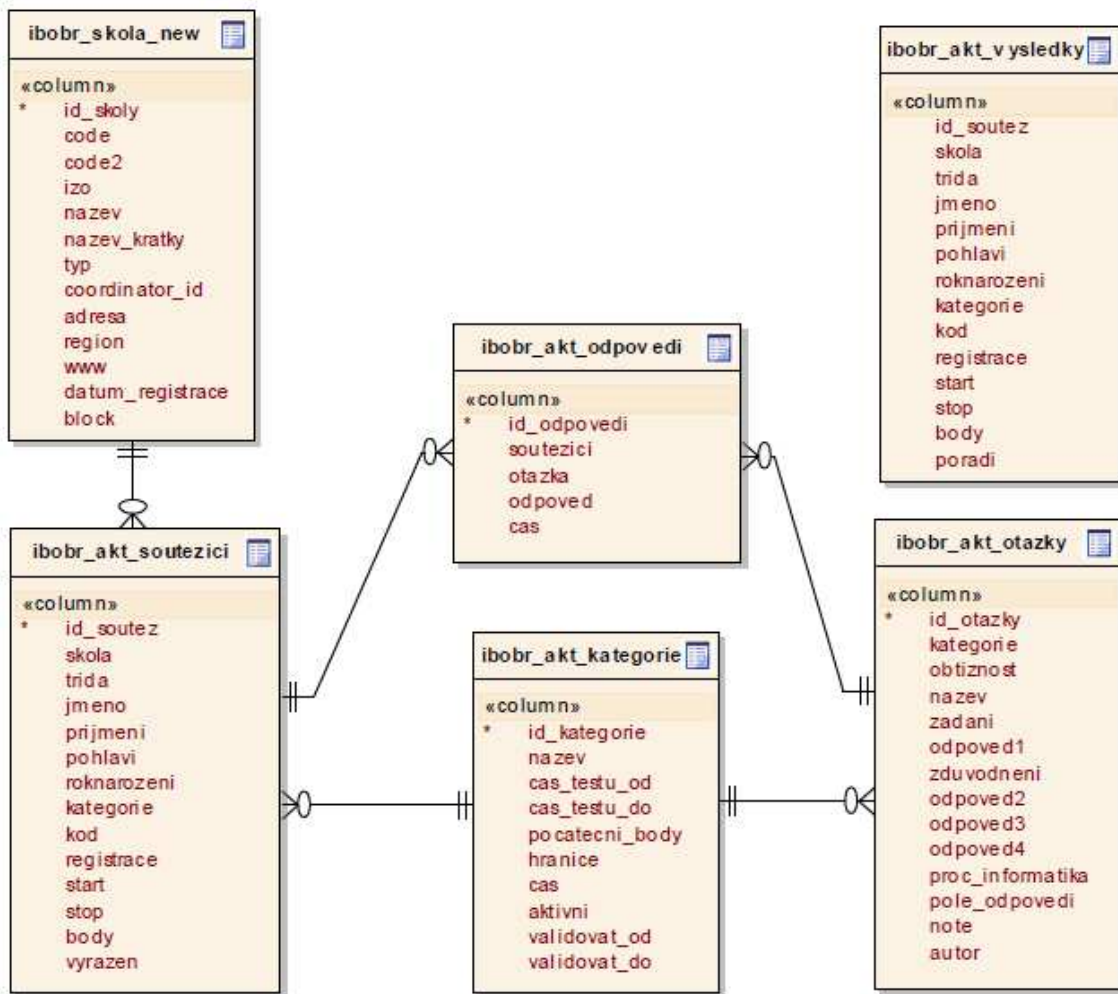
Koordinátor je osoba, která svou školu registruje a přihlašuje do každého ročníku soutěže, zajišťuje průběh soutěžního dne na škole, potvrzuje seznam soutěžících, vyřazuje soutěžící, kteří nesplnili podmínky nebo pravidla soutěže a také komunikuje s organizátory soutěže. [1]

## 2.4 Ukládání dat

Veškeré informace o školách, kategoriích, výsledcích, soutěžících a jejich odpovědích se ukládají do MySQL<sup>1</sup> databáze. Důležité tabulky můžete vidět na obrázku 1, jedná se o tabulky z aktuálního ročníku soutěže. Do tabulky `ibobr_skola_new` se ukládají data, jako například název, adresa, typ, region, osobní číslo koordinátora školy a mimo jiné také kódy pro registraci soutěžících. V tabulce `ibobr_akt_kategorie` se nacházejí například tato data: číslo kategorie, název, datum a čas, od kdy je možné v této kategorii soutěžit, datum a čas, do kdy je v této kategorii možné soutěžit, bodová hranice (podle té se určuje, zda byl soutěžící úspěšný nebo ne), časový limit pro složení testu, a další. Do tabulky `ibobr_akt_soutezici` se ukládají informace o registrovaných účastnících soutěže. Mezi tyto informace patří přidělené osobní číslo, škola, pod kterou se registroval, třída, jméno a příjmení, kategorie, ve které soutěží, datum a čas registrace, datum a čas spuštění testu, datum a čas ukončení testu, získané body a informace o tom, zda byl vyřazen školním koordinátorem. V tabulce `ibobr_akt_odpovedi` jsou uchovávány informace o odpovědích, tedy který soutěžící ji zvolil, k jaké otázce náleží, číslo zvolené odpovědi a přesný čas jejího odeslání. Po vyhodnocení soutěže je do databáze přidána tabulka `ibobr_akt_vysledky`, ve které jsou stejné informace jako u tabulky `ibobr_akt_soutezici`, informace o vyřazení je zde však nahrazena umístěním soutěžících. Vyřazení účastníci již v této tabulce uvedeni nejsou.[12]

---

<sup>1</sup> MySQL - databázový systém, oficiální stránky: [www.mysql.com](http://www.mysql.com)



Obrázek 1: Diagram důležitých stávajících tabulek soutěže

### 3 Možnosti podvádění

Mareš v časopise *Pedagogika* sepsal různé způsoby podvádění a rozlišil je do dvou kategorií. Tyto kategorie nazval tradiční podvádění ve škole a elektronické podvádění. Obě kategorie rozdělil podle typu podvádění a k nim vypsals konkrétní postupy.

#### 3.1 Tradiční podvádění

##### 3.1.1 Získávání, poskytování a vyžadování informací

Mezi tento typ podvádění patří tyto konkrétní postupy:

- Dát opsat domácí úkol spolužákovi. Opsat domácí úkol od spolužáka.
- Opsat test od spolužáka při písemném zkoušení. Student využije zmatku a získá dva testové formuláře, zatímco do jednoho píše odpovědi, druhý předá dalším třídám.
- Napovídat spolužákům při písemném zkoušení pomocí neverbální komunikace.
- Předávat informace mezi dvěma spolužáky při písemném zkoušení.
- Po rozsazení si sednout co nejdále od učitele, pak se přihlásit s tím, že něčemu nerozumím a po cestě k učiteli nahlížet do testů svých spolužáků.
- Napovídat při ústním zkoušení. Získávat informace o testu od studentů, kteří ho již absolvovali. [3]

##### 3.1.2 Nepovolené materiály, plagiátorství

K tomuto typu podvádění Mareš uvádí tyto konkrétní postupy:

- Opisovat z jakéhokoliv zdroje při písemné zkoušce, když je používání pomůcek zakázáno.
- Opisovat text z určitého zdroje, aniž by byl tento zdroj citován. Parafrázovat text z určitého zdroje, aniž by byl tento zdroj citován.
- Použít opakovaně vlastní seminární práci, esej a tak podobně a vydávat ji za práci novou.
- Komukoliv nabízet k znovu použití hotovou práci.
- Opsat od spolužáka jeho práci s jeho souhlasem a vydávat ji za práci svou.

- Opsat od spolužáka jeho práci, aniž by o tom on věděl, a vydávat ji za svou práci.
- Upravovat naměřená data tak, aby žák dospěl k žádaným nebo statisticky významným výsledkům.
- Vymýšlet si údaje. Uvádět odkazy na prameny, které neexistují. [3]

### 3.1.3 Ovlivňování průběhu a hodnocení zkoušky

Konkrétní postupy:

- Při testu využít zmatek při odevzdávání a test neodevzdat, poté co jsou vyhlášovány výsledky tvrdit, že test byl odevzdán a tím se domoci opakování testu.
- Žádat někoho, aby za mne složil zkoušku.
- Jít za někoho složit zkoušku. Lhát o tom, že byly splněny podmínky k připuštění ke zkoušce.
- Lhát o zdravotních nebo sociálních problémech tak, aby byly získány mimořádné ohledy při zkoušení.
- Získat výhodu nabízením úplatku či protislužby.
- Schovat materiály z knihovny tak, aby si je ostatní studenti nemohli prostudovat.
- Podvodem získat informace o tom, co se bude zkoušet.
- Zatajit učitelovo přeslechnutí nebo chybu při písemné zkoušce.
- V již opraveném testu opravit chybné odpovědi na správné a vyžadovat změnu klasifikace. [3]

## 3.2 Elektronické podvádění

### 3.2.1 Získávání, poskytování a vyžadování informací

Konkrétní postupy:

- Pokud mají všichni studenti stejné znění testu, pak je výhodné si v počítačové učebně sednout vedle spolužáka, který je lepší. Horší spolužák od něj opisuje správné odpovědi. [4]
- Posílat správné odpovědi pomocí mobilního telefonu svým spolužákům.
- Opsat odpověď přijatou od spolužáka na mobilní telefon.
- Nechat si napovídat pomocí miniaturní vysílačky.

