UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

WWW aplikace pro správu třídního fondu

Roman Svoboda

Bakalářská práce
2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2009/2010**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Roman SVOBODA**
Osobní číslo:	**I07794**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**WWW aplikace pro správu třídního fondu**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Teoretická část se bude zabývat rozborem možností zabezpečení dat uložených v databázích a využívaných WWW servery před neoprávněnými přístupy prostřednictvím Internetu včetně problematiky SQL Injection.
Implementační část obsahuje návrh a tvorbu internetové aplikace s využitím relačních databází pro správu třídního fondu.

Tato WWW aplikace musí umožnit:
* registraci libovolné školy a následně registraci libovolné třídy této školy včetně žáků třídním učitelem
* evidenci plateb žáků do třídního fondu
* efektivní vkládání informací o čerpání finančních prostředků (společné akce školy či třídy, individuální čerpání pro jednoho žáka, omezená skupina účastníků akce, různé kombinované modely, atd.)
* náhled žáků či jejich rodičů na výši jejich příspěvků a čerpání z fondu
* generování pdf a xls souborů pro písemnou archivaci dat a tvorbu peněžního deníku
* automatické generování elektronické výzvy k vložení nového příspěvku při poklesu "osobního" účtu pod nastavenou mez

Všechny přístupy nutno zabezpečit před neoprávněnými uživateli přiměřeným způsobem. Důležitými aspekty při návrhu aplikace jsou správný databázový model a jednoduchý způsob obsluhy pro učitele i rodiče dětí.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

***Castagnetto, J. a kol. Programujeme PHP profesionálně. Computer Press, 2004.**

***Kout, P. Praktický JavaScript. Zoner Press, 2004.**

***Riordan, R.M. Vytváříme relační databázové aplikace. Computer Press, 2001.**

***Ullman, L. PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek. Computer Press, 2004.**

Vedoucí bakalářské práce: **RNDr. David Žák, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 2. 5. 2010 Roman Svoboda

Poděkování

Zde bych velice rád poděkoval panu RNDr. Davidu Žákovi, Ph.D. za pomoc a cenné připomínky při vypracovávání bakalářské práce. Dále nesmím opomenout svou rodinu a přátelé, kteří mě během studia podporovali. Děkuji.

Anotace

Bakalářská práce se v teoretické části zabývá rozborem možností zabezpečení dat uložených v databázích před neoprávněnými přístupy prostřednictví internetu včetně problematiky SQL Injection. Praktická část popisuje vytvořenou www aplikaci pro kompletní správu třídního peněžního fondu vhodného pro většinu škol v České republice. Aplikace byla vytvořena za využití technologií: HTML, PHP, CSS, JavaScript a databázového serveru MySQL.

Klíčová slova

třídní fond, MySQL, PHP, SQL Injection, zabezpečení

Title

WWW application for administration of class monetary fund

Annotation

In theoretical part the bachelor thesis deals with analysis of securing the data stored in database against unauthorized access through the Internet including SQL Injection. Practical part describes the created www application for complete administration of class monetary fund suitable for majority of schools in Czech Republic. Application was designed using these technologies: HTML, PHP, CSS, JavaScript and DBMS MySQL.

Keywords

class fund, MySQL, PHP, SQL Injection, security

Obsah

Seznam zkratek

CSS	Cascading Style Sheets
ER	Entity-Relationship
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
MD5	Message-Digest Algorithm 5
PDF	Portable Document Format
PHP	Personal Home Page (Hypertext Preprocessor)
SHA	Secure Hash Algorithm
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
UTF	Unicode Transformation Format

Seznam obrázků

1 Úvod

Teoretická část bakalářské práce je zaměřena na zabezpečení dat uložených v databázích před neoprávněným přístupem prostřednictví internetu včetně problematiky SQL Injection.

Praktická část bakalářské práce obsahuje nejdůležitější a nejzajímavější informace týkající se tvorby webové aplikace Třídního fondu a celkový popis její funkcionality.

Teoretická část

2 Náplň teoretické části

V teoretické části se budu zabývat rozborem možností zabezpečení dat uložených v databázích a využívaných WWW servery před neoprávněnými přístupy prostřednictvím Internetu včetně problematiky SQL Injection. Vysvětlím zde nejdůležitější zásady tvorby bezpečných webových stránek využívajících databáze. Většina uvedených příkladů bude zaměřena na spolupráci MySQL databáze s programovacím jazykem PHP a to z důvodu, že se v dnešní době jedná asi o nejvíce používanou kombinaci při tvorbě webu. Všechny příklady budou vždy principiálně vysvětleny a po menší obměně by bylo možné je využít i na další databázové systémy.

2.1 PHP Injection

PHP injection je velice starý způsob hackování www stránek [1]. Jedná se o chybu logiky vkládání souborů do www stránky za pomoci některé z funkcí include či require. Této chyby se dopouští převážně začínající programátoři, chyba umožňuje útočníkovi vložit na server jakýkoliv script.

Typická adresa stránky generující obsah podle parametrem GET předávaného souboru vypadá takto:

```
http://mojestranky.cz/index.php?promenna=stranka.php
```

Nebezpečný kód, který může obsahovat index.php vypadá například takto:

```
<? include($promenna); ?>
```

Pokud útočník napíše nebezpečný script, zveřejní ho na Internetu a vloží jeho URL adresu do naší stránky (jako je tomu níže), bude nejspíše PHP Injection úspěšná.

```
http://mojestranky.cz/index.php?promenna=http://web.cz/zakernyScript.txt
```

Proti tomuto druhu injection je jednoduchá obrana. První a nejbezpečnější způsob je **nepoužívat vůbec include souborů pomocí proměnných**.

Pokud je to opravdu nutné, velice výhodným řešení je použít switch [1]:

```
<?
switch ($promenna)
{
  case "novinky.php":
    include("novinky.php");
    break;
  case "clanky.php":
    include("clanky.php");
    break;
  // atd.
 default:
   //V případě, že proměnná obsahuje něco jiného než novinky.php nebo
                clanky.php, se zobrazí hlášení o chybě.
    echo "Chyba";
    break;
}
?>
```

Tento script kontroluje, zda proměnná odpovídá konkrétním hodnotám. Podle vložené hodnoty se vykoná include správné www stránky. Pokud někdo proměnnou neoprávněně změní, provede se sekce `default`.

Další nezbytnou, jednoduchou a velice účinnou obranou je nastavení direktivy `safe_mode ON`.

2.2 SQL Injection

„Pod pojmem SQL injection se skrývá podvržení vstupních dat (hodnot proměnných odesílaných serveru) tak, aby byl nějakým způsobem pozměněn výsledek SQL dotazu. Pokud útočník zná strukturu tabulky (nejlépe i SQL dotazů), které svými proměnnými ovlivní, má vše daleko jednodušší, než když musí odhadovat, jaké sloupce jsou použity, jaké mají asi datové typy a jak se jmenují. Proto je nesmírně důležité, aby Vaše skripty podávaly co nejméně informací o struktuře databází a tabulek v případě, že se vyskytne nějaká chyba. Osvědčený způsob je volat vždy v případě neúspěchu svou funkci, která zjistí, jestli může zobrazit informaci o chybě (například když má klient IP adresu 127.0.0.1, což znamená, že spouštíte aplikaci na svém počítači) anebo ne.“ [2]

2.2.1 Co hrozí

- Útočník může získat přístup k citlivým datům – nick, heslo, email atd. [2].
- Je možné, že se útočníkovi podaří přihlásit na administrátorský či jakýkoliv jiný účet.
- Dále není vyloučena ani modifikace nebo dokonce smazání všech dat z databáze.

2.2.2 Co nehrozí (v MySQL ve spojení s PHP)

- Nehrozí nám zde provedení více SQL dotazů najednou [2]. Například problém s řetězcem typu `'; truncate tabulka;`, který by vymazal data z tabulky, úplně odpadá, protože PHP v jednom volání funkce `mysql_query()` nepodporuje více SQL dotazů. Provede se vždy jen příkaz před prvním středníkem.
- Spousta potencionálně nebezpečných dotazů odpadá také díky tomu, že v MySQL neexistuje způsob, kterých bychom mohli volat externí aplikaci, jako je tomu například u alternativy od Microsoftu – MSSQL.
- Pokud v databázi neukládáme zdrojový kód, který by se poté funkcí eval() prováděl, tak nám nehrozí ani jeho vypsání, ani provedení cizího kódu na našem serveru.
- V případě, že nemáme použita všude stejná hesla, nehrozí ani prozrazení přístupu k FTP.

2.2.3 Příklad č. 1 (přihlašovací formulář)

Přihlašovací formuláře jsou dnes na internetu v podstatě všude. Níže bude vysvětlen příklad SQL Injection za účelem přihlášení se na administrátorský účet [3].

Nejprve si uvedeme asi nejběžnější příklad ověřování přihlašovacích údajů:

```
SELECT * FROM users WHERE log='$nick' AND password='$passwd'
```

Příklad je vhodný pro ilustraci, ovšem ve skutečnosti, pokud vycházíme z předpokladu, že před provedením dotazu nebyly vstupy nijak ošetřeny, je to naprostá katastrofa.

Představme si, že za vstupní proměnnou `$nick` dosadíme například řetězec:

```
'--
```

Řetězec by nám modifikoval naše ověření na:

```
SELECT * FROM users WHERE log=''--' AND password=''
neboli
SELECT * FROM users WHERE log=''--
```

Apostrof nám zajistil ukončení vstupní proměnné a znak dvou pomlček (v MySQL komentář) odstřihl zbytek dotazu. Nyní stačí dále modifikovat vstupní proměnnou tak, aby podmínka za klauzulí WHERE byla vždy platná.

Zkusme tedy do vstupní proměnné `$nick` vložit řetězec:

```
admin' OR 1=1--
```

Výsledkem bude:

```
SELECT * FROM users WHERE log='admin' OR 1=1--
```

Vložením pouze řetězce `admin'--` bychom mohli nejprve vyzkoušet, zda existuje uživatel s nickem admin, avšak přidáním `OR 1=1` zajistíme, že i v případě, kdy admin existovat nebude, podmínce stejně vyhovíme, protože `1=1` vždy [3].

Tímto jsme se s největší pravděpodobností přihlásili na uživatelský účet admina, pokud by neexistoval, jsme přihlášeni na první osobu v tabulce `users`, což zpravidla bývá právě admin nebo testovací uživatel s plnými právy [3].