- Předávat informace o obsahu a podobě testu prostřednictvím fotoaparátu v mobilním telefonu. [3]
- Předávat informace o obsahu a podobě aktuálně psaného testu prostřednictvím "zkopírování obrazovky". Jde o uložení textu pro použití anebo odeslání textu spolužákovi. [4]
- Nechat si napovídat při počítačovém zkoušení pomocí ukrytého hands-free mobilu. [5]

### 3.2.2 Nepovolené materiály, plagiátorství

Konkrétní postupy:

- Opisovat při testu na počítači z elektronického přístroje. [6]
- Opisovat při písemném zkoušení pomocí zvukového zařízení, pokud hrozí nebezpečí odhalení, přepne student na hudbu. [7]
- Použít předem připravenou práci z přenosného nosiče a vydávat ji za práci vypracovanou při zkoušení.
- Opisovat při zkoušení tak, že si student předem připraví soubor do systému, ten pak vhodně přejmenuje a uloží na požadované místo. Tuto práci poté vydává za práci vypracovanou při zkoušení. [8]
- Stáhnout si práci vytvořenou jiným autorem z internetu a vydávat za práci vlastní. [9]
- Zkopírovat text z určitého zdroje, aniž by byl zdroj textu citován. [9]
- Naskenovat text nebo obrázek z knihy a převést do elektronické podoby, nebo zkopírovat text, obrázek nebo graf do své práce a vše vydávat za své dílo, aniž by byl citován původní zdroj. [8], [9]
- Zkopírovat text v cizím jazyce z určitého zdroje, aniž by byl zdroj citován, přeložit ho a vydávat za text vlastní.
- Napsat text v mateřském jazyce, ten následně přeložit do cizího jazyka. Výsledek vydávat za práci dokládající studentovu jazykovou kompetenci při zkoušení v cizím jazyce. [9]

### 3.2.3 Ovlivňování průběhu a hodnocení zkoušky

Konkrétní postupy:

- Přihlásit se do systému za někoho jiného a složit za něj test. [4]



- Přepnout počítač do režimu sdílení s jiným počítačem, pak jen předstírat činnost. Mezitím někdo jiný řeší zadané úlohy.
- Při distančním zkoušení se do systému přihlásit pod svými údaji, ale mít vedle sebe někoho, s kým se zkoušený radí na správných odpovědích. [4]
- Při zkoušení žádat o pomoc pomocí elektronické diskusní skupiny, poté některé části textu zkopírovat a výsledný text vydávat za vlastní. [8]
- Pomocí crackingu prolomit heslo, přihlásit se s ním do systému a tak získat přístup ke klíči testu. [4]
- Podvádět za účelem získat učitelovo heslo. Např. vydávat se za operátora systémové podpory a žádat o učitelovo heslo. [4]
- Přihlásit se do systému jako učitel a přečíst si klíč správných odpovědí. [4]
- Nahrát do systému spyware, monitorovat aktivitu učitele nebo spolužáků a tím získávat informace jak správně postupovat při odpovídání. [4]
- Útočit na ochranu systému, tím získat práva a přístup do zabezpečených míst systému. Následně získat utajované informace nebo pozměnit školní data. [4]
- Při psaní jednotného testu ve stejný den využít časového posunu a na internetu dohledat znění testu. Příklad: Studenti z Anglie, kteří psali test v Honkongu, umístili jeho znění na internet pro studenty, kteří ho psali ve stejný den v Anglii. [3]

## 4 Softwarové odhalování podvádění

Organizátoři online soutěží jsou v nesnadném postavení. Ačkoli mají zodpovědnost za výsledky soutěže, není v jejich silách uhlídat, zda nikdo nepodváděl, jelikož při skládání testů ve školách nejsou fyzicky přítomni. Dobré výsledky studenta v soutěži přitom přinášejí výhody, jelikož jsou zohledňovány u přijímacích zkoušek či v programu ministerstva školství, mládeže a tělovýchovy Excellence [13]. Tento úkol tedy spadá na bedra osob, které jsou u testu dozorem. Jejich přístup k podvádění však může být velice rozdílný. V momentě, kdy by byl student ze soutěže vyřazen, má dozor možnost jej pouze slovně napomenout a nechat soutěžit dál. Z tohoto důvodu bylo zapotřebí vytvořit aplikaci, která bude na základě objektivních dat odhalovat podvádějící účastníky.

Pro odhalování podvádějících soutěžících byla tedy vytvořena aplikace, která za pomoci tzv. dolování dat ze strukturovaných dat, jež jsou generována v průběhu soutěže, tyto soutěžící odhalí na základě v této kapitole zmiňovaných jevů a zapíše do databázové tabulky. Tuto aplikaci spustí správce aplikace, a to po ukončení jejího aktuálního ročníku. Dále je potřeba uložená data nějakým způsobem zobrazit. Z tohoto důvodu byla do administračního prostředí testu přidána stránka (viz kapitola 5), na které se tato data dají kdykoliv zobrazit. Tato stránka je však k dispozici jen k tomu oprávněným uživatelům.

### 4.1 Zahrnuté způsoby podvádění

Nejprve bylo pro vytvoření aplikace potřeba zjistit, jakými způsoby je možné v online testu soutěže Bobřík informatiky podvádět. Problematikou podvádění při online soutěžích se zabývá článek Mgr. Šimandla v časopise *JTIE* [10].

V tomto článku uvedl jevy, které upozorňují na zvýšenou pravděpodobnost, že došlo k podvádění. Tyto jevy byly rozděleny do dvou skupin, podle počtu podezřelých účastníků, a to na samostatné (individuální), kdy je podvádějícím jedinec, a na hromadné, kdy se v roli podvádějícího vyskytuje celá skupina. V následujících dvou podkapitolách uvedu, které jevy byly zahrnuty do vytvářené aplikace.

### 4.1.1 Samostatné rizikové jevy

Prvním z jevů, které byly použity v aplikaci, bylo sledování počtu shodných odpovědí, a to jak správných, tak chybných. Tato situace vypovídá o opisování mezi jednotlivými účastníky soutěže v rámci jedné skupiny. Původní aplikace se v tomto ohledu zaměřovala pouze na plný počet testových odpovědí, tedy 15. Tato aplikace však zkoumá i 14 a 13 shodných odpovědí. Domnívám se, že někteří soutěžící ze strachu z odhalení při opisování odpovědí od spolužáka, schválně označí odpověď na jednu či dvě otázky jinak. Magistr Šimandl ve svém článku tento jev označil písmenem **S**.

Jiným faktorem, kterému bylo třeba věnovat pozornost, byl čas. Máme k dispozici čas registrace, spuštění testu, odeslání jednotlivých otázek, ukončení testu. Tyto hodnoty nám navíc umožňují vypočítat průměrný čas skládání testu, který je klíčovým pro jev, který je rovněž důležité sledovat. Jedná se o situaci, kdy je čas vypracování testu jedincem výrazně kratší než průměrný čas. V takovém případě lze uvažovat například o opisování z taháku, opakování testu, popřípadě se může jednat o souvislost s jinými rizikovými jevy (práce ve dvojici, absence dozoru...). Písmenem označujícím tento jev je **R**.

Dále je třeba zohlednit možnost, že absentující student získá tajný registrační kód od spolužáků či dokonce od vyučujícího a zúčastní se soutěže bez dozoru, s volným přístupem k informacím. Rovněž je zde možnost, že mu bude asistovat jeden či více spolužáků. Tento jev lze odhalit podle malého počtu osob, které začaly soutěžit ve stejnou dobu. Tento jev značíme písmenem **M**.

Posledním z individuálních rizikových jevů je prodleva. Může se vyskytovat mezi registrací a spuštěním testu. Během této doby může student spolupracovat se spolužáky, kteří již test spustili, a tak již znají otázky, aniž by sám ztrácel čas, který se odpočítává až po spuštění testu. Písmenem označujícím tento jev je **P**.

### 4.1.2 Hromadné rizikové jevy

Prvním faktorem, který jsme sledovali v rámci hromadných rizikových jevů, byl obdobný jako u samostatných, zabývali jsme se s ním však v rámci skupin. Je jím průměrný čas skládání testu u žáků dané školy v určité kategorii. Pokud je výrazně kratší, než celkový průměrný čas pro kategorii, je možné, že žáci spolupracovali, popřípadě že se do spolupráce zapojil i sám dozorující učitel. Tento jev značíme písmenem **K**.

Druhým faktorem je průměrný bodový zisk žáků dané školy v kategoriích. Pokud je tento u jedné z kategorií značně vyšší než u ostatních, je zde možnost, že při skládání testu této kategorie došlo ke spolupráci mezi žáky, popřípadě i dozorujícího učitele. Tento rizikový jev značíme písmenem **L**.

Posledním hromadným rizikovým jevem, jehož odhalení bylo třeba zajistit, bylo postupné zlepšování výsledků skupin dané školy v určité kategorii. Pokud se stejné kategorie účastní postupně několik skupin studentů, vzniká zde možnost, že studenti, kteří již test ukončili, sdělí obsah testu další skupině, která bude test skládat později, a ta bude mít vyšší bodový zisk. Písmenem označujícím tento rizikový jev je **Z**.

## **4.2 Návrh implementace odhalování rizikových jevů**

Identifikátor S byl řešen takto: Nejprve se zaznamenávají všechny odpovědi nejen všech takzvaných úspěšných řešitelů soutěže školy v kategorii, ale také soutěžící, kteří byli blízko tomu stát se úspěšným řešitelem soutěže (jedná se o soutěžící se ziskem o maximálně 20 bodů nižším než je hranice úspěšnosti), do vícerozměrného pole, a to vždy osobním číslem soutěžícího zapsaného do pole určujícího konkrétní otázku a konkrétní odpověď. To znamená, že ve vnitřních polích se nacházejí vždy soutěžící, kteří zvolili stejnou odpověď na danou otázku. Poté se toto vícerozměrné pole prochází a počítá se, kolikrát se určitá dvojice soutěžících nachází ve stejném vnitřním poli. Pokud je tento počet vyšší nebo roven 13 zjišťuje se, zda tito soutěžící začali s vyplňováním testu nejvýše s 5 minutovým rozdílem, nebo zda jejich rozdíl ukončení testu byl nejvýše 10 minut. Pokud tomu tak bylo, zaznamenají se do výsledné tabulky jen soutěžící, kteří získali vyšší nebo rovný počet bodů než je bodová hranice pro danou kategorii.

Identifikátor R byl řešen takto: Nejprve se vypočítá průměrný čas soutěžení úspěšných řešitelů soutěže v dané kategorii. Tento průměrný čas je počítá z času spuštění testu a z času odeslání poslední odeslané odpovědi. Takto vypočítaný čas by nám měl poskytnout skutečný čas, který soutěžící věnoval k vyplňování testu. Dále se stejným způsobem vypočítá průměrný čas skládání testu zkoumaného úspěšného řešitele soutěže. Tyto dva časy se spolu porovnají, a pokud průměrný čas skládání testu zkoumaného soutěžícího je nejméně o 5 minut kratší než průměrný čas skládání testu všech úspěšných řešitelů v kategorii zaznamená se zkoumaný úspěšný řešitel soutěže do výsledné tabulky.

Identifikátor M byl řešen takto: V tomto případě byli systematicky kontrolováni všichni soutěžící z dané školy a kategorie. U toho to jevu bylo potřeba kontrolovat čas začátku soutěžení. V rozsahu 10 minut před začátkem a 10 minut po začátku spuštění testu u zkoumaného soutěžícího byli zaznamenáváni ostatní soutěžící, kteří po tuto dobu také spustili test. Pokud v této době spustili svůj test nejvýše 4 soutěžící (včetně zkoumaného soutěžícího), byl zkoumaný soutěžící zapsán do výsledné tabulky pouze v případě, že se stal úspěšným řešitelem soutěže.

Identifikátor P byl řešen takto: U této metody byl kontrolováni pouze úspěšní řešitelé soutěže. U těchto soutěžících se vypočítal rozdíl času spuštění testu a času registrace do soutěže. Pokud tento rozdíl byl vyšší jak 10 minut zaznamená se tento úspěšný řešitel soutěže do výsledné tabulky.

Identifikátor K byl řešen takto: Nejprve se vypočítá průměrný čas skládání testu všech úspěšných řešitelů soutěže v dané kategorii. Dále se vypočítá průměrný čas skládání testu všech úspěšných řešitelů soutěže na zkoumané škole v dané kategorii. Tyto dva časy se poté porovnají. Pokud je průměrný čas skládání testu na zkoumané škole nejméně o 5 minut kratší než průměrný čas skládání testu všech úspěšných řešitelů soutěže v kategorii, zapíše se úspěšní řešitelé soutěže ze zkoumané školy v kategorii do výsledné tabulky.

Identifikátor L byl řešen takto: Nejdříve se vypočítají bodové průměry všech věkových kategorií, kterých se účastnilo nejméně 10 soutěžících, na zkoumané škole. Dále se tyto bodové průměry porovnají. Pokud jedna kategorie měla bodový průměr o více jak 25 bodů vyšší než všechny ostatní kategorie, zapíše se všichni úspěšní řešitelé soutěže z této kategorie do výsledné tabulky.

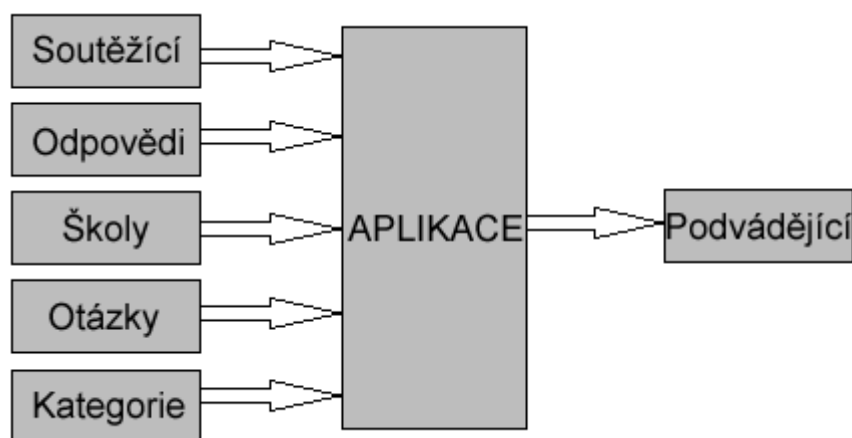
Identifikátor Z byl řešen takto: Jako první v této metodě proběhne seřazení všech soutěžících ze zkoumané školy a kategorie, která měla v dané kategorii nejméně 12 soutěžících. Soutěžící se seřadí vzestupně podle času spuštění testu. Poté se soutěžící rozdělí na dvě skupiny. Z první poloviny soutěžících se stane skupina kontrolní. Druhá polovina soutěžících se rozdělí ještě na třetiny (tj. šestiny původní skupiny soutěžících), tyto tři skupiny se stanou skupinami prověřovanými. Pro každou skupinu (skupina kontrolní a tři skupiny prověřované) se vypočítá bodový průměr zde zařazených soutěžících. Pokud vypočítaný průměr některé z prověřovaných skupin byl alespoň o 30% vyšší než bodový průměr kontrolní skupiny, zapíše se úspěšní řešitelé soutěže z dané prověřované skupiny do výsledné tabulky.

### 4.3 Vyhodnocení pravděpodobnosti podvádění

Situace ve třídách mohla být ovlivněna různými faktory, o kterých se pouze ze získaných dat nedozvíme. Rozhodnutí o tom, zda soutěžící skutečně podváděl, tedy nemůže být dáno pouze jedním z výše uvedených identifikátorů. Proto byla každému identifikátoru přidělena konkrétní váha, která vyjadřuje míru pravděpodobnosti podvodného jednání. Všem identifikátorům samostatných rizikových jevů byla nastavena na hodnotu 10 s výjimkou identifikátoru S, který má váhy nastaveny na hodnotu 10 pro 15 shodných odpovědí, 7 pro 14 shodných odpovědí a 4 v případě 13 shodných odpovědí. Všem hromadným identifikátorům byla nastavena váha na hodnotu 3. Pokud aplikace zaznamená u jednoho soutěžícího větší počet identifikátorů rizikových jevů, hodnoty jejich vah se sčítají. Čím vyšší celkovou váhu bude soutěžící mít, tím je vyšší pravděpodobnost, že daný soutěžící při testu podváděl.

### 4.4 Vytvoření samotné aplikace

Aplikaci jsem vytvořil v programovacím jazyce PHP<sup>2</sup>, podle výše uvedeného návrhu. Aplikaci jsem propojil s již existující databází soutěže Bobřík informatiky a využil tak již existujících tabulek. Nemusí se tedy žádná data exportovat do jiného umístění. Aplikace si vždy sama vytvoří tabulku, do které bude ukládat své výsledky.



Obrázek 2: Diagram zpracování dat

Na obrázku 2 můžeme vidět s jakými tabulkami aplikace spolupracuje. Aplikace načítá data z tabulek soutěžící, odpovědi, školy, otázky a kategorie. Tato data poté zpracuje, vyhodnotí a výsledky uloží do tabulky podvádějící.

<sup>2</sup> PHP - Hypertextový preprocesor, skriptovací programovací jazyk pro programování dynamických webových aplikací

## 4.5 Nastavení aplikace

V této kapitole uvádím důležité informace pro správce aplikace, jak pracovat s jednotlivými položkami nastavení. Mezi nejdůležitější patří nastavení spojení s databází (kapitola 4.5.3) a nastavení názvů tabulek (kapitola 4.5.4). Kapitoly 4.5.1 a 4.5.2 jsou popsány pro případné budoucí změny v programu. Nastavení uvedené v těchto dvou kapitolách jsou do kódu zapsány pevně a není je možné měnit přes GUI<sup>3</sup> aplikace z důvodu zachování stejné míry přísnosti aplikace pro jednotlivé ročníky soutěže a také z důvodu náročnosti aplikace na procesorový čas, který se pohybuje v řádech hodin (viz kapitola 4.7), tudíž by nastavit požadovanou přísnost mohlo trvat i několik dnů.

Veškeré nastavení vyhodnocujícího programu je uloženo ve složce **inc** v kořenovém adresáři programu v souboru **config.php** (vyhodnoceni/inc/config.php). Nastavení je tvořeno formou konstant.

### 4.5.1 Nastavení rozsahů

Toto nastavení popisují zejména z důvodu možné budoucí úpravy, jak kvůli změně některých pravidel soutěže nebo některých skutečností, tak kvůli možnosti zpřísnění či naopak zmírnění nastavených rozsahů.

Nastavení rozsahů určuje, jak velké časové nebo bodové rozsahy jsou povoleny. Všechny časové rozsahy se nastavují v sekundách.

- První konstanta **RYCHLEJŠÍ\_O** nastavuje, o jaký maximální možný čas může být soutěžící rychlejší, než je průměrný čas skládání testu všech soutěžících v kategorii. Využívá ho identifikátor R.
- Druhá **RYCHLEJSI\_SKOLA\_O** udává, o jaký maximální čas může být škola v jedné kategorii v průměru rychlejší se skládáním testu, než je průměrný čas skládání testu všech škol v dané kategorii. Tato hodnota se využívá u identifikátoru K.
- Další konstanta **START\_DELE\_O** udává maximální povolený časový rozestup mezi registrací a následným spuštěním testu. Patří k identifikátoru P.
- **MIMO\_DOZOR** nastavuje dobu, po kterou začal soutěžit jednotlivec nebo malá skupinka (viz další konstanta), který mohl soutěžit mimo dozor.

---

<sup>3</sup> GUI - grafické uživatelské rozhraní

- **POCET\_MIMO\_DOZOR** souvisí s předchozí konstantou a udává maximální velikost skupinky, u které je pravděpodobné, že soutěžila mimo dozor. Obě hodnoty patří k identifikátoru M.
- Konstanta **MIN\_POCET\_KATEGORIE** nastavuje minimální počet soutěžících v kategorii na škole, který je potřebný pro výpočet bodového průměru. Patří k identifikátoru pro zjištění, zda nebyla na škole výrazně lepší kategorie, tedy k identifikátoru L.
- **PRUMER\_KATEGORIE\_O** udává, o jaký maximální průměrný bodový zisk mohla být kategorie na škole úspěšnější, než kategorie ostatní. Tato konstanta také patří k identifikátoru L.
- **MIN\_POCET\_ZLEPSOVANI** udává minimální počet soutěžících, u kterých je možné zjistit, zda se na škole v dané kategorii v průběhu dne zlepšovala úspěšnost.
- **LEPSI\_O\_PROC** je konstanta nastavující, o kolik procent bodů více může mít skupina maximálně, oproti skupině, která skládala test dříve. Hodnota se nastavuje jako desetinné číslo, které se přičte k 1 (+100%). Při nastavení 25% bude hodnota 1,25 ( $25\% = 0,25; 1 + 0,25 = 1,25$ ). Předchozí dvě konstanty patří k identifikátoru K.
- **SPOLU\_ZACATEK\_O** je nastavena na maximální možný rozdíl mezi začátkem soutěžení dvou soutěžících, kteří mají 13 a více shodných odpovědí.
- **SPOLU\_KONEC\_O** je nastavena na maximální možný rozdíl mezi koncem soutěžení dvou soutěžících, kteří mají 13 a více shodných odpovědí.
- Konstanta **SNIZENI\_HRANICE** nastavuje, o kolik bodů méně může mít soutěžící oproti hranici, aby byl ještě zahrnut do porovnávání stejných odpovědí v identifikátoru S.

Samotný kód tohoto nastavení:

```
define("RYCHLEJSI_O", 300);
define("RYCHLEJSI_SKOLA_O", 300);
define("START_DELE_O", 600);
define("MIMO_DOZOR", 600);
define("POCET_MIMO_DOZOR", 4);
define("MIN_POCET_KATEGORIE", 10);
```



```
define("PRUMER_KATEGORIE_O", 25);
define("MIN_POCET_ZLEPSOVANI", 12);
define("LEPSI_O_PROC", 1.3);
define("SPOLU_ZACATEK_O", 300);
define("SPOLU_KONEC_O", 600);
define("SNIZENI_HRANICE", 20);
```

#### 4.5.2 Nastavení vah

Toto nastavení slouží pro nastavení velikosti vah, které budou přiděleny jednotlivým identifikátorům.

- **SAMOSTATNA\_VAHA** je konstanta určená pro samostatné identifikátory podvádění.
- **HROMADNA\_VAHA** nastavuje váhu hromadných identifikátorů.
- **TRINACT\_VAHA** patří k identifikátoru shodných odpovědí a určuje váhu pro 13 shodných.
- **CTRINACT\_VAHA** patří také k identifikátoru shodných odpovědí, určuje váhu pro 14 shodných.
- **PATNACT\_VAHA** patří ke stejnému identifikátoru jako předchozí dvě a nastavuje váhu pro 15 shodných odpovědí.

Kód tohoto nastavení:

```
define("SAMOSTATNA_VAHA", 10);
define("HROMADNA_VAHA", 3);
define("TRINACT_VAHA", 4);
define("CTRINACT_VAHA", 7);
define("PATNACT_VAHA", 10);
```

#### 4.5.3 Spojení s databází

Tato část nastavení slouží pro spojení aplikace s databázovým serverem.

- **DB\_SERVER** označuje server, na kterém běží databáze. Často bývá nastaven jako localhost.
- **DB\_USER** značí uživatelské jméno.
- **DB\_PASS** značí uživatelské heslo.
- K výběru databáze slouží **DB\_NAME**, kde se nastavuje název databáze.