Možností pro správnou záměnu přihlašovacích řetězců je více [3]:

```
OR 1=1–
" OR "p"="p
" or 2=2--
') or ('p'='p
' or 'p'='p
```

Způsob obrany

V případě, že by v dotazu za klauzulí `WHERE` bylo nejprve heslo, můžeme aplikovat stejný princip prolomení. Ovšem proti napadení skrz proměnnou `$heslo` existuje velice jednoduchý způsob obrany – nikdy nevkládat do databáze hesla v čisté podobě. Vždy je třeba hesla hashovat (podrobnější vysvětlení dále v BP).

Co se týká proměnné `$nick`, tak je několik možností, jak zabezpečit správný vstup. Pokud máme přesnou představu, jak má nick vypadat, můžeme povolit pouze správný tvar pomocí regulárních výrazů. Další možností je escapovat (nahradit – vložit před ně \) všechny nebezpečné znaky pomocí nastavení direktivy `magic_quotes ON` nebo za použití PHP funkcí k tomu určených (např. `mysql_real_escape_string($nick)`).

2.2.4 Příklad č. 2 (výpis celého obsahu tabulky)

Zde si představíme způsob, jak pomocí neošetřeného dotazu do databáze vypsat celou tabulku i s daty v této tabulce [2].

Původní dotaz vypisující jméno a email jedné osoby s konkrétním id:

```
SELECT jmeno, email FROM uzivatel WHERE id = $id
```

Modifikace proměnné `$id` (zda zasíláme proměnnou metodou POST či GET není rozhodující) :

```
9 OR 1=1 --
```

Výsledný dotaz:

```
SELECT jmeno, email FROM uzivatel WHERE id = 9 OR 1=1 --
```

Takto modifikovaný dotaz nám zaručí, že podmínka bude vždy splněna. Pokud je navíc výpis tvořen cyklem, vypíše se nám celá tabulka `uzivatel`.

Způsob obrany

Pozor, ve výše uvedeném případě i přes nastavenou direktivu `magic_quotes ON` zmíněná modifikace projde přes nastavení serveru vždy [2]. Je tomu tak proto, že vložený řetězec neobsahuje žádné escapované znaky (' ani "), je tedy jen na nás, jak tento problém vyřešíme.

PHP nám pro tyto případy poskytuje funkce pro ověřování číselné hodnoty v proměnné, například funkce `is_numeric()`, `is_integer()`. Popřípadě umožňuje přetypování na číslo. Tudíž stačí jednoduchá podmínka, nebo přetypování proměnné před samotným vstupem do dotazu a nebezpečí je zažehnáno.

2.2.5 Příklad č. 3 (klauzule UNION)

Klauzule union umožňuje spojení dvou SQL dotazů do jednoho výpisu [2]. Níže si popíšeme možné nebezpečí, které plyne z jejího vsunutí do proměnné.

Původní SQL dotaz:

```
SELECT jmeno, email FROM uzivatel WHERE nick = '$nick' LIMIT 1
```

Hodnota vložená do proměnné `$nick`:

```
1' UNION SELECT heslo AS jmeno, nick AS email FROM uzivatel --
```

Výsledný dotaz:

```
SELECT jmeno, email FROM uzivatel WHERE nick = '1' UNION SELECT heslo AS
jmeno, nick AS email FROM uzivatel -- LIMIT 1
```

Původním dotazem vybereme jednoho uživatele (jeho jméno a email) a přidaným `UNION` dotazem získáme ostatní uživatele z tabulky. Vypíšeme si jejich hesla a nicky do sloupců `jmeno` a `email`. Omezení `LIMIT 1` jsme odstranili pomocí komentáře.

Způsob obrany

Pokud se uživateli podaří vložit takovýto řetězec a vypisujeme informace v cyklu, není již obrany. Musíme zajistit, že tento řetězec neprojde až k SQL dotazu, například znemožněním vkládání nebezpečných znaků nebo omezením délky vstupního řetězce [2].

2.2.6 Shrnutí zabezpečení

Nejdůležitější je nevěřit ničemu, co přichází od uživatele!

Obecně bychom měli dodržet co nejvíce z následujících bodů:

- ověřovat délku vstupních dat v proměnné (omezení délky řetězců u vstupního formuláře),
- u číselných proměnných ověřovat, zda jsou to opravdu čísla (`is_numeric()`, `is_integer()`), popřípadě je přetypovávat,
- pomocí regulárních výrazů nevpouštět nebezpečné (nechtěné) řetězce,
- znakům jako je ['] nebo ["] vůbec neumožnit vstup do SQL dotazu, nebo využívat escapovacích funkcí (`mysql_real_escape_string()`, `direktiva magic_quotes ON`),
- přidělovat uživatelům jen nejnutnější práva na práci s databází.

2.3 Zabezpečení hesel

Při komplikovanosti dnešních systémů a šikovnosti hackerů prakticky nelze vyloučit prolomení zabezpečení databáze [4]. Že v nepovolaných rukou skončí emailové adresy je sice velice nepříjemná věc, avšak mnohem větší nebezpečí může plynout ze seznamu uložených hesel. Jednak to dovolí útočníkovi získat v podstatě jakoukoliv identitu na daném serveru, ale také, pokud jsou uživatelé líní vymýšlet nová hesla, se může dostat k jejich účtům kdekoliv jinde.

Dnešní hackeři se dělí na dvě velké skupiny [4]. Jedna se svým výkonem pochlubí a data nijak nepoužije – v podstatě pomáhají zlepšit zabezpečení aplikací. O jejich morálce se velice často debatuje a píše. Druhou skupinou jsou lidé, kteří získají Vaše data, nikomu o tom neřeknou a nelegálně je využijí ve svůj prospěch. Tato skupina je diskutována méně, protože se většinou nikdo nedozví, že k nějakému úniku dat došlo.

2.3.1 Ukládání čistého hesla

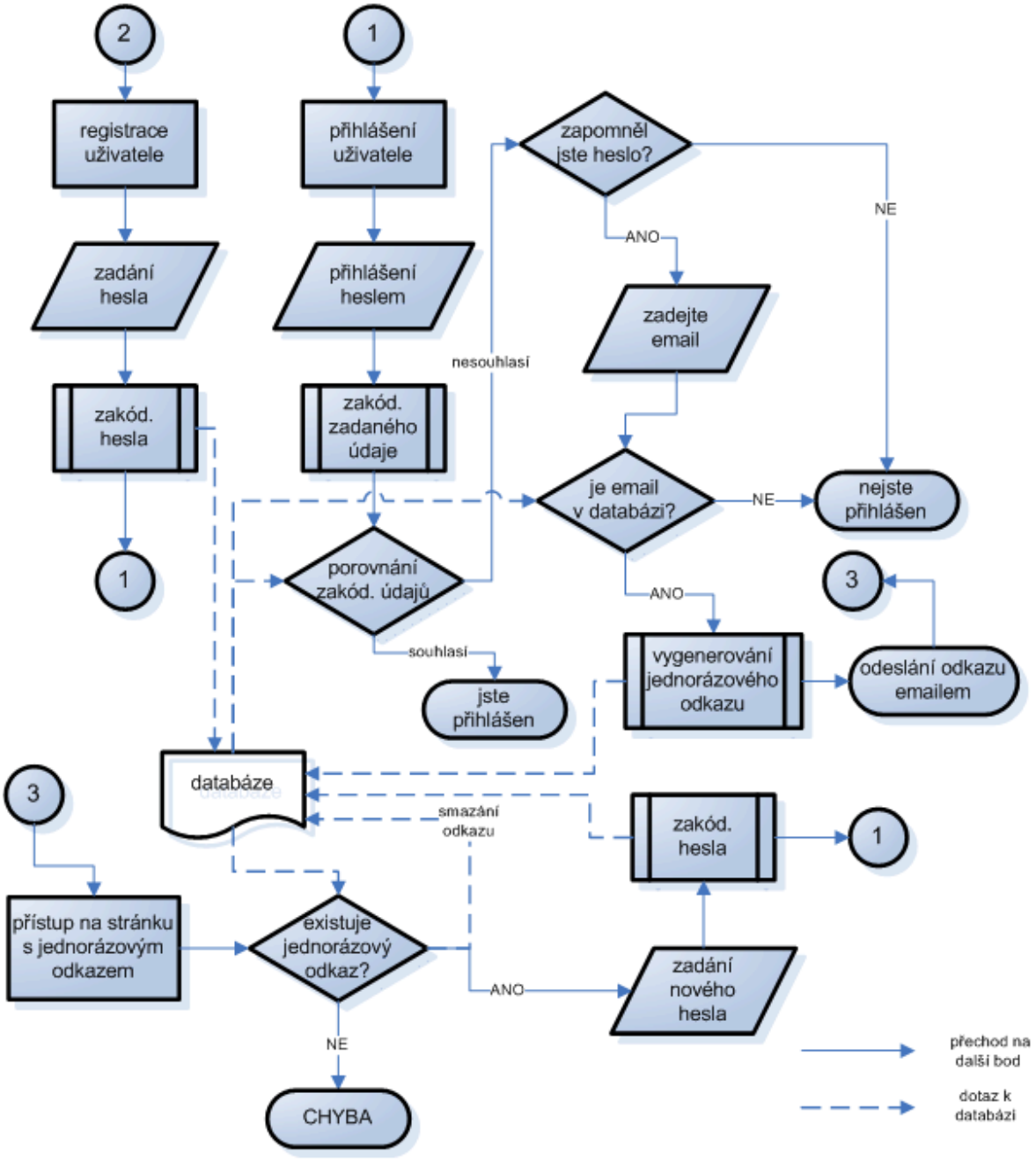
Ukládání čistého hesla má „výhodu“ jen pro případ zapomenutí hesla uživatelem. Administrátor může shlédnout záznam v databázi a heslo bez větších obtíží uživateli zaslat zpět. Toto je však jediná, ještě k tomu velice sporná výhoda. V případě že máme hesla ukládána do databáze v čisté podobě, plynou nám z toho již výše zmíněná a nezanedbatelná rizika.

2.3.2 Ukládání otisku (hashe) hesla

Hashování znamená jednosměrné zakódování neomezeně dlouhého řetězce do výchozího otisku (hashe) se vždy stejnou výchozí délkou [4]. Například md5 šifrování je 128bitové a sha1 160bitové. Po hashovacích funkcích požadujeme co nejkvalitnější

„rozemletí“ zprávy do výchozího řetězce. Pokud bychom v terabajtovém souboru změnili jediný bit, očekáváme, že se změní v průměru 50% bitů v otisku.

Jakým způsobem budeme porovnávat hesla při přihlašování uživatele? Odpověď je jednoduchá, neporovnáváme čistá hesla, ale rovnou jejich otisky. Jak poskytneme uživateli heslo v případě ztráty? I na tuto otázku následuje jednoduchá odpověď. Uživateli vložíme a odešleme nové heslo. Níže uvedený algoritmus by měl dostatečně vystihovat celou situaci.