Kód tohoto nastavení:

```
define("DB_SERVER","localhost");
define("DB_USER","uzivatelske_jmeno");
define("DB_PASS","heslo");
define("DB_NAME","nazev_databaze");
```

#### 4.5.4 Nastavení tabulek

Nastavení tabulek je potřebné měnit, pokud chceme kontrolovat starší ročníky soutěže. To je možné nahrazením prostřední části názvů tabulek, kdy **akt** nahradíme rokem konání soutěže. Například pro kontrolu ročníku 2014 bude tabulka soutěžících změněna z **ibobr\_akt\_soutezici** na **ibobr\_2014\_soutezici**. Takovýmto způsobem se mění všechny názvy tabulek až na tabulku škol, konkrétně **TBL\_SKOLA** (**ibobr\_skola\_new**), ta zůstává vždy stejná.

- Do konstanty **TBL\_SOUTEZICI** připisujeme název tabulky se soutěžícími (**ibobr\_akt\_soutezici**).
- Do **TBL\_ODPOVEDI** patří tabulka odpovědí (**ibobr\_akt\_odpovedi**).
- Dále **TBL\_OTAZKY** označuje tabulku s otázkami (**ibobr\_akt\_otazky**).
- Pro zjištění bodových hranic jednotlivých kategorií musíme znát tabulku kategorií **TBL\_KATEGORIE** (**ibobr\_akt\_kategorie**).
- A poslední tabulka **TBL\_VYSLEDKY** označuje tabulku, do které se budou ukládat výsledná data aplikace (**ibobr\_akt\_podvadejici**).

Toto nastavení se dá měnit hromadně. Stačí změnit hodnotu proměnné **rocnik** na druhém řádku souboru na rok konání ročníku soutěže, který chceme aplikací vyhodnotit. Kód tohoto nastavení:

```
$rocnik = "akt";
define("TBL_SOUTEZICI","ibobr_".$rocnik."_soutezici");
define("TBL_ODPOVEDI","ibobr_".$rocnik."_odpovedi");
define("TBL_OTAZKY","ibobr_".$rocnik."_otazky");
define("TBL_SKOLA","ibobr_skola_new");
define("TBL_KATEGORIE","ibobr_".$rocnik."_kategorie");
define("TBL_VYSLEDKY","ibobr_".$rocnik."_podvadejici");
```

## 4.6 Spuštění programu

### 4.6.1 Webový prohlížeč

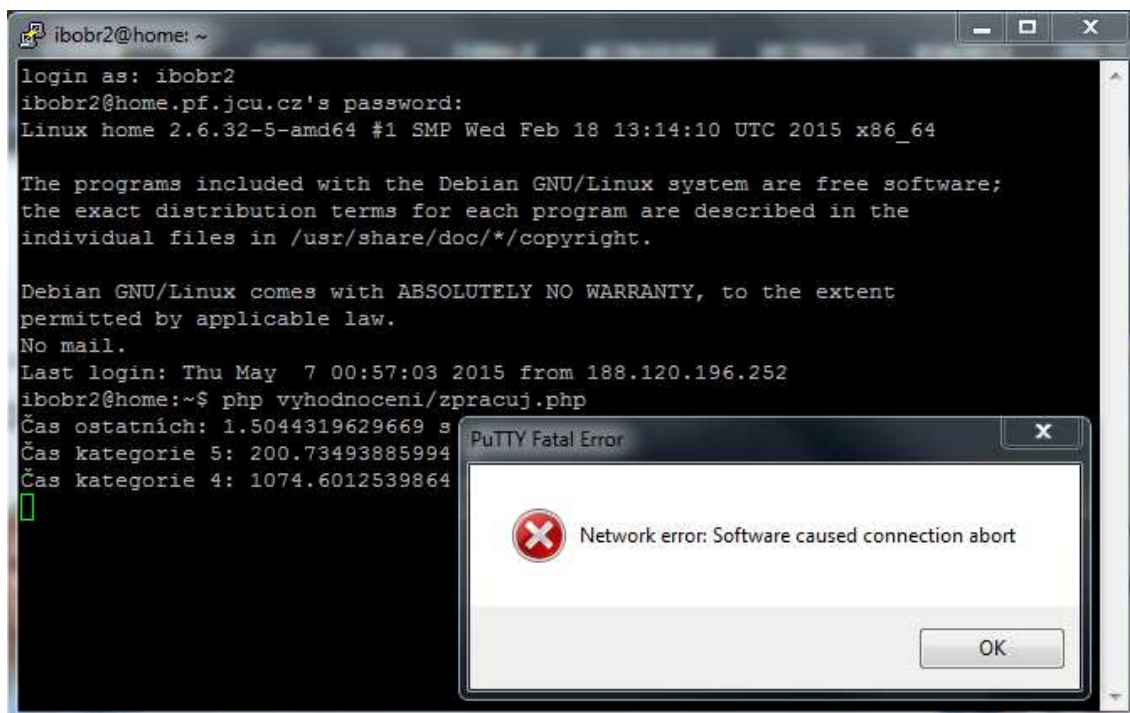
Původně bylo předpokládáno, že se program bude spouštět přímo z webového prohlížeče a to načtením stránky, na které by byla umístěna tlačítka, přes která by se

mohl program spustit celý nebo každá kategorie zvlášť. Vzhledem k tomu, že program zpracovává velký objem dat, se doba provedení skriptu pohybuje v řádech hodin (záleží na počtu soutěžících, čím více soutěžících, tím je doba provedení delší). Problémem je, že PHP server má nastavenou maximální dobu provádění skriptu. Defaultní hodnota je 30 sekund. [14] Po této době je vypsaná tato chyba: "Fatal error: Maximum execution time of 30 seconds exceeded in" a skript je ukončen. Toto omezení je ochranou před špatně napsanými zacyklenými skripty, které by na serveru běžely, dokud by je někdo ručně neukončil. Defaultní nastavení lze změnit v souboru `php.ini` pod položkou **max\_execution\_time**. Pro náš program by byla potřeba hodnota řádově v hodinách, později při stále se zvyšujícím počtu soutěžících dokonce v řadu desítek hodin. V ideálním případě, kdy bychom nastavili hodnotu na 0, by byla doba pro provedení skriptu neomezená. Každopádně měnit tuto hodnotu je nežádoucí, a proto bylo hledáno jiné řešení.

Pokud by byl tento čas upraven, je možné program spustit z této adresy: <http://server.doména/umístění/vyhodnoceni/zpracuj.php>.

#### 4.6.2 Bash

Druhým a výhodnějším řešením je skript spouštět přímo z `bash`. Skript v `bash` lze spustit přes příkaz: "*php umístění\_souboru/spouštěcí\_soubor.php*". V tomto případě nás nijak neomezuje funkce `max_execution_time` a takto spuštěný program nemá omezenou dobu provedení, ale může se vyskytnout problém se sítí (například neočekávané odpojení od internetu) v průběhu provádění skriptu (viz Obrázek 3). V takovém případě se skript ukončí.



Obrázek 3: Chyba sítě při spuštění aplikace přes terminál

Proto je nanejvýš výhodné spustit skript na pozadí pomocí příkazu **nohup**. Pokud použijeme příkaz **nohup**, přesměruje se výstup skriptu do defaultně nastaveného souboru "**nohup.out**". V tomto souboru lze tedy nalézt již zmíněný výstup skriptu, ale také případné hlášení chyb. Poté se lze odhlásit a terminál zavřít, spuštěný skript na serveru poběží dál, proto se nemusíme bát problémů se sítí.

Spuštění programu z bashe lze provést několika způsoby. Příkaz na spuštění a výpis výstupu programu do terminálu: `php umístění/vyhodnoceni/zpracuj.php`. Spuštění na pozadí s výpisem do souboru `nohup.out`, který je umístěn do adresáře, ze kterého je skript spuštěn: `nohup php umístění/vyhodnoceni/zpracuj.php &`.

## 4.7 Doba běhu aplikace

Vzhledem k velkému počtu zpracovávaných dat se doba běhu aplikace zpracovávající některý z ročníků odhaduje v řádu hodin.

Aplikace byla testována na školním webovém serveru, kde je umístěna testovací verze soutěže Bobřík informatiky, na ročnících konaných v letech 2013 a 2014. Po celou dobu běhu aplikace bylo přidělené CPU<sup>4</sup> vytíženo na 100% a procesor celkově byl vytěžován přibližně na 25%. (viz Obrázek 4, celkové vytížení se nachází na 3 řádce s názvem %Cpu(s) jako první hodnota)

<sup>4</sup> CPU - centrální procesorová jednotka

```
top - 15:34:16 up 1 day, 22:26, 2 users, load average: 0,50, 0,18, 0,10
Tasks: 111 total, 2 running, 109 sleeping, 0 stopped, 0 zombie
%Cpu(s): 26,4 us, 0,1 sy, 0,0 ni, 73,1 id, 0,4 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem: 2527860 total, 2422216 used, 105644 free, 233848 buffers
KiB Swap: 7678972 total, 0 used, 7678972 free, 1732864 cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
15260	ibobrr2	20	0	182m	24m	7424	R	100,0	1,0	1:05.30	php
14635	www-data	20	0	244m	21m	4336	S	5,6	0,9	0:15.10	apache2
3021	mysql	20	0	508m	95m	8000	S	0,7	3,9	5:02.72	mysqld

Obrázek 4: Náhled vytížení serveru při běhu aplikace

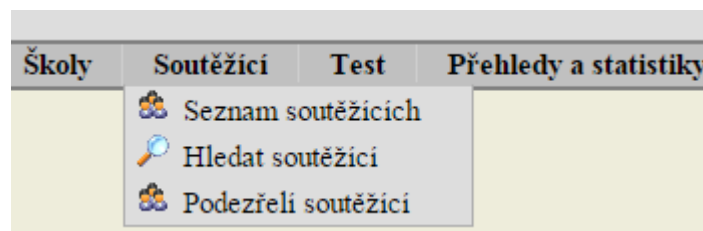
Doba běhu aplikace při kontrole ročníku konaném v roce 2013 s 34 460 soutěžícími trvalo aplikaci vyhodnotit přibližně za 1 hodinu a 30 minut. Při kontrole ročníku konaném v roce 2014, kterého se účastnilo přibližně o 10 000 více (tj. 44 083) účastníků, trvalo aplikaci vyhodnocení přibližně 2 hodiny a 53 minut.

## 5 Zobrazovací stránka

Součástí bakalářské práce bylo vytvoření načítání výsledků aplikace. K zobrazování výsledků bylo využito softwaru administrační sekce, který zajišťuje správu a nastavení všech částí soutěžní aplikace. Za pomoci již existujících funkcí byla přidána nová položka do administrátorského horizontálního hlavního menu, která odkazuje na nově přidanou webovou stránku, na které je možno zobrazit výsledky vyhodnocující aplikace.

### 5.1 Vytvoření nové stránky

Položka do horizontálního hlavního menu byla přidána v souboru `test/admin/adm.php`. K položkám s názvem **Seznam soutěžících** a **Hledat soutěžící** patřící pod nadřazenou položku seznamu s názvem **Soutěžící** byla přidána položka **Podezřelí soutěžící**. Na obrázku 5 můžeme vidět přidaný odkaz na nově tvořenou stránku.



Obrázek 5: Náhled přidané položky **Podezřelí soutěžící** do menu

Samotná stránka byla potom přidána přidáním volby `case` do funkce `switch` do souboru `test/admin/sekce/adm_sprava_soutezicich.php`. S využitím funkce `nacistModul()`, která načítá moduly ze složek, v tomto případě ze složky `sprava_soutezicich`.

Jako poslední krok k přidání a načtení nové stránky bylo potřeba vytvořit nový PHP soubor právě do složky `sprava_soutezicich`. Celá cesta k souboru je `test/admin/sekce/sprava_soutezicich/seznam2.php`. V tomto souboru bude kód, který bude zpracovávat data, která se budou následně na stránce zobrazovat.

#### 5.1.1 Vytvoření pohledu

Pro jednodušší práci a lepší přehlednost byl z potřebných dat vytvořen takzvaný databázový pohled<sup>5</sup>. Ten byl složen ze sloupců napříč několika databázových tabulek. Samotný SQL dotaz pro vytvoření pohledu ze všech potřebných tabulek je celkem

<sup>5</sup> Databázový pohled - databázový objekt, který poskytuje data jako tabulka, ale nejsou v něm tato data přímo uložena

složité, dlouhý a víceméně nepřehledný, ale o to více se nám pak ulehčí práce s daty ve vytvořeném pohledu.

Z důvodu dvojího zobrazování dat na stránce (pokud není dostupná tabulka s výsledky soutěže (př.: `ibobr_akt_vysledky`), zobrazují se data bez sloupce **Validován**, pokud je tato tabulka dostupná jsou zobrazena data i s tímto sloupcem) se při prvním načítání stránky rozhoduje, zda se vytvoří pohled se sloupcem **valid** (tj. Validován) nebo bez něj. Samotný pohled je vytvořen při pokusu o načtení stránky, pokud neexistuje a pokud jsou dostupné tabulky pro jeho vytvoření. PHP kód tohoto rozhodování:

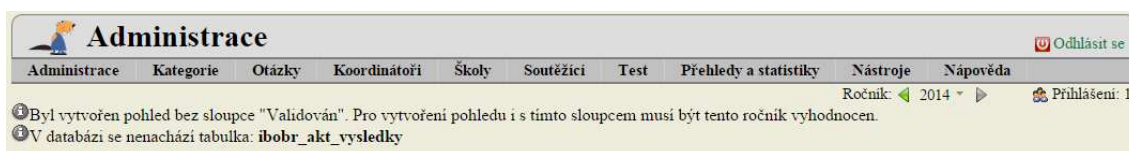
```
if(!mysql_query("SELECT * FROM ".tbl('zobrazeni')." WHERE 0=1", $spojeni)){
    if(!mysql_query("CREATE VIEW ".tbl('zobrazeni')."
AS SELECT ".tbl('podvadejici')." .soutezici, ".tbl('soutezici')." .jmeno,
".tbl('soutezici')." .prijmeni, ".tbl('soutezici')." .skola, name_short AS
jmeno_skoly, ".tbl('soutezici')." .kategorie, nazev AS nazev_kategorie,
(sam + spolu + z_rychle_skoly + opozdeny_zacatek + rychly_cas +
lepsi_kategorie + zlepshujici_se) AS vaha, ".tbl('soutezici')." .body, sam,
spolu, z_rychle_skoly, opozdeny_zacatek, rychly_cas, lepsi_kategorie,
zlepshujici_se, IFNULL(".tbl('vysledky')." .poradi DIV
".tbl('vysledky')." .poradi,0) AS valid
FROM (((".tbl('soutezici')." INNER JOIN ".tbl('podvadejici')."
ON ".tbl('soutezici')." .id = ".tbl('podvadejici')." .soutezici)
INNER JOIN ".tbl('skola')." ON ".tbl('soutezici')." .skola =
".tbl('skola')." .id) INNER JOIN ".tbl('kategorie')."
ON ".tbl('soutezici')." .kategorie = ".tbl('kategorie')." .id)
LEFT JOIN ".tbl('vysledky')." ON ".tbl('soutezici')." .id =
".tbl('vysledky')." .id)", $spojeni)){
    echo('<br />');
    if(!mysql_query("CREATE VIEW ".tbl('zobrazeni')."
AS SELECT ".tbl('podvadejici')." .soutezici, jmeno, prijmeni, skola, name_short
AS jmeno_skoly, kategorie, nazev AS nazev_kategorie, (sam + spolu +
z_rychle_skoly + opozdeny_zacatek + rychly_cas + lepsi_kategorie +
zlepshujici_se) AS vaha, ".tbl('soutezici')." .body, sam, spolu, z_rychle_skoly,
opozdeny_zacatek, rychly_cas, lepsi_kategorie, zlepshujici_se
FROM ".tbl('soutezici')." , ".tbl('podvadejici')." , ".tbl('skola')." ,
".tbl('kategorie')."
WHERE ".tbl('soutezici')." .id = ".tbl('podvadejici')." .soutezici
AND ".tbl('soutezici')." .skola = ".tbl('skola')." .id
AND ".tbl('soutezici')." .kategorie = ".tbl('kategorie')." .id", $spojeni)){
        echo('Pro
vytvoření pohledu, chybí v databázi některá z těchto tabulek: <span
class="tucne">'.tbl('soutezici')." , '.tbl('podvadejici')." </span><br />');
    }else{
```

```

        echo('Byl
vytvořen pohled bez sloupce "Validován". Pro vytvoření pohledu i s tímto
sloupcem musí být tento ročník vyhodnocen.<br />');
        echo('
V databázi se nenachází tabulka:
<span class="tucne">' .tbl('vysledky') . '</span><br />');
    }
}
}

```

V první podmínce se rozhoduje, zda již existuje pohled s názvem **zobrazeni**, pokud ano načtou se data z tohoto pohledu a zobrazí se, pokud však pohled ještě neexistuje, pokusí se tento pohled vytvořit se všemi sloupci. Povede-li se pohled vytvořit přejde se na zobrazování dat. Pokud se tento pohled z důvodu nepřítomnosti některé z potřebných tabulek v databázi vytvořit nepodaří, pokusí se vytvořit pohled bez sloupce **valid** (chybí tabulka s výsledky). Vytvoří-li se pohled bez tohoto sloupce, přejde se na zobrazování dat a nad zobrazenou tabulkou se zobrazí text: **Byl vytvořen pohled bez sloupce "Validován". Pro vytvoření pohledu i s tímto sloupcem musí být tento ročník vyhodnocen. V databázi se nenachází tabulka:** název tabulky s výsledky (viz Obrázek 6). Pokud se však ani při tomto pokusu pohled nevytvoří, znamená to, že v databázi chybí některá další z potřebných tabulek. V tomto případě se vypíše text: **Pro vytvoření pohledu, chybí v databázi některá z těchto tabulek:** názvy potřebných tabulek (viz Obrázek 7).



Obrázek 6: Náhled vrchní části stránky po vytvoření pohledu bez sloupce Validován



Obrázek 7: Náhled vrchní části stránky v situaci, kdy nebyl vytvořen pohled



### 5.1.2 Zobrazení dat z vytvořeného pohledu

Dalším krokem je rozhodnutí, která data se mají zobrazovat. Zda data i se sloupcem **Validován** nebo bez něj. K tomuto slouží tyto dvě do sebe vnořené podmínky:

```
if(!mysql_query("SELECT valid FROM ".tbl('zobrazeni')." WHERE
0=1", $spojeni)){
    if(mysql_query("SELECT * FROM ".tbl('zobrazeni')." WHERE
0=1", $spojeni)){
        echo ('<br />
Zobrazují se data bez sloupce "Validován".<br />
Pro zobrazení dat i
se sloupcem "Validován" musí být tento ročník vyhodnocen.<br />
Pokud už je tento
ročník vyhodnocen, aktualizujte databázový pohled tlačítkem
aktualizovat.<br />');
    }
}
```

První podmínka kontroluje, zda se v pohledu s názvem zobrazení nachází sloupec pojmenovaný **valid**. Pokud se v tomto pohledu tento sloupec nachází, pokračuje se dále v kódu, který zajišťuje zobrazování dat i s tímto sloupcem. Pokud tento sloupec v pohledu není, kontroluje se druhá podmínka, zda je vůbec pohled nazývaný zobrazení vytvořený. Když je tento pohled vytvořen znamená to, že neobsahuje sloupec **valid**, v takovém případě se vypíše následující informace: **Zobrazují se data bez sloupce "Validován". Pro zobrazení dat i se sloupcem "Validován" musí být tento ročník vyhodnocen. Pokud už je tento ročník vyhodnocen, aktualizujte databázový pohled tlačítkem aktualizovat.** (viz Obrázek 8). A přejde se na kód zajišťující zobrazování dat bez tohoto sloupce.

Obrázek 8: Náhled vrchní části stránky při zobrazování bez sloupce Validován

Kliknutím na tlačítko **Aktualizovat** se nejprve smaže již existující pohled. O to se postará funkce se SQL dotazem:

```
mysql_query("DROP VIEW ".tbl('zobrazeni')." ", $spojeni);
```

a dále probíhá stejný proces jako při prvním načítání stránky.

## 5.2 Zobrazované sloupce dat

Mezi zobrazované sloupce byla vybrána tato data:

- ID
- Jméno
- Příjmení
- Škola
- Kategorie
- Váha
- Body
- Validován
- Atributy

**ID** je osobní číslo přidělené každému soutěžícímu při registraci. Toto číslo spojuje všechna data patřící danému soutěžícímu. **Jméno** a **Příjmení** jsou identifikačními údaji účastníků. Sloupce **Škola** a **Kategorie** jsou uvedeny pro přehlednost, z které školy je soutěžící a v které kategorii soutěžící soutěžil, a také pro možnost filtrování. Sloupec **Váha** zobrazuje váhu, kterou soutěžícímu přidělila vyhodnocující aplikace. Body získané v testu se zobrazují ve sloupci **Body**. Sloupec **Validován** se zobrazuje pouze v případě, že jsou výsledky aktuálně zobrazovaného ročníku dostupné (viz Obrázek 9 a Obrázek 10). Sloupec s názvem **Atributy** obsahuje takzvané atributy. Ty jsou zobrazeny podle toho, ve kterých identifikátorech byl

soutěžící označen vyhodnocující aplikací. Při najetí kurzorem na jednotlivé atributy se zobrazují jejich popisy.

Data lze abecedně řadit podle většiny sloupců. Jen sloupec s atributy takto řadit nelze.

#	ID	Jméno	Příjmení	Škola	Kategorie	Váha	Body	Atributy
1	38389	Martin	J.	Nezmáná škola	Mini	16	121	K R Z
2	35026	Patrik	G.	Nezmáná škola	Benjamin	17	204	M S Z

Obrázek 9: Náhled zobrazovaných sloupců - bez sloupce "Validován"

#	ID	Jméno	Příjmení	Škola	Kategorie	Váha	Body	Validován	Atributy
1	38389	Martin	J.	Nezmáná škola	Mini	16	121	Ano	K R Z
2	35026	Patrik	G.	Nezmáná škola	Benjamin	17	204	Ano	M S Z

Obrázek 10: Náhled zobrazovaných sloupců - se sloupcem "Validován"

### 5.3 Filtrování dat

Filtry pro filtrování dat byly voleny takové, aby bylo možné zobrazovat seznamy soutěžících podle školy, za kterou soutěžící soutěžili, podle kategorie, ve které studenti soutěžili, podle váhy přidělené vyhodnocující aplikací, a podle toho zda byl soutěžící validován.

Použité filtry:

- Škola
- Kategorie
- Váha rovna
- Váha větší nebo rovna
- Váha menší nebo rovna
- Validován

Filtr s váhou je zastoupen celkem třikrát a to z důvodu poskytnutí filtrování libovolného rozsahu vah. Díky tomu lze získat seznamy soutěžících, u kterých je podvádění vyloučeno, nepravděpodobné nebo pravděpodobné. Například pokud chceme zobrazit seznam soutěžících, u kterých je podvádění vyloučeno zvolíme filtr **Váha <= 9** (viz Obrázek 11). Pro zobrazení seznamu soutěžících, u kterých je podvádění pravděpodobné zvolíme filtr **Váha = 10** a k vypsání seznamu soutěžících, u kterých bylo podvádění označeno za pravděpodobné, použijeme filtr **Váha >= 13**.

Filtrovat podezřelé podle: - Škola - - Kategorie - - Váha rovna - - Váha >= - - Váha <= 9 - Validován - Výchozí

#	ID	Jméno	Příjmení	Škola	Kategorie	Váha	Body	Validován	Atributy
1	60	Vojtěch	A.	Nezmáná škola	Senior	4	153	Ano	S
2	74	Jan	O.	Nezmáná škola	Senior	7	156	Ano	S
3	78	Jakub	T.	Nezmáná škola	Senior	7	176	Ano	S
4	157	Martin	S.	Nezmáná škola	Senior	4	212	Ano	S
5	166	Jiří	G.	Nezmáná škola	Senior	4	208	Ano	S
6	172	Daniel	M.	Nezmáná škola	Senior	4	240	Ano	S
7	228	Anna	I.	Nezmáná škola	Senior	4	172	Ano	S
8	307	Jiří	H.	Nezmáná škola	Senior	4	152	Ne	S
9	309	Karel	J.	Nezmáná škola	Senior	4	152	Ne	S
10	322	Václav	C.	Nezmáná škola	Senior	4	176	Ano	S

Počet zobrazených 10   1 2 3 4 5 6 7 8 9 10 11   Zobrazeno 10 z 2466 celkem. Strana 1 z 247.

Obrázek 11: Seznam se zvoleným filtrem "Váha <= 9"

Filtr pro zobrazení, zda byl soutěžící validován nebo ne se zobrazuje a jde použít pouze, pokud jsou dostupné výsledky právě zobrazovaného ročníku soutěže (viz Obrázek 12 a Obrázek 13), tak jako je to u stejnojmenného sloupce dat.

Jednotlivé filtry se dají libovolně kombinovat a lze tak zobrazit například jen seznam soutěžících, kteří jsou z určité školy, soutěžili v určité kategorii a měli přidělenou váhu vyšší nebo rovnou než zvolená hodnota.

Filtrovat podezřelé podle: - Škola - - Kategorie - - Váha rovna - - Váha >= - - Váha <= - Validován - Výchozí

Obrázek 12: Náhled lišty s použitými filtry - bez filtru "Validován"

Filtrovat podezřelé podle: - Škola - - Kategorie - - Váha rovna - - Váha >= - - Váha <= - Validován - Výchozí

Obrázek 13: Náhled lišty s použitými filtry - s filtrem "Validován"

Zobrazovaný seznam lze kdykoliv exportovat do formátu XLS<sup>6</sup> kliknutím na tlačítko **Exportovat**. Vždy se exportují jen data se zvolenými filtry, což lze využít například k zaslání seznamu soutěžících určité školy jejímu koordinátorovi.

<sup>6</sup> XLS - souborový formát je používán jako nativní formát souboru tabulkovým procesorem Microsoft Office Excel 2003 a staršími verzemi tohoto software [11]

## 6 Testování

Vytvořená aplikace byla podrobena několika testováním. Nejprve bylo nutné ověřit, zda se v kódu nevyskytují chyby a vše funguje správně. Bylo potřeba zkontrolovat, jestli zvolené rozsahy v nastavovacím souboru opravdu odpovídají požadavkům a nevyskytuje se žádná odchylka a zda jsou její výsledky validní.

Poté jsem její výstupy porovnal s výsledky uvedenými v článku v časopise *JITE* [10].

### 6.1 Simulace

Byl vytvořen fiktivní soubor se 45 soutěžícími ze dvou škol. Tito soutěžící se zapojili celkem do 4 kategorií soutěže. Na těchto fiktivních soutěžících byla testována funkčnost aplikace.

#### 6.1.1 Samostatné identifikátory

Testovací instituce pro testování samostatných identifikátorů byla nazvána Škola 1. Její žáci se zúčastnili kategorie č. 5, tedy Senior.

Soutěžící č. 1 a 2 se téměř shodují ve třech faktorech, a to v čase započetí testu, čase jeho ukončení (čas poslední odeslané odpovědi) a všechny jejich odpovědi byly shodné, ať již byly správné či chybné. Soutěžící zmiňované v této podkapitole lze vyhledat na obrázku 14 pod jejich čísly ve sloupci **soutezici**. Z obrázku je patrné, že aplikace správně identifikovala tyto dva soutěžící jako podezřelé. Ve sloupci **spolu** (identifikátor S) je uvedena hodnota 10. Aplikace tedy správně určila počet shodných odpovědí, tedy 15.

Časy započetí testu soutěžících č. 3 a 4 byly podobné předešlým dvěma. Tito studenti pracovali na testu spolu, měli 14 shodných odpovědí. Pouze jedna odpověď se lišila. Hodnotou ve sloupci **spolu** (identifikátor S) pro tyto soutěžící je 7, což je váha odpovídající 14 shodným odpovědím.

Soutěžící č. 5 se na spolupráci předchozích dvou studentů aktivně nepodílel, pouze výsledky opsal z jejich obrazovek. Takto získal 13 odpovědí. Na zbývající dvě otázky odpověděl sám a jinak. Na obrázku 14 lze u tohoto soutěžícího nalézt hodnotu 4, která značí 13 shodných odpovědí.

Co se týče soutěžících č. 6 a 7, test plnili bez dozoru. Vhodný povšimnutí je časový rozdíl mezi registrací a spuštěním testu u studenta č. 6. Jednalo se o více než 10 minut. Všechny jeho odpovědi byly shodné s odpověďmi soutěžících č. 1 a 2, nebyl

však jejich plnění testu přítomen. Otázky plnil v jiný čas. U studenta číslo 7 je třeba zdůraznit krátký čas, během kterého k zodpovězení všech testových otázek došlo. Test ukončil hned po odeslání poslední odpovědi. U obou těchto studentů byla ve sloupci **sam** (identifikátor M) výstupem hodnota 10. Aplikací tedy bylo správně rozpoznáno, že soutěžili bez dozoru (shlukuje totiž do skupinek soutěžící, kteří započali test psát v podobný čas a zároveň náleží do stejné školy a kategorie, je-li jejich počet menší než 5). U soutěžícího č. 6 je ve sloupci **opozdeny\_zacatek** (identifikátor P) hodnota 10, značící delší časovou prodlevu mezi registrací a spuštěním testu. Soutěžící č. 7 má ve sloupci **rychly\_cas** (identifikátor R) hodnotu 10, bylo tedy úspěšně odhaleno, že čas, který potřeboval ke složení testu, byl výrazně kratší než průměrný čas.