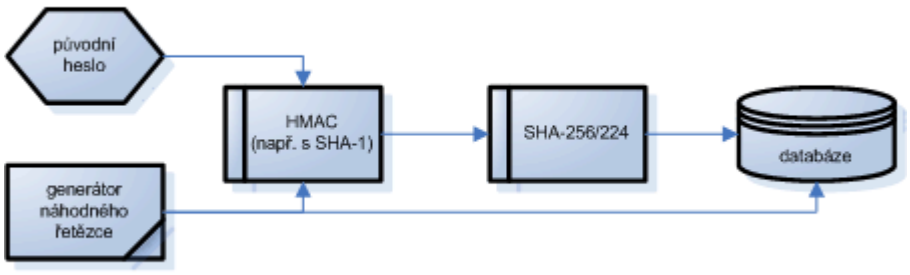
Obrázek 1 – Komplexní systém zabezpečení přístupových údajů, zdroj [5]

2.3.3 Kombinace hashů, nebo jejich iterace

Vkládané heslo může vypadat např. takto: `md5(sha1(heslo))`. Ukládání hesla do databáze uvedeným způsobem je o něco bezpečnější než jednoduchý hash. Pokud se bude útočník snažit prolomit šifrování, tak ve chvíli, kdy prolomí jeden otisk, nevznikne jím očekávané heslo, nýbrž pouze další hash.

2.3.4 Otisk se špetkou soli (v češtině třešničkou)

Nejbezpečnější způsob pro ukládání hesla do databáze je vytvoření hashe se špetkou soli [4]. Konkrétní příklad může vypadat například takto: `sha1('nějaký řetězec' + heslo)`. Lze tak ošetřit nedostatečně dlouhá a málo složitá hesla uživatelů. Útočníkovi bez znalosti naší soli velice zkomplikujeme situaci. Je tu sice varianta, při použití stále stejné třešničky, že šikovný kryptoanalytik časem odhalí naši sůl, ale je dosti malá. Nikdo také neříká, že musíme používat vždy stejnou sůl. V praxi bývá sůl náhodně generována a spolu s heslem bezpečným způsobem ukládána do databáze.



Obrázek 2 - Bezpečné uložení hesla pomocí hashování a soli, zdroj [5]

2.3.5 Nejobvyklejší způsoby útoků

Slovníkový útok

Slovníkový útok je útok takzvaně hrubou silou [6]. Tento druh útoku je založen na metodě pokus-omyl a vkládání hesel podle určitého slovníku. Povětšinou se za tyto slova přidávají i kombinace číslic.

Rainbow tables

Velké množství počítačů hashuje různé řetězce a tvoří z nich databázi [6]. Útočníkovi pak stačí porovnat Vaše otisky s otisky v databázi – pokud se shodují, zjistit přeložený řetězec.

2.3.6 Ideální heslo

Jak jsme si již vysvětlili, do databáze nesmíme nikdy ukládat hesla v čisté podobě. Nyní zbývá popsat, co by samotné heslo ještě před zašifrováním (v ideálním případě) mělo splňovat [7]:

- mělo by být alespoň osm znaků dlouhé,
- nesmí být spojeno s informacemi o uživateli (např. rodné číslo, jméno přítelkyně),
- nesmí v sobě obsahovat smysluplné slovo,
- musí obsahovat malá i velká písmena,
- obsahuje číslice a speciální znaky (např. !, ?, _, # apod.).

2.4 Zálohování

Zálohování dat, například dat v databázi, je dnes naprosto neodmyslitelnou součástí vyspělého počítačového světa. RNDr. David Žák uvedl na jedné ze svých přednášek, že **hodnotu dat si uvědomíme teprve v okamžiku, kdy o ně přijdeme!** Domnívám se, že všichni, kdo se nad tímto zamyslí, musí souhlasit. V případě, že někdo prolomí Vaše zabezpečení nebo se mu povede aplikovat jakýkoliv druh injection, je aktuální záloha velice důležitá. Níže přiblížím alespoň základní teorii o způsobech a druzích zálohování.

2.4.1 Základní dělení – dle úložiště

Fyzická záložní databáze

Fyzická záložní databáze využívá k zálohování a aktualizaci soubory, které jsou přenášeny pomocí síťových služeb [8]. Velkou výhodou tohoto druhu zálohování je, že soubory mohou být ukládány v případě potřeby na druhé straně planety.

Uvedu-li jako příklad velkou pražskou povodeň v roce 2002, tak právě za takovýchto okolností jsou zálohy uložené mimo Vaši kancelář velice užitečné.

Logická záložní databáze

Zde je prováděna aktualizace dat na úrovni SQL [8]. Část uživatelů pracuje aktivně na produkčním systému, zatímco jiní uživatelé, kteří pracují s daty např. pro dlouhodobější analýzu (výzkum chování zákazníka za poslední měsíc) bez požadavku na aktuální data, využívají záložní server.

2.4.2 Druhy aktualizace dat v záložní databázi

Synchronní aktualizace dat

Tento druh aktualizace záložní databáze zajistí aktualizaci dat přesně ve stejnou dobu, kdy je provedena změna v produkční databázi [8].

Asynchronní aktualizace dat

Asynchronní aktualizace dat znamená zápis do záložní databáze s jistým zpožděním [8]. To může být výhodné v případě selhání určitých operací či chyb samotného uživatele.

2.4.3 Způsoby zálohování

Kompletní zálohování

Tento způsob zálohování znamená zálohu všech dat i struktury databáze [8]. V případě dostatečně aktuální zálohy je k dispozici velice jednoduchý a rychlý způsob obnovy. Nevýhodou kompletní zálohy je velké množství přenášených dat, proto se většinou využívá u velkých databází v delších časových intervalech. Avšak u menších aplikací není s množstvím dat žádný problém.

Přírůstkové (rozdílové) zálohování

Narozdíl od kompletního zálohování se ukládají pouze změny od poslední zálohy [8]. Výhodou je menší přenos dat. Obnova probíhá nejprve nahráním kompletní základní zálohy a dále postupným nahráváním přírůstkových záloh.

Záloha transakčního žurnálu

V tomto případě se zálohuje pouze transakční žurnál, neboli transakce provedené od poslední zálohy [8].

2.5 Možnost zabezpečení pomocí .htaccess

Dle mého názoru je soubor .htaccess nepostradatelnou součástí zabezpečení proti neoprávněnému přístupu na webové stránky, a následně i k datům v databázi.

2.5.1 K čemu slouží soubor .htaccess

Htaccess je speciální soubor, díky kterému si uživatel hostingových služeb, který nemá plná práva ke změně konfigurace serveru (popřípadě nechce tuto změnu provádět na globální úrovni), může sám upravit některé vlastnosti daného serveru [9]. Za použití těchto nastavení můžeme zakázat zobrazování PHP chyb (jedna ze základních funkcionalit, která útočníkovi zkomplikuje prolomení ochrany), zapínat či vypínat direktivu magic_quotes

(escapování nebezpečných znaků), zabránit přístupu do určitého adresáře a mnoho dalšího. Základní nastavení si popíšeme níže.

Soubor .htaccess je použitelný na hostingových službách, které jsou založeny na serveru Apache [9]. Na Unixových systémech bude tento soubor díky tečce na začátku skrytý. Soubor platí vždy pro adresář, kde je umístěn a všechny jeho podadresáře.

2.5.2 Základní příklady využití

Zde uvedu opravdu jen několik základních příkladů. Soubor .htaccess má samozřejmě mnohem širší možnosti a lze v něm nastavit prakticky vše, co může ovlivnit administrátor v konfiguračním souboru httpd.conf.

```
#definice vlastní chybové stránky
ErrorDocument 404 /chybovaStranka.html

# nastaveni výchozí stránky adresáře
DirectoryIndex start.html index.html index.php

# zákaz výpisu adresářové struktury ve webovém prohlížeči
#pokud vypustíme znaménko – bude výpis povolen
Options -Indexes

#takto povolíme escapování znaků [‘] a [“]
php_flag magic_quotes_gpc on

#tímto příkazem vypneme (nebo zapneme [on]) zobrazování PHP chyb
php_flag display_errors off

#takto můžeme i při vypnutém zobrazování PHP chyb logovat chyby do
souboru
php_flag log_errors on
php_value error_log /cesta/k/souboru/soubor

#vypne výpis chyb ve tvaru HTML (v případě logu do souboru, se budou
záznamy lépe číst)
php_flag html_errors off

#znepřístupnění celého adresáře všem (aplikace s ním může pracovat, jen
zvenku se k němu nikdo nedostane)
deny from all

#zpřístupnění celého adresáře všem
allow from all

#povolení přístupu z konkrétní IP adresy
allow from 215.123.156.189

#zákaz přístupu z rozsahu IP adres
deny from 215.123
```

Praktická část

Praktická část bakalářské práce se zabývá rozborem důležitých milníků při tvorbě aplikace a popisem její konečné funkcionality.

3 Analýza třídního fondu

Cílem bakalářské práce je vytvořit internetovou aplikaci pro komplexní správu třídního fondu. Aplikace bude obsahovat tři základní role: neregistrovaného uživatele, třídního učitele a studenta (rodiče studenta).

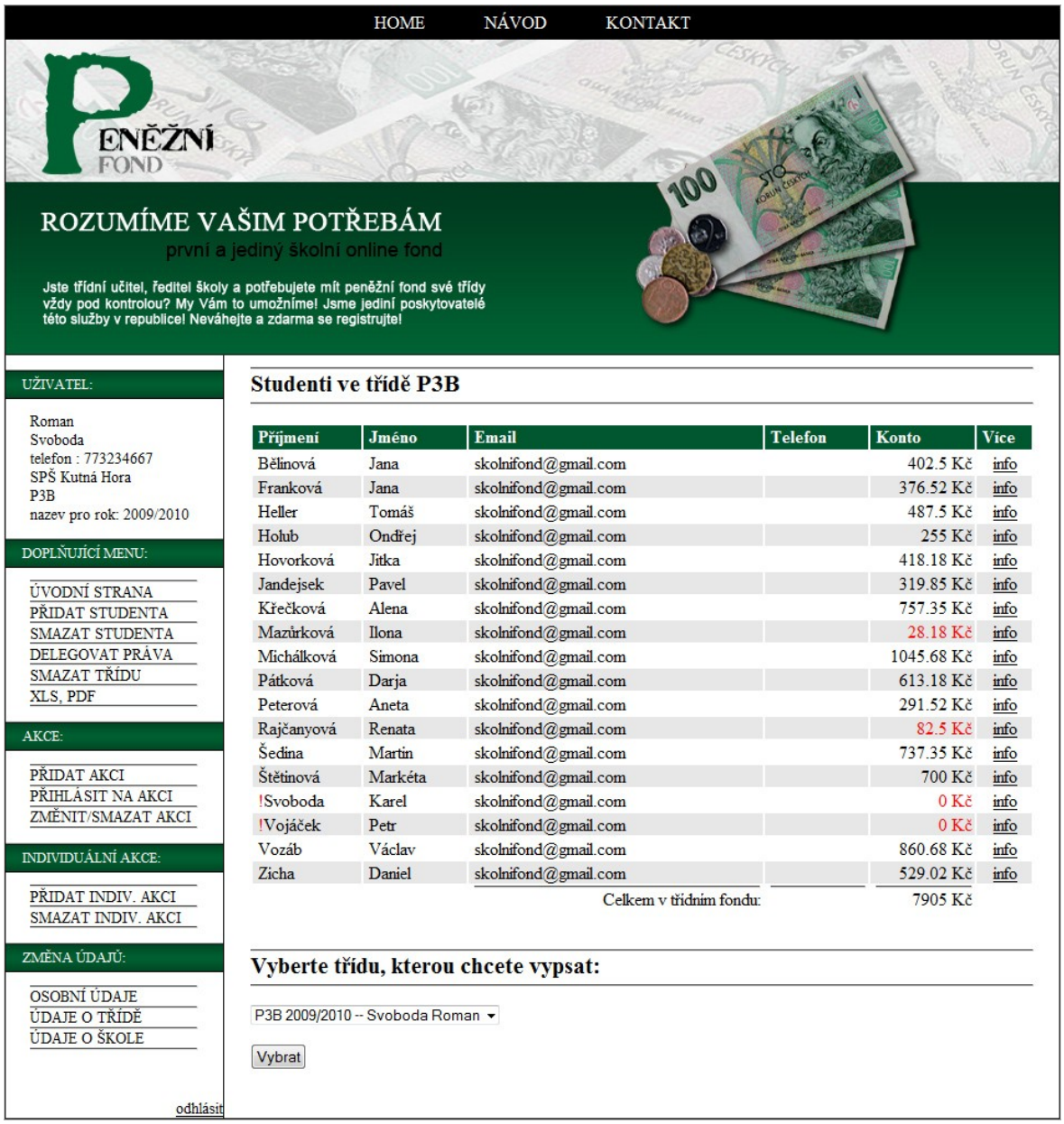
3.1 Požadavky na aplikaci (role uživatelů)

Neregistrovanému uživateli bude vysvětlena funkčnost celé aplikace. V případě, že má zájem, musí mu být umožněna registrace školy (třídy) v jakékoliv obci České republiky. Tento požadavek bude splněn importem všech obcí ČR z databáze Českého statistického úřadu (platné pro rok 2009).

Třídní učitel musí mít k dispozici plnohodnotnou správu třídního fondu, vkládání a mazání studentů, vkládání a úpravy individuálních i hromadných akcí, jednoduché a intuitivní přihlašování studentů k akci a neustálý přehled o cenách těchto akcí. Aplikace musí přehledně vypisovat všechny požadované informace o studentech, jejich čerpání a zůstatcích na kontu. Dále aplikace musí zajistit automatické varování rodiče studenta v případě, že stav účtu uživatele (studenta) klesl pod předem nastavenou hraniční mez. Informace o nedostatečném kreditu je samozřejmě třeba přehledně a automaticky vypisovat i třídnímu učiteli. Učitel by měl mít neustále k dispozici souhrnné informace o celé jeho třídě (jako podklady pro třídní schůzky, tvorbu peněžního deníku atd.). Tuto skutečnost aplikace zabezpečí pomocí generování PDF a XLS dokumentů. Pro případ nepřítomnosti uživatele je nutná možnost delegování práv na vkládání akcí i jinému kolegovi ze stejné školy. V neposlední řadě bude moci třídní učitel měnit své osobní údaje, informace o jeho třídě, hodnotu minimálního zůstatku studentů ve třídním fondu, a pokud je zakladatelem školy, tak i informace o škole. V případě smazání účtu studenta či celé třídy aplikace vygeneruje email s konečnými zůstatky, které budou zaslány jak jednotlivým studentům, tak třídnímu učiteli.

Student (jeho rodič) by měl mít neustálý přehled o čerpání, vkládání a všeobecném pohybu na kontu. Toto aplikace zabezpečí výpisem na www stránkách, ale i za pomoci PDF souboru. Dále je vhodné, aby mohl shlédnout informace o škole, třídě, třídním učiteli a jeho spolužácích. Změna osobních údajů je samozřejmostí.

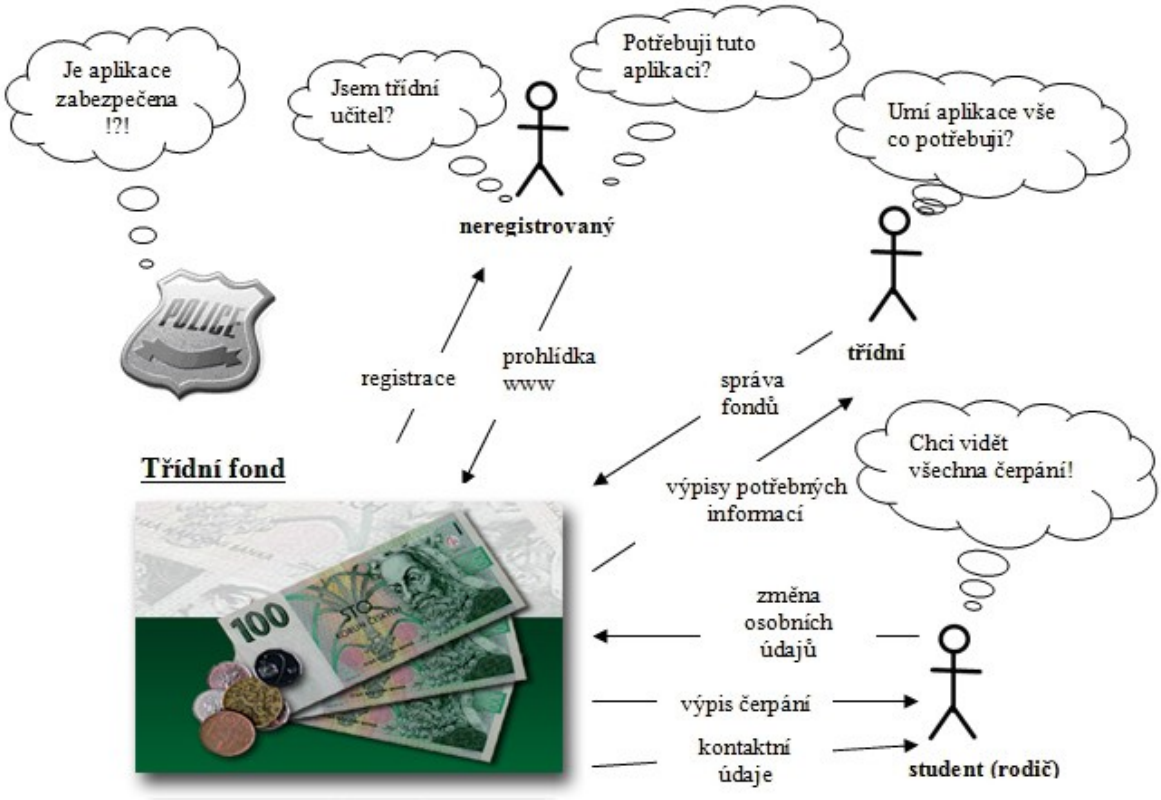
3.2 Vzhled aplikace



Obrázek 3 – Vzhled aplikace

3.3 Rich Picture diagram

Rich picture je nástrojem sloužícím k zachycení a výstižnému vyjádření určité situace [10]. Diagram nemá žádnou formální definici – slouží pro identifikaci a vymezení hranic systému. Měl by zachycovat uživatelské role, jejich potřeby a interakci s okolím.

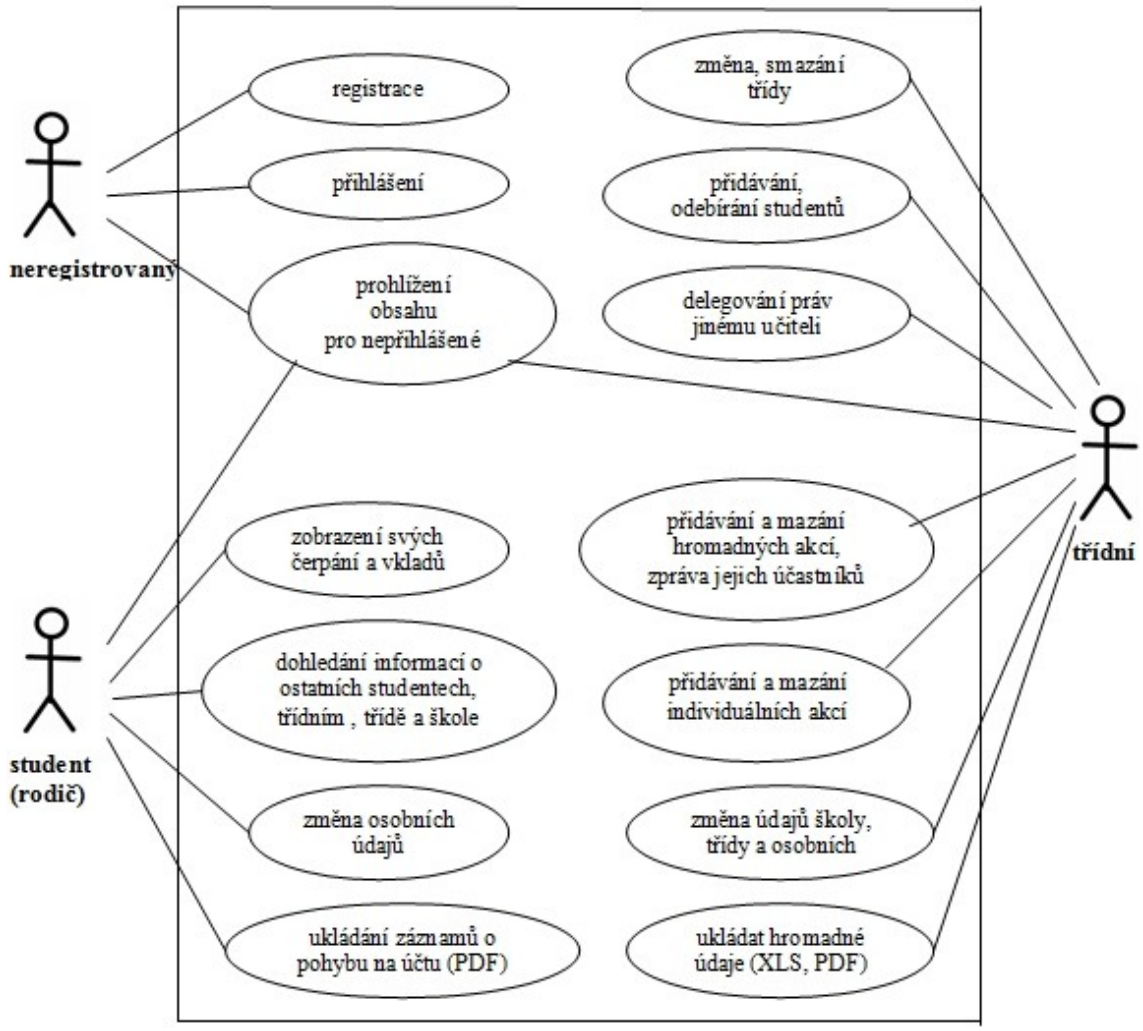


Obrázek 4 – Rich picture diagram

Konkrétně diagram vytvořený pro školní fond zobrazuje role uživatelů v systému, jejich možné myšlenkové pochody a otázky, které si mohou pokládat. Dále zde obrázek policejního odznaku připomíná velice důležitou složku aplikace, kterou je zabezpečení.