Čas spuštění testu studentem č. 8 byl přibližně stejný jako u prvních pěti soutěžících. Tento student ukončil test jen dvě minuty před vypršením časového limitu, což sice není proti pravidlům, nicméně jeho poslední odpověď byla odeslána mnohem dříve. Z těchto dalších údajů je tedy patrné, že otázky zodpověděl za čas, který byl výrazně kratší než průměrný čas a pozdním ukončením testu se tento fakt snažil skrýt. Aplikace tento skutek zaznamenala, tudíž můžeme vidět ve sloupci **rychly\_cas** (identifikátor R) u soutěžícího č. 8 hodnotu 10.

### 6.1.2 Hromadné identifikátory

Dále došlo k testování hromadných identifikátorů ze tří hledisek. Nejprve bylo zkoumáno, zda průměrný čas studentů této školy není nápadně kratší, než je tomu u průměrných časů jiných škol soutěžících v konkrétních kategoriích.

Dále jsme se zaměřili na výkony soutěžících školy v jednotlivých kategoriích. Pokud byl průměrný bodový zisk v jedné kategorii výrazně vyšší než v ostatních, je zde možnost, že dozor dané skupiny byl studentům při řešení nápomocen, popřípadě mohlo dojít ke spolupráci studentů samotných.

Třetím faktorem, který byl pozorován byly zlepšující se výsledky v rámci jedné kategorie na jedné škole s přibývajícím časem. Pokud se skupiny soutěžících plnící postupně tentýž test zlepšují, je patrné, že si mezi sebou předávaly informace o jeho obsahu.

#### 6.1.2.1 Krátký průměrný čas školy

Jako testovací instituce pro testování tohoto identifikátoru byla použita Škola 1. Její nově přidání žáci se zúčastnili kategorie č. 4, tedy Junior. Tato množina obsahuje celkem deset soutěžících, jejichž průměrný soutěžní čas je 35 minut. Dále pracujeme se

Školou 2 soutěžící ve stejné kategorii. Průměrný čas skládání testu na této instituci je 25,4 minuty. Je tedy téměř o 10 minut kratší než v předešlém případě. Testování tedy byli soutěžící školy s nápadně kratším časem skládání testu.

Soutěžící č. 20 potřeboval k zodpovězení otázek 20 minut, což lze považovat za velice krátký čas. Student č. 21 však potřeboval dokonce jen 18 minut. V tomto trendu pokračoval i další student Školy 2. Soutěžící č. 22 složil test za 23 minut. V opozici k těmto údajům stojí soutěžící č. 23 a 24. Prvnímu z nich trvalo vypracování testu 36 minut, což je doba, která se blíží časovému limitu 40 minut, který je ke složení testu k dispozici. Druhý z nich, student č. 24, strávil tímto úkonem 30 minut, tedy čas blíží se průměru (31,8 min).

Na obrázku 14 se nyní zaměříme na sloupec s názvem **z\_rychle\_skoly** (identifikátor K). Soutěžící č. 20 až č. 24 v něm obdrželi hodnotu 3, jelikož patří do školy, v níž je průměrný čas nápadně krátký. Je však třeba věnovat pozornost i sloupci **rychly\_cas** (identifikátor R), jelikož v něm byli s hodnotou 10 zaznamenáni studenti č. 20, 21 a 22. Získali tedy celkem váhu 13, která již signalizuje existující pravděpodobnost podvádění. Soutěžící č. 23 a 24 byli ohodnoceni celkovou váhou 3, podvádění je u nich tedy vyloučeno.

#### 6.1.2.2 Lepší kategorie na škole

První testovací institucí pro testování tohoto identifikátoru je Škola 1. Studenti se zúčastnili kategorie č. 4, tedy Junior. V tomto případě pracujeme s deseti studenty, jejichž bodový zisk je v průměru 190 bodů (tito studenti figurují i v 6.1.2.1).

Další kategorií, do které se tato škola zapojila, byla kategorie č. 3, Kadet. Té se zúčastnilo opět deset studentů, kteří tentokrát získali v průměru 216 bodů. Tato hodnota je o 26 bodů vyšší než předešlá. Maximální přípustná hranice je však 25 bodů. Tato hodnota je uvedena v nastavení aplikace.

Všichni studenti z kategorie č. 3 jsou tedy uvedeni ve výsledné tabulce (viz Obrázek 14) a to jako soutěžící s čísly 30 až 39. Ve sloupci **lepsi\_kategorie** (identifikátor L) mají uvedenu hodnotu 3, což je hodnota určená pro hromadné identifikátory.

Dále byl testován případ, kdy byl bodový průměr kategorie č. 3 na dané škole snížen na 214 bodů. Žádný ze studentů na Škole 1 v kategorii č. 3 nebyl uveden ve výsledné tabulce, a to proto, že jejich bodový průměr porovnaný s bodovým průměrem kategorie č.4 na stejné škole byl pod maximální povolenou hranicí.

### 6.1.2.3 Zlepšující se výsledky

Testovací institucí pro testování tohoto identifikátoru je Škola 1. Zapojila se do kategorie č. 2, tedy Benjamin. Obsahuje celkem 12 soutěžících. První polovina studentů začala test skládat dříve a jejich bodový zisk byl průměrně 160. Druhá polovina byla rozdělena na třetiny (tedy na dvojice).

Bodový průměr první dvojice byl vyšší o 15%, druhé o 30% a třetí o 45% oproti výsledku první polovina skupiny. Bylo předpokládáno, že první a druhá dvojice (soutěžící č. 46, 47, 48, 49) nebudou uvedeny ve výsledné tabulce. V případě třetí dvojice (soutěžící č. 50, 51) by tomu tak mělo být.

Na obrázku 14 můžeme vidět, že soutěžící č. 50, 51 jsou skutečně uvedeni ve výsledné tabulce a ve sloupci **zlepsujici\_se** (identifikátor Z) mají uvedenou hodnotu 3, přičemž soutěžící č. 46, 47, 48, 49 se v dané tabulce nenachází. Kdyby byl bodový průměr u druhé dvojice vyšší než u první poloviny soutěžících o více než 30%, byla by ve výsledcích uvedena i druhá dvojice (tj. soutěžící č. 48 a 49).



id	soutezici	sam	spolu	z_rychle_skoly	opozdeny_zacatek	rychly_cas	lepsi_kategorie	zlepsujici_se
146	1	0	10	0	0	0	0	0
147	2	0	10	0	0	0	0	0
148	3	0	7	0	0	0	0	0
149	4	0	7	0	0	0	0	0
150	5	0	4	0	0	0	0	0
143	6	10	0	0	10	0	0	0
144	7	10	0	0	0	10	0	0
145	8	0	0	0	0	10	0	0
151	20	0	0	3	0	10	0	0
152	21	0	0	3	0	10	0	0
153	22	0	0	3	0	10	0	0
154	23	0	0	3	0	0	0	0
155	24	0	0	3	0	0	0	0
133	30	0	0	0	0	0	3	0
134	31	0	0	0	0	0	3	0
135	32	0	0	0	0	0	3	0
136	33	0	0	0	0	0	3	0
137	34	0	0	0	0	0	3	0
138	35	0	0	0	0	0	3	0
139	36	0	0	0	0	0	3	0
140	37	0	0	0	0	0	3	0
141	38	0	0	0	0	0	3	0
142	39	0	0	0	0	0	3	0
156	50	0	0	0	0	0	0	3
157	51	0	0	0	0	0	0	3

Obrázek 14: Tabulka výsledků při testování aplikace

## 6.2 Porovnání výsledků ostré soutěže

Výsledky aplikace ze 6. ročníku konaného v roce 2013 byly porovnány s výsledky v článku Mgr. Václava Šimandla publikovaném v časopise *JITE* [10]. U 7. ročníku konaného v roce 2014 bylo provedeno pouze porovnání soutěžících, které by vyřadila aplikace a soutěžících, které vyřadili školní koordinátoři.

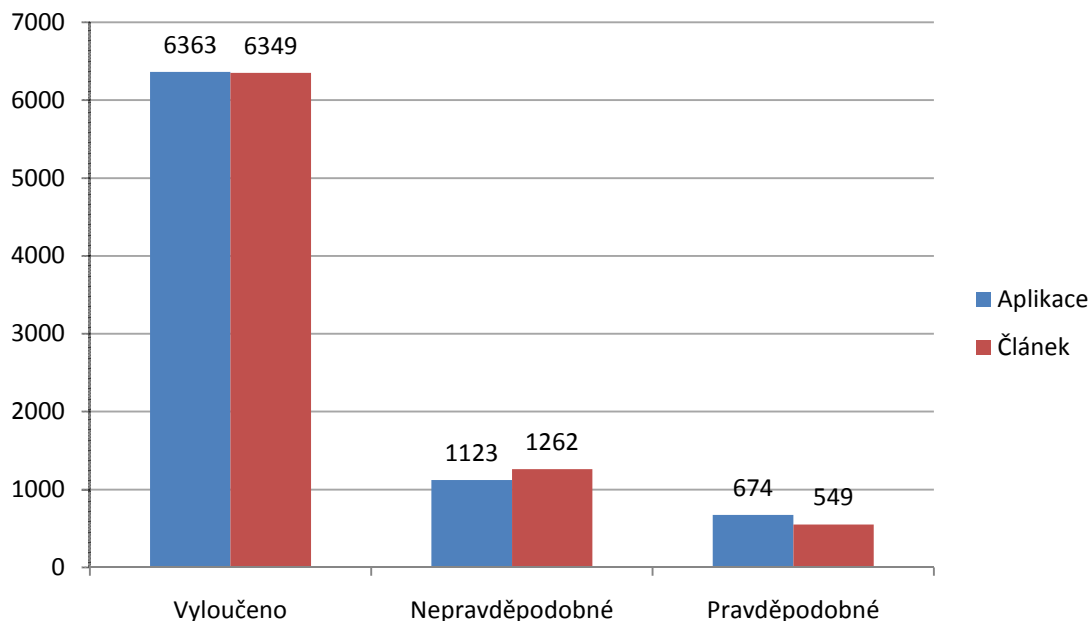
### 6.2.1 Ročník 2013

Ročníku pořádaném v roce 2013 se účastnilo celkem 34 454 soutěžících, z toho 8 501 soutěžících se stalo úspěšnými řešiteli soutěže. Školními koordinátory bylo vyřazeno 341 úspěšných soutěžících, řádných úspěšných řešitelů tedy bylo 8 160.

V článku bylo uvedeno, že podvádění bylo vyloučeno u 6 349 (tj. 77,8 %) řádných úspěšných řešitelů, jako nepravděpodobné bylo označeno u 1 262 (tj. 15,47 %)

řádných úspěšných řešitelů a za možné až pravděpodobné bylo označeno 549 (tj. 6,73 %) řádných úspěšných řešitelů. [10]

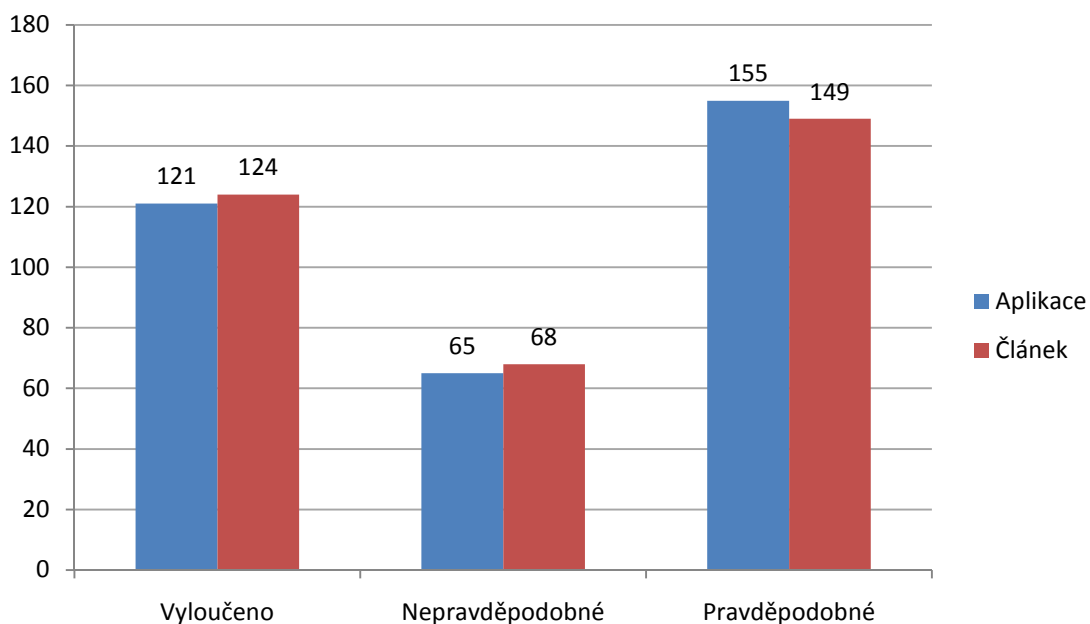
Podle aplikace bylo podvádění vyloučeno u 6 363 (tj. 77,98 %) řádných úspěšných řešitelů, jako nepravděpodobné bylo podvádění označeno u 1 123 (tj. 13,76 %) řádných úspěšných řešitelů a jako pravděpodobné bylo podvádění označeno u 674 (tj. 8,26 %) řádných úspěšných řešitelů. Přímé porovnání můžeme vidět na grafu 1.



**Graf 1: Podvádění u řádných úspěšných soutěžících, rok 2013**

Další porovnání je zaměřeno na úspěšné řešitele, kteří byli vyřazeni školními koordinátory. V článku je uvedeno, že z celkových 341 vyřazených úspěšných soutěžících bylo podvádění označeno jako možné až pravděpodobné u 149 (tj. 43,7 %), jako nepravděpodobné u 68 (tj. 19,94 %) vyřazených úspěšných řešitelů a jako vyloučené u 124 (tj. 36,36 %) vyřazených úspěšných řešitelů. [10]