3.4 Use Case diagram

Diagram užití zachycuje vnější pohled na modelovaný systém a tím pomáhá odhalit hranice systému [10]. Jde o posloupnost souvisejících transakcí mezi účastníkem a systémem.

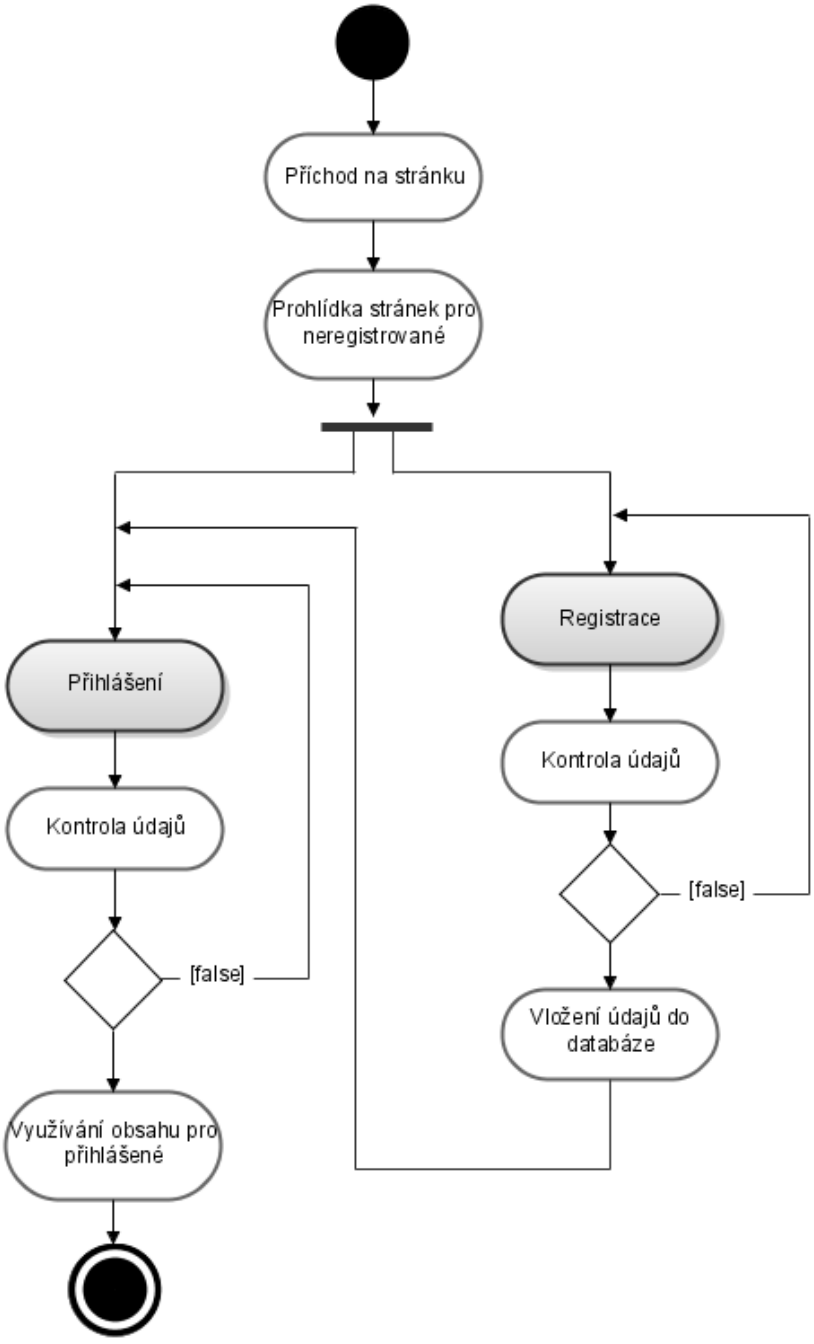


Obrázek 5 – Use Case diagram

3.5 Activity diagram – příchod na www

Diagram aktivit popisuje jednotlivé procesy pomocí aktivit reprezentujících jeho akční stavy a přechody mezi nimi [10].

Activity diagram, který je níže, popisuje velmi zjednodušeně možné chování uživatele a průběh systému při vstupu na www stránky této aplikace.



Obrázek 6 – Activity diagram příchodu na stránku

4 Teoretické otázky před tvorbou aplikace

Po promyšlení základní funkcionality aplikace vyvstalo několik otázek, na které bylo nutné odpovědět ještě před samotným začátkem tvorby. Ty nejzásadnější a nejzajímavější i s jejich výsledky rozeberu níže.

4.1 Oracle x MySQL

První otázkou, která nastala před tvorbou aplikace, byla volba databázového serveru. Na univerzitě jsme absolvovali dva semestry předmětu Databázové systémy, kde se pracovalo se serverem Oracle.

4.1.1 Oracle

Oracle je bezesporu nejsilnějším databázovým nástrojem s nejvíce možnostmi na nynějším trhu, ale nic není zadarmo. Pro Oracle databázi mluvilo mnoho aspektů, z nichž mohu vyjmenovat například velkou stabilitu, silnou podporu PL/SQL a v neposlední řadě bohaté zkušenosti na základě absolvovaných semestrů výuky. Tento databázový systém využívají největší firmy na celém světě, ovšem pro internetovou aplikaci v rámci bakalářské práce je v podstatě nemožné získat zdarma (za rozumnou cenu) hosting, který by tento druh databáze podporoval.

4.1.2 MySQL

Oproti tomu MySQL je asi nejrozšířenějším databázovým systémem pro střední a malé projekty. Je to způsobeno právě jeho dostupností. Většina hostingových společností nabízí MySQL databáze v různých verzích již jako standard. Podpora PL/SQL není taková jako u Oraclu, ovšem pokud jde o princip funkčnosti, má MySQL na českém internetu velkou podporu v komunitě jeho uživatelů. Neposlední faktorem, který mluvil pro MySQL, byla jeho přímá podpora z programovacího jazyka PHP.

4.1.3 Výsledek

Po pečlivém zvážení všech faktorů, především pro možné nasazení www aplikace do online provozu, jsem i přes fakt, že s ním nemám dosud žádné zkušenosti, zvolil databázový server MySQL.

4.2 Druh tabulek – MyISAM x InnoDB

„Zvláštností MySQL je, že při vytváření nové tabulky se musí zadat její typ. MySQL podporuje různé typy tabulek, které se navzájem liší řadou vlastností. Mezi nejdůležitější typy tabulek patří MyISAM, InnoDB a HEAP.“ [11]

4.2.1 Tabulky typu MyISAM

„MyISAM představuje v MySQL standardní typ tabulek. Jedná se o stabilní, vyspělý a jednoduše spravovatelný typ tabulek. Pokud nemáme žádný zvláštní důvod, proč použít jiný typ, většinou použijeme tento.“ [11]

Příklad tvorby tabulky MyISAM:

```
CREATE TABLE example (
         id INT,
         data VARCHAR(100)
       );
```

4.2.2 Tabulky typu InnoDB

Typ InnoDB je v porovnání s předchozím typem nový. Podporuje všechny vlastnosti typu MyISAM, navíc se ještě pyšní dvěmi odlišnostmi [11]:

- databázové operace v tabulkách InnoDB se dají spouštět jako transakce, což je v mnoha případech bezpečnější a rychlejší,
- tabulky InnoDB podporují používání pravidel integrity (pravidla cizího klíče atd.).

Bohužel existují i důvody, které hovoří proti používání InnoDB tabulek [11]:

- InnoDB ještě nedosáhl tak vysoké stability jako MyISAM (v době vydání publikace),
- u tabulek InnoDB nemůžeme vytvářet fulltextový index,
- správa tabulek InnoDB je o něco složitější (ovšem to programátorům v PHP může být docela jedno),
- komerční licence v MySQL s podporou InnoDB je dvojnásobně dražší než verze bez ní.

Příklad tvorby tabulky InnoDB:

```
CREATE TABLE example (
         id INT,
         data VARCHAR(100)
       ) ENGINE=InnoDB;
```

4.2.3 Výsledek

Vzhledem k faktu, že pravidla integrity jsou jednou z nejzásadnějších věcí z pohledu bezpečnosti databáze a k faktu, že v aplikaci budu využívat transakce, byl konečný verdikt naprosto jasný. Ve vývoji přistoupím k práci s tabulkami InnoDB. K tomu, co jsou to transakce, se vrátím v dalších kapitolách bakalářské práce.

4.3 Funkce pro přístup k MySQL pomocí PHP – mysql x mysqli

Rozhraní mysql je pro přístup k databázi velice známe a podporované v PHP již od dávných dob [11]. Od PHP 5 se nám ovšem nabízí nové rozhraní a to mysqli. Po přečtení několika článků o této problematice jsem zjistil, že tvůrci mysqli slibují rychlejší přístup do databáze, efektivnější správu paměti a kromě všeho, co bylo v mysql i nemálo nových funkcí. Například funkce pro předpřipravené databázové dotazy (prepared statements), zpracovávání více databázových dotazů (ukládaných procedur) najednou a další.

Mysqli lze použít jak ve strukturovaném, tak v objektově orientovaném programování [11]. V případě objektově orientovaného programování zpřehledňuje kód. Nesmíme ale opomenout, že mysqli je podporováno až od PHP verze 5 a MySQL 4.1 a vyšší.

4.3.1 Výsledek

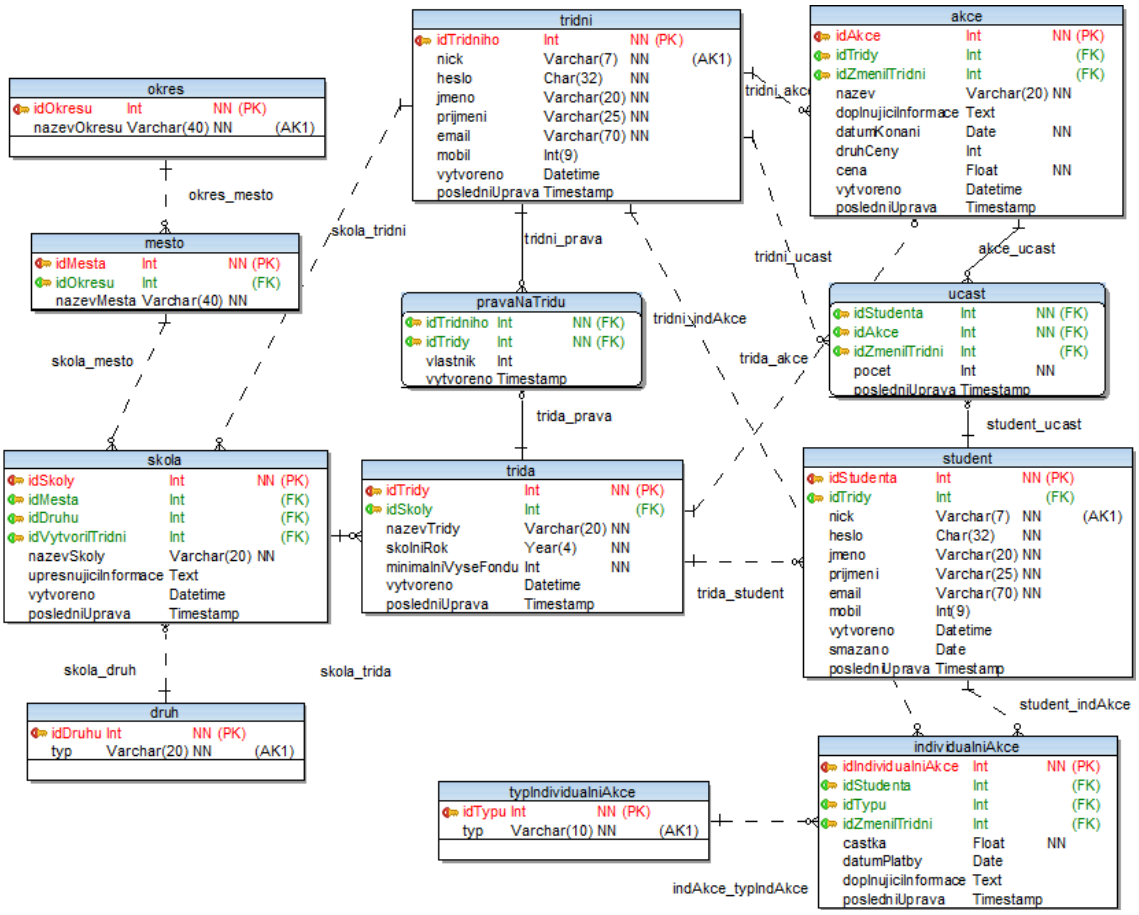
Nakonec jsem se rozhodl k využití rozhraní mysqli. Motivace k tomuto rozhodnutí byla v alespoň částečném zpřehlednění zdrojového kódu a učení se novým, modernějším postupům.

5 Návrh databáze

Kvalitně a přehledně navržená struktura databáze je jednou ze základních podmínek pro dobře fungující internetovou aplikaci využívající k uchovávání dat databázový server.

Databáze pro třídní fond je navržena tak, aby umožňovala registrace jakékoliv školy a třídy z libovolné obce České republiky. Dalším kritériem návrhu databáze bylo dodržení integrity dat.

5.1 ER-diagram



Obrázek 7 – ER diagram

6 Ukázky a vysvětlení některých důležitých částí BP

6.1 Transakce

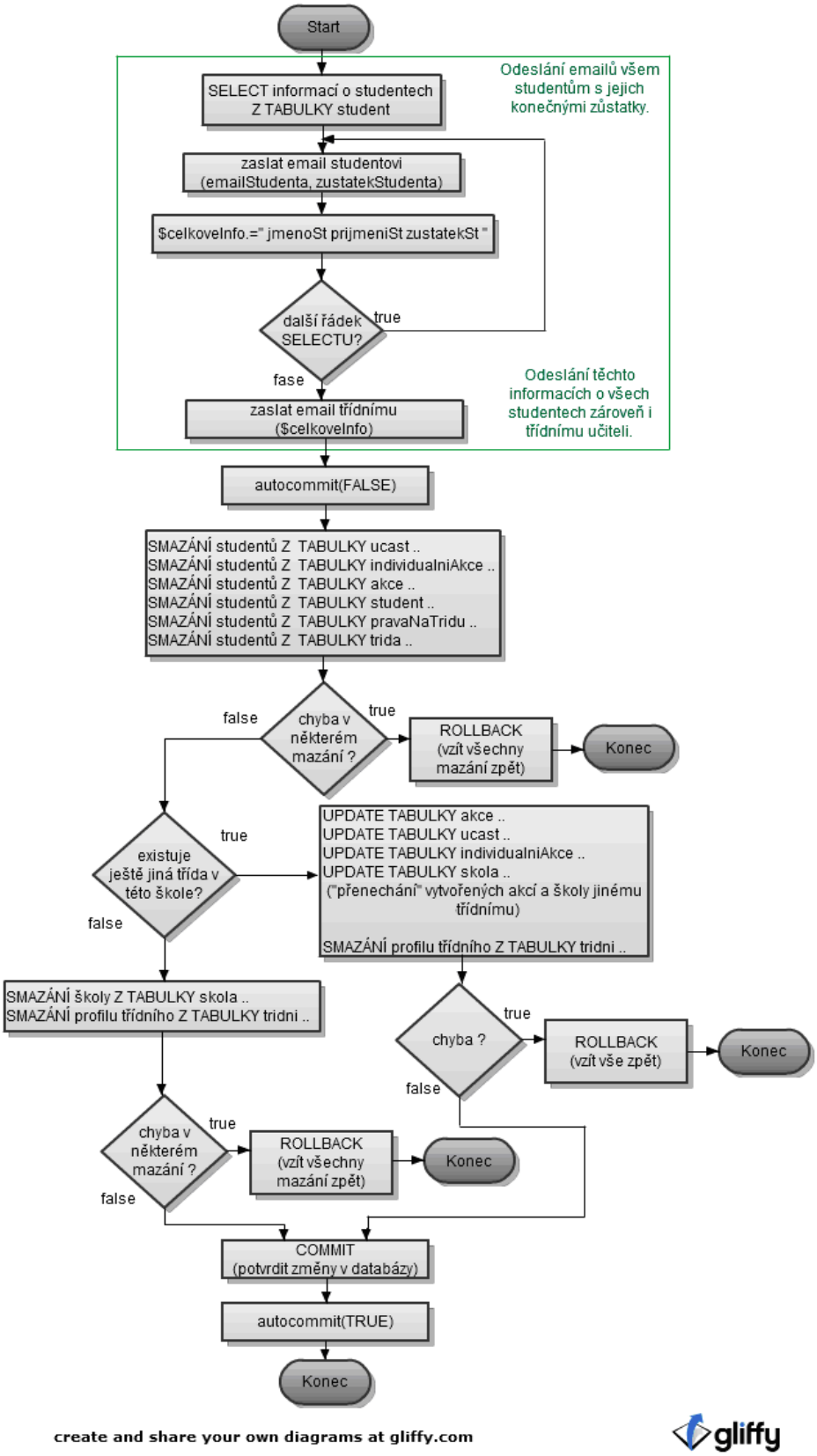
Transakce je určité seskupení, logická část několika SQL příkazů (insert, delete atd.), která musí vždy proběhnout celá. Nikdy nesmí nastat, že by proběhla jen část transakce. Používání transakcí si klade za cíl udržení konzistence dat.

6.1.1 Příklad registrace nové třídy a učitele

V případě bakalářské práce jsem využil transakce například pro registraci nového třídního učitele a jeho třídy, kde je nutné vložit postupně informace do tří tabulek. Pokud by v průběhu scriptu nastala chyba a transakce nebyly použity, v databázi by zůstala uložena jen část dat, což samozřejmě není vítané.

```
// vypnutí automatického potvrzování transakce po každém příkazu
$mysqli->autocommit(FALSE);

// vklad informací do tabulky třídní
$mysqli->query("insert into tridni (nick, heslo, jmeno, prijmeni,  email,
mobil, vytvoreno) values
('".$nickTridni."','".md5(sha1($hesloTridni))."','".$jmeno."',
'".$prijmeni."','".$email."', ".$mobil.", sysdate())") or die ("Chyba
databáze, omlouváme se!");

// zjištění naposledy vloženého id záznamu o učiteli
    $idTridnihoDotaz=$mysqli->insert_id;

// vklad informací do tabulky třída
$mysqli->query("insert into trida (idSkoly, nazevTridy, skolniRok,
minimalniVyseFondu, vytvoreno) values
(".$idSkoly.",'".$nazevTridy."',".$skolniRok.", ".$minimalniVyseFondu.",
sysdate())") or die ("Chyba databáze, omlouváme se!");

// zjištění naposledy vloženého id záznamu o třídě
    $idTridyDotaz=$mysqli->insert_id;

// přiřazení vlastnictví třídy třídnímu učiteli
    $mysqli->query("insert into pravaNaTridu (idTridniho, idTridy,
vlastnik) values (".$idTridnihoDotaz.", ".$idTridyDotaz.",1)") or die
("Chyba databáze, omlouváme se!");
// pokud někde do této části nastala chyba, byla vyvolána funkce die(),
vypsána omluva a ukončeno zpracování skriptů, tím pádem se již neprovede
příkaz COMMIT (níže), ale naopak automaticky ROLLBACK, který vrátí
všechny provedené změny zpět

// pokud chyba až dosud nenastala --> správné zpracování všech informací,
příkazem COMMIT potvrdíme trvalé zapsání dat do databáze
    $mysqli->commit();
// znovu zapneme automatické potvrzování
    $mysqli->autocommit(TRUE);
```

6.1.2 Příklad smazání celé třídy



Obrázek 8 – Vývojový diagram smazání třídy

Pro jednoduchost a přehlednost jsem tento příklad vyjádřil pouze vývojovým diagramem. V první ohraničené části je nastíněn postup odesílání emailů s konečnými zůstatky studentů před jejich postupným mazáním a odeslání emailu s výčtem studentů a jejich zůstatky třídnímu učiteli. V druhé části je vidět, jak je pomocí transakcí zajištěno, aby aplikace při vzniku chyby nesmazala jen část záznamů o třídě.