Aplikace označila podvádění za pravděpodobné u 155 (tj. 45,46 %) vyřazených úspěšných řešitelů. Dále u 65 (tj. 19,06 %) vyřazených úspěšných řešitelů bylo podvádění označeno jako nepravděpodobné a u 121 (tj. 35,48 %) vyřazených úspěšných řešitelů bylo podle aplikace podvádění vyloučeno. Porovnání v grafické podobě můžeme vidět na grafu 2.



**Graf 2: Podvádění u koordinátory vyřazených úspěšných soutěží, rok 2013**

V tabulce 1 můžete vidět porovnání vah jednotlivých náhodně vybraných soutěží. Jak je vidět váhy se v některých případech shodují. U případů, kde se váhy zcela neshodují, je rozdíl vah většinou 4 nebo 7, to je dáno identifikátorem S, tedy shodnými odpověďmi, kdy má aplikace bere v úvahu i 13 a 14 shodných odpovědí. Pokud se váha liší výrazněji, přisuzuji to trochu jinému pohlížení na průměrný čas.

<b>Osobní číslo</b>	<b>Váha předchozího zkoumání</b>	<b>Váha nové aplikace/10</b>
12022	3,6	4
8243	3,3	3,3
28289	2,6	3,3
28311	2,6	3,3
34343	3,3	3,3
34416	2,3	3,3
31085	3	3
1487	2	2,7
4984	2	2,7
15970	1,6	1,6
26889	1,6	1,6
17333	1	1
18660	1	1,7
22535	1	1
31178	1	1
33646	1	0

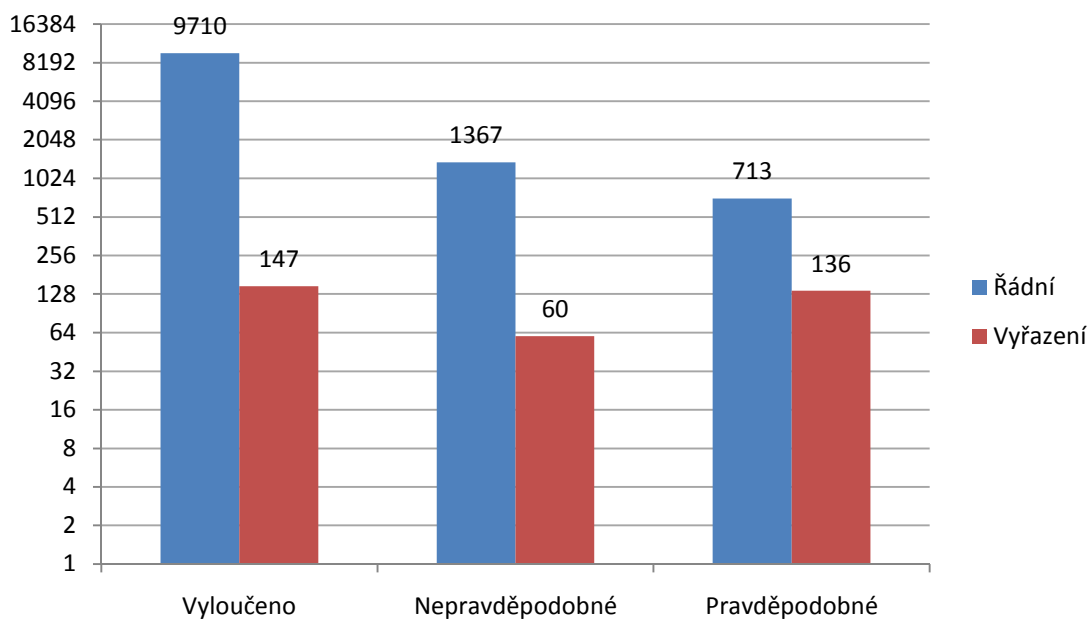
**Tabulka 1: Porovnání vah náhodně vybraných soutěžících**

### **6.2.2 Ročník 2014**

Soutěže se v 7. ročníku účastnilo 44 083 soutěžících, z toho se 12 133 soutěžících stalo úspěšnými řešiteli soutěže. Školní koordinátoři vyřadili ze soutěže 343 úspěšných řešitelů. Řádných úspěšných řešitelů tedy bylo 11 790.

V tomto roce aplikace podvádění vyloučila u 9 710 (tj. 82,36 %) řádných úspěšných řešitelů. U dalších 1 367 (tj. 11,59 %) řádných úspěšných řešitelů aplikace označila podvádění za nepravděpodobné a jako možné až pravděpodobné ho označila u 713 (tj. 6,05 %) řádných úspěšných řešitelů.

Mezi úspěšnými řešiteli vyřazenými koordinátory bylo podvádění vyloučené u 147 (tj. 42,86 %) z celkového počtu 343. Nepravděpodobné podvádění bylo shledáno u 60 (tj. 17,49 %) úspěšných řešitelů a jako pravděpodobné bylo určeno u 136 (tj. 39,65 %) řádných úspěšných řešitelů. Tato data můžete vidět na grafu 3.



**Graf 3: Podvádění u řádných a koordinátory vyřazených úspěšných soutěžících**

Dále proběhlo zkoumání, z jakých důvodů mohli být úspěšní soutěžící, u kterých bylo podvádění vyloučeno, vyřazení koordinátory. Ze 147 takových řešitelů bylo 8 soutěžících identifikováno jako učitelé, dalších 6 soutěžících zadalo nesmyslné údaje, 59 soutěžících bylo vyřazeno z důvodu soutěžení v nižší věkové kategorii a 42 soutěžících bylo vyřazeno z důvodu, kdy koordinátor nepotvrdil celou třídu. U zbývajících 32 soutěžících nebylo možné z dostupných dat důvod vyřazení identifikovat.

## 7 Závěr

Byla vytvořena aplikace na odhalování podvádějících soutěžících v soutěži Bobřík informatiky. Tato aplikace byla sestavena podle návrhu implementace odhalování rizikových jevů vyskytujících se v soutěži. Nejprve bylo uvažováno nad tím, že jednotlivé povolené rozsahy, jako jsou například čas, o který může být soutěžící rychlejší ve skládání testu nebo o kolik bodů může být jedna kategorie lepší, budou řešeny pomocí statistiky a normálního rozdělení, ale nakonec byly použity osvědčené hodnoty z předchozího zkoumání.

Kromě aplikace, jejímž cílem je odhalování podvádějících účastníků v soutěži, byl také vytvořen modul, který slouží k zobrazování výstupů této aplikace. Obě tyto části byly umístěny přímo na server soutěže Bobříka informatiky (obě složky test a vyhodnocení se umisťují do kořenového adresáře soutěže a potřebné soubory se přepisují). Na něm tedy lze po dokončení aktuálního ročníku soutěže spustit aplikaci pro odhalování podvodníků bez nutnosti exportovat databázové tabulky, a následně v administrátorském prostředí testu pohodlně prohlížet výstupy aplikace.

Z testování aplikace, jak po stránce nasimulování situace, kdy jedinci nebo skupina soutěžících dosahovala úspěšného zdolání soutěže pomocí podvodných praktik, tak po stránce porovnání výsledků na ročníku konaném v roce 2013, můžeme usoudit, že aplikace správně pracuje s daty a úspěšně odhaluje v ní zahrnuté způsoby podvádění.

## Seznam použité literatury a zdrojů

- [1] VANÍČEK, J. Rozvoj inforatických kompetencí žáků pomocí soutěže: kauza Bobřík informatiky [online]. In: *Sborník konference Počítač ve škole*. Nové Město na Moravě: Gymnázium Vincence Makovského, 7.-9. 4. 2009. : autor neznámý. Dostupné z: <http://www.pocitacveskole.cz/system/files/>.
- [2] *Bobřík informatiky: Informatická soutěž pro žáky základních a středních škol [online]*. [cit. 2015-06-23]. Dostupné z: <http://www.ibobr.cz/>.
- [3] MAREŠ, J.. *Tradiční a netradiční podvádění ve škole*. *Pedagogika*. 2005, roč. 55, č. 4, s. 310–335. ISSN 0031-3815.
- [4] ROWE, N. C. Cheating in Online Student Assessment: Beyond Plagiarism. In: *Distance Learning Administration*. 2004, roč. 7, č. 2. Dostupné z: <http://www.westga.edu/~distance/ojdla/summer72/rowe72.html>.
- [5] CHAMPMAN, C. *Cheats Who Phone a Friend Vaught on Screen*. *Times Higher Education Supplement*. 2004, 164, s.17. ISSN 0049-3929.
- [6] LEIBOWITZ, W.R. Colleges Start to Ask Wheather Students Should Use Laptops in Test. *Chronicle of Higher Education*. 1999, 46, 17, s. 12. ISSN 0009-5982.
- [7] CIZEK, G.J. *Detecting and Preventing Classroom Cheating*. Thousand Oaks: Crowin Press, 2003. 167 s. ISBN 0-7619-4655-1.
- [8] UNDERWOOD, J. a SZABO, A. Acadcmic Offences and E-learning: Individual Propensities in Cheating. *British Journal of Educational Technology*. 2003, 34, 4, s. 467-477.
- [9] PRUITT-MENTLE, D. Plagiarism in the 21<sup>st</sup> Century: Paper Mills, Cybercheating, and Internet Detectives in the Electronic Age. *Educational Technology Outreach*. University of Maryland, 2004.
- [10] ŠIMANDL, V. Odhalování podvádění v online soutěžích. *Journal of technology and information education = Časopis pro technickou a informační výchovu*. 2014, : 92-99. ISBN 1803-537X. ISSN 1803-537X.
- [11] File-Extensions.org [online]. 2010 [cit. 2015-06-22]. File extension XLS. Dostupné z: <http://www.file-extensions.org/xls-file-extension>.
- [12] ŠIMANDL, V. Ústní sdělení na katedře informatiky.
- [13] *Ministerstvo školství, mládeže a tělovýchovy: Program Excellence* [online]. [cit. 2015-06-24]. Dostupné z: <http://excellence.msmt.cz/>

[14] *PHP: Hypertext Preprocessor* [online]. [cit. 2015-06-24]. Dostupné z:  
<http://php.net/manual/en/function.set-time-limit.php>



## Seznam obrázků

Obrázek 1: Diagram důležitých stávajících tabulek soutěže .....	13
Obrázek 2: Diagram zpracovávání dat.....	22
Obrázek 3: Chyba sítě při spuštění aplikace přes terminál.....	28
Obrázek 4: Náhled vytížení serveru při běhu aplikace .....	29
Obrázek 5: Náhled přidání položky Podezřelí soutěžící do menu .....	30
Obrázek 6: Náhled vrchní části stránky po vytvoření pohledu bez sloupce Validován .	32
Obrázek 7: Náhled vrchní části stránky v situaci, kdy nebyl vytvořen pohled .....	32
Obrázek 8: Náhled vrchní části stránky při zobrazování bez sloupce Validován .....	34
Obrázek 9: Náhled zobrazovaných sloupců - bez sloupce "Validován" .....	35
Obrázek 10: Náhled zobrazovaných sloupců - se sloupcem "Validován" .....	35
Obrázek 11: Seznam se zvoleným filtrem "Váha <= 9" .....	36
Obrázek 12: Náhled lišty s použitými filtry - bez filtru "Validován" .....	36
Obrázek 13: Náhled lišty s použitými filtry - s filtrem "Validován" .....	36
Obrázek 14: Tabulka výsledků při testování aplikace .....	41

## **Seznam grafů**

Graf 1: Podvádění u řádných úspěšných soutěžících, rok 2013 .....	42
Graf 2: Podvádění u koordinátory vyřazených úspěšných soutěžících, rok 2013 .....	43
Graf 3: Podvádění u řádných a koordinátory vyřazených úspěšných soutěžících.....	45

## **Seznam tabulek**

Tabulka 1: Porovnání vah náhodně vybraných soutěžících.....	44
-------------------------------------------------------------	----