6.2 Funkce pro zjištění zůstatku studenta

Do funkce nám vstupuje objekt mysqli (připojení k databázi) a primární klíč studenta (`$idStudenta`). Funkce nám pomocí příkazu `SELECT`, který se skládá z dalších šesti vnořených dotazů, zjistí aktuální zůstatek na účtu studenta a jako parametr ho odešle zpět.

```
// definice funkce, její název a vstupní parametry
function zustatekNaUctuStudenta($mysqli, $idStundenta){
// do proměnné $result se ukládá výsledek dotazu
$result=$mysqli->query("
//celkový SELECT
select
      // zjištění sumy všech vkladů studenta, pokud žádné
      nejsou – vrácení hodnoty 0
        (IFNULL((select sum(i.castka) from individualniAkce i,
typIndividualniAkce tia where i.idStudenta=".$idStundenta." and
i.idTypu=tia.idTypu and tia.typ='vklad'),0))
      // odečet sumy všech výplat studenta
      -  (IFNULL((select sum(i.castka) from individualniAkce i,
typIndividualniAkce tia where i.idStudenta=".$idStundenta." and
i.idTypu=tia.idTypu and tia.typ='výplata'),0))
      // odečet sumy všech čerpání studenta
        - (IFNULL((select sum(i.castka) from individualniAkce i,
typIndividualniAkce tia where i.idStudenta=".$idStundenta." and
i.idTypu=tia.idTypu and tia.typ='čerpání'),0))
// odečet součtu  všech hromadných akcí studenta
  (akcí s jednotkovou cenou + akcí s celkovou cenou)
        - (IFNULL((select sum(akce.cena*ucast.pocet) from akce, ucast
where akce.idAkce=ucast.idAkce and ucast.idStudenta=".$idStundenta." and
akce.druhCeny=1),0)
        + (IFNULL((select sum(ucast.pocet*a.cena/(select sum(ucast.pocet)
from ucast where ucast.idAkce=a.idAkce))from ucast join akce a on
a.idAkce=ucast.idAkce where ucast.idStudenta = ".$idStundenta." and
a.druhCeny=2),0))) cena
     ") or die ("Chyba databáze, omlouváme se!");

// načtení výsledku dotazu do proměnné $row
    $row=$result->fetch_assoc();
// dealokace proměnné $result
    $result->close();
// návrat výsledku
    return $row['cena'];
}
```

6.3 Zabezpečení hesla, přihlášení

Ukládání hesla v čitelné formě do databáze je jednou ze základních bezpečnostních chyb. Pokud by se útočník nějakým způsobem dostal přes zabezpečení až k výpisu dat z tabulky, mohl by přečíst přístupové údaje všech uživatelů aplikace. Proto je velice důležité využívat pro zabezpečení hashovací funkce. Vstupem do funkce je jakýkoliv řetězec (v našem případě zadané heslo), jakékoliv délky a výstupem z funkce je takzvaný hash (otisk) konkrétní a neměnné délky.

V bakalářské práci jsem pro zabezpečení hesel v databázi zvolil metodu dvojitého hashování. Nejprve zašifruji heslo pomocí funkce sha1 a vzniklý hash ještě jednou pomocí funkce md5.

6.3.1 Konkrétní příklad

mnou zadané heslo je:	heslo123
md5 (heslo) hash je:	6a284155906c26cbca20c53376bc63ac
sha1 (heslo) hash je:	849b28dcbe2c37b2c60d994e5dbd4b21535d0701
md5(sha1(heslo)):	**c369cb73237bfd5e650690466f121ee2**

V závěru je do databáze ukládán místo řetězce „**heslo123**“, řetězec „**c369cb73237bfd5e650690466f121ee2**“, který i v případě prolomení šifrování metodou md5() stále neprozrazuje nic o konkrétním heslu.

Vzhledem k tomu, že heslo je většinou používáno při **logování uživatelů**, uvedu zde ještě jednu ochranu, kterou jsem využil proti neoprávněnému přístupu a prolomení hesla. Pro případ, že by se útočník pokusil o prolomení pomocí **slovníkového útoku**, bude mu to velice znepříjemněno pomocí **funkce** `sleep(1)`**.** Tato funkce je volána po odeslání přihlašovacích údajů a zapříčiní čekání po dobu jedné sekundy před jejich ověřením. Z toho plynoucí důsledek je, že slovníkový útok, obsahující několik tisíc slov, je prodloužen na neúměrně dlouhou dobu.

6.4 Zabezpečení vstupů z formulářů

Všechny uživatelské vstupy jsou hlídány pomocí regulárních výrazů a funkcí k tomu určených v PHP. K samotnému SQL dotazu, který by mohl být nějakým způsobem nabourán, se nedostane nic jiného, než řetězec ve správném tvaru. V případě, že řetězec neodpovídá zadaným kritériím, nebo není vyplněn povinný údaj, uživatel bude vyzván ke znovu zadání chybných údajů.

U všech SQL dotazů je zakázán výpis chyb. V případě, že by nějaká chyba přeci jen nastala (nejspíše jako důsledek SQL Injection, nebo nedostupné databáze), bude zobrazena pouze informační zpráva („Chyba databáze, omlouváme se!“) a ukončen běh dalších skriptů na stránce. Stejně tak u PHP je v souboru .htaccess zakázán výpis chyb a z bezpečnostních důvodů nastavena direktiva `magic_quotes_gpc ON`.

6.4.1 Číselný vstup

U číselných vstupů je pomocí funkce `is_numeric()`, popřípadě funkce `is_integer()`, vždy ověřováno, zda je vstupem opravdu číslo. Dále je hlídán (např. u mobilního telefonu) počet zadaných číslic, a pokud je tento vstup povinný, tak pomocí funkce `is_empy()` také to, aby byl výraz opravdu zadán.

Příklad ověření mobilního telefonu:

```
if((is_numeric($_POST['mobil'])==false || (strlen($_POST['mobil'])!=9))
&& !empty ($_POST['mobil']))
            {
                $chyba = true;
                $upozorneni .= 'Špatně zadaný mobilní telefon<br>';
            }
```

6.4.2 Řetězec

U vstupů skládajících se z textových řetězců (např. název třídy) je pomocí regulárních výrazů ověřováno, zda se v řetězci vyskytují opravdu jen písmena a číslice, popřípadě několik dalších povolených znaků. Nikdy však nesmí být v řetězci znaky [‘] a [“], které jsou pro tento text zbytečné, a mohly by umožnit napadení databáze. Aplikace dále detekuje vyplnění povinného pole a maximální možnou délka vstupu.

Příklad ověření názvu třídy:

```
if((empty($_POST['nazevTridy'])) || (strlen($_POST['nazevTridy'])>20) ||
!jenPismenaAcisla($_POST['nazevTridy']))
            {
                $chyba = true;
                $upozorneni .= 'Špatně zadaný název třídy<br>';
            }
```

Obsah funkce jenPismenaAcisla – regulární výraz:

```
function jenPismenaAcisla($slovo){
    if (preg_match('/^[a-žA-Ž0-9 ]*$/', $slovo)){
        return true;
    }else {return false;}
}
```

6.4.3 Ověření data

Do funkce pro ověření data vstupuje řetězec ve tvaru [dd.mm.yyyy] – v České republice asi nejvíce uživatelsky příjemný způsob zadávání data. Funkce nejprve rozdělí řetězec podle znaku [.] a zjistí, zda se jedná u dne, měsíce a roku opravdu jen o čísla. Dále zkontroluje pomocí PHP funkce `checkdate()` správnost a pravdivost zadaného data (např. 29.2. nemůže nastat, protože únor má jen 28 dní) a nakonec rozmezí zadaného roku (pro tuto aplikaci jsem nastavil rok 2000 až 2100). Návratová hodnota je datum ve formátu [yyyy-mm-dd] – nejvhodnější formát pro zápis do MySQL databáze. V případě nalezení chyby v zadaném řetězci je návratovou hodnotou „`false`“.

```
function spravneDatum($datum){
$pole = explode(".", $datum);
$den=$pole[0];
$mesic=$pole[1];
$rok=$pole[2];

    if (is_numeric($den) && is_numeric($mesic) && is_numeric($rok)){
        if (checkdate($mesic, $den, $rok)){
            if ($rok>2000 && $rok<2100){
                $vratit=$rok.'-'.$mesic.'-'.$den;
                return $vratit;
            }
        }
    }
        return false;
}
```

6.5 Generování PDF

Před generováním PDF souborů jsem se nejprve musel rozhodnout, co k tomu využít. Po nějaké době hledání, čtení článků a recenzí, jsem narazil na dvě asi nejběžnější (alespoň dle zdrojů na Internetu) možnosti, jak PDF soubory z PHP generovat. Samozřejmě možností existuje daleko více, ale soustředil jsem se na jednoduchý způsob a software zdarma.

6.5.1 FPDF třída

Tato třída vypadá velice slibně. Dle manuálu umí spoustu funkcí [12]. Jedinou její nevýhodou je, že primárně podporuje jen kódování ISO-8859-1, což mě v tomto případě odradilo od jejího využití. V diskusních článcích jsem se dočetl, jak správně používat FPDF i s kódováním UTF-8, ale nakonec jsem se rozhodl hledat dále.

6.5.2 mPDF třída

Tato třída má velkou podporu v uživatelské komunitě a velice pěkně zpracovaný manuál [13]. Nedokážu posoudit, zda toho umí více, než výše zmiňované FPDF, ale rozhodující pro její využití byla přímá podpora UTF-8 a od prvního okamžiku přehledný a jednoduchý způsobe generování PDF souborů. Ukázka níže.

6.5.3 Praktická ukázka

V praktické ukázce bude vygenerováno jednoduché PDF s nápisem: „Hallo World!“.

```
<?php
// výběr cesty ke stažené složce s mPDF
include("MPDF43/mpdf.php");

// nastavení hlavičky na typ PDF
header("Content-Type: application/pdf");

// nový PDF soubor, možné nastavení kódování, velikosti stránky a
okrajů..
$mpdf=new mPDF();

// výpis HTML kódu (lze nejprve zapisovat do proměnné)
$mpdf -> WriteHTML ('<p> Hallo World! </ p>');

// tisk do PDF souboru
$mpdf -> Output ();
?>
```

Třída mPDF toho samozřejmě dokáže daleko více. Dovolil bych si tvrdit dokonce vše běžně potřebné. Můžete zde jednoduše nastavovat autora práce, přesné rozměry stránky, kódování, typ, barvu a velikost písma, záhlaví a zápatí PDF, vodotisk, čísla stránek a mnoho dalšího. Třída mPDF také podporuje CSS. Pokud generujete stránku v PHP, povětšinou Vám postačí místo „echování“ vypsat vše i s celým HTML kódem do proměnné a tu pomocí výše předvedené funkce `$mpdf -> WriteHTML($proměnná)` vytisknout do PDF souboru. Při nutnosti generování PDF tuto třídu vřele doporučuji.

6.6 Generování XLS

Generování XLS se zprvu zdálo být velkým oříškem. Na Internetu existuje několik tříd zaměřených na XLS souborů v PHP, stejně jako na PDF soubory. Tyto třídy nabízí naprosto nepřeberné množství funkcí pro práci s XLS, ale jsou povětšinou velice komplikované.

Nakonec jsem se pozastavil u informace o tom, že Microsoft Excel dokáže zpracovat HTML stránku s tabulkou [14]. Možnosti jsou sice omezené, například při výběru barvy buněk a práci s nimi, ale pro „jednoduché“ generování je tato vlastnost plně dostačující.

6.6.1 Praktická ukázka

Pro jednoduchost si vygenerujeme XLS tabulku se jmény a příjmeními studentů. V tabulce budou vloženi dva studenti.

```
<?php
// nastavení hlavičkového souboru na typ MS Excel
header("Content-Type: application/vnd.ms-excel");
header("Cache-control: private");

// nastavení názvu souboru
header("Content-Disposition: attachment; filename=soubor.xls");

// uložení HTML hlavičky a tabulky do proměnné $html
$html = '<html><head><meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"></head><body>';
$html .= '<table>
<tr><th>Jméno</th><th>Příjmení</th></tr>
<tr><td>Roman</td><td>Svoboda</td></tr>
<tr><td>Pavel</td><td>Jandejsek</td></tr>
</table>';
$html .= '</body></html>';

// výpis proměnné
echo $html;
?>
```

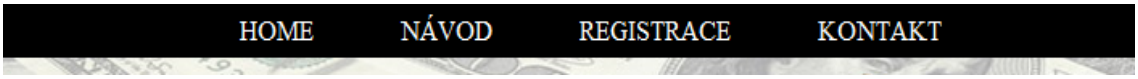
Po provedení tohoto skriptu budeme dotázáni, zda XLS soubor stáhnout či otevřít. V případě potřeby lze využívat základní HTML tagy (např. pro podbarvení buňky: `<th bgolor="red">Obsah buňky</th>`).

7 Podrobný popis funkcionality

V tomto oddíle se budu věnovat popisu funkcionality z pohledu čtenáře bakalářské práce jako uživatele příslušné úrovně.

7.1 Neregistrovaný uživatel

Jako neregistrovaný uživatele máte na výběr z tohoto menu:



Obrázek 9 – Neregistrovaný uživatel – Menu

HOME

Tato položka Vás odkáže na úvodní stranu.

NÁVOD

Odkaz NÁVOD Vás přesměruje na stránku, kde si lze stáhnout uživatelský manuál k aplikaci.

REGISTRACE

V této sekci webové aplikace můžete zaregistrovat svou školu, nebo třídu. Registrace proběhne v několika krocích:

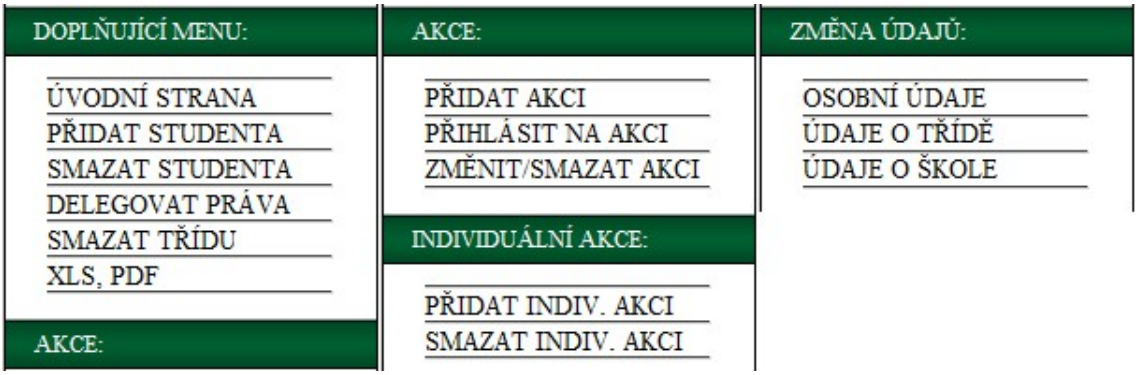
- nejprve vyberte okres, ve kterém se Vaše škola nachází,
- vyberte město v tomto okresu (databáze je aktualizována dle českého statistického úřadu a jeho výsledků v roce 2009),
- podívejte se, zda Vaše škola již není registrována, pokud ano, můžete si v ní založit třídu, pokud ne nebo neznáte zakládající osobu, založte si svou vlastní školu,
- vyplňte všechny požadované údaje a stiskněte odeslat, po odeslání údajů Vám bude doručen email s nickem a heslem – pokud nemůžete email najít, podívejte se do spam složky.

KONTAKT

V této sekci najdete kontakt, na který se můžete obrátit v případě jakýchkoli dotazů a připomínek.

7.2 Třídní učitel

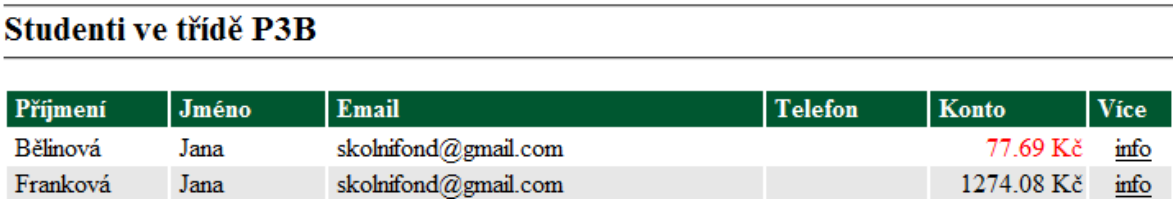
Po registraci Vám budou doručeny na email přihlašovací údaje. Pokud je ve své poště nemůžete nalézt, zkontrolujte prosím spam složku. Je možné, že Váš email nesprávně zařadil naši zprávu. Po obdržení kontaktních údajů se přihlaste.



Obrázek 10 – Třídní učitel – Menu

ÚVODNÍ STRANA

Tato položka menu odkazuje na úvodní stránku, kde jsou zobrazeny všechny informace o studentech ve Vaší třídě, včetně zůstatku na jejich kontech. Pokud je zůstatek studenta menší, než je Vámi nastavená hranice, zůstatek je vypsán červeně. Pokud smažete studenta z Vaší třídy, jeho jméno a zůstatek bude pro lepší kontrolu financí stále zobrazován, ale před jménem se objeví červený znak „!“. Pro více informací týkajících se konkrétního studenta klikněte na odkaz info.

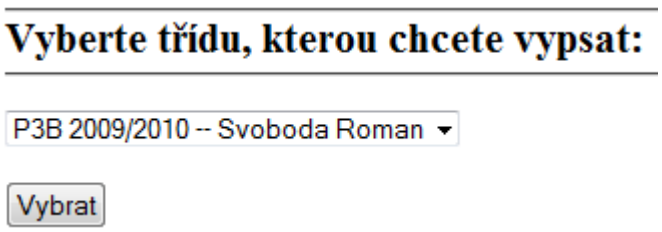


Studenti ve třídě P3B

Příjmení	Jméno	Email	Telefon	Konto	Více
Bělinová	Jana	skolnifond@gmail.com		77.69 Kč	info
Franková	Jana	skolnifond@gmail.com		1274.08 Kč	info

Obrázek 11 – Třídní učitel – Úvodní stránka

Obrázek níže znázorňuje možnost výpisu informací o Vám delegovaných třídách.



Vyberte třídu, kterou chcete vypsat:

P3B 2009/2010 -- Svoboda Roman

Vybrat

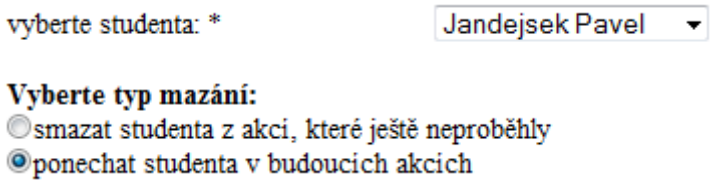
Obrázek 12 – Třídní učitel – Výpis informací o delegovaných třídách

PŘIDAT STUDENTA

Jednoduše vyplníte jméno, příjmení studenta a kontaktní email, na který bude po zadání odeslána informace o registraci s přihlašovacími údaji. Na tento email budou zasílány i „varovné zprávy“ o nízkém kreditu ve školním fondu, proto doporučujeme zadávat email na zákonného zástupce. Políčko pro mobilní telefon není nutné vyplňovat. Po kliknutí na tlačítko odeslat bude automaticky přidán student do Vaší třídy.

SMAZAT STUDENTA

Zde lze ze své třídy vymazat studenta. Používejte tuto volbu jen v ojedinělých případech, ke smazání celé třídy slouží jiný odkaz.



Obrázek 13 – Třídní učitel – Smazání studenta

Jak postupovat:

- vyberte studenta,
- vyberte typ mazání,
- odeslat informace pomocí tlačítka smazat.

Při výběru typu mazání máte tyto možnosti:

- smazat studenta z akcí, které ještě neproběhly – student bude před odesláním emailu s konečným zůstatkem smazán z akcí, které mají datum konání vyšší než je aktuální datum,
- ponechat studenta v budoucích akcích – studentovi bude odeslán email s nynějším zůstatkem.

Vždy po smazání Vám bude vypsán konečný zůstatek ve fondu studenta. Vymazanému studentovi můžete stále zaznamenávat individuální akce.

DELEGOVAT PRÁVA

Na této stránce je zobrazeno, komu jste již delegovali práva na svou třídu. Dále zde můžete již zmiňovaná práva nově předávat či odebírat.

Delegování práv = umožnit jinému učiteli z Vaší školy přidávat hromadné a individuální akce a přiřazovat k nim Vaše studenty (např. v době nepřítomnosti).

SMAZAT TŘÍDU

Pro smazání Vaší třídy existují dvě varianty:

1. Smazat třídu i s profilem:
   - bude zrušena jak třída a studenti v ní, tak i Váš profil,
   - pokud existuje ve škole ještě jiný třídní učitel, musíte vybrat, komu bude přiřazeno „založení“ školy,
   - v případě, že jste jediný učitel ve Vaší škole, bude z databáze smazána i škola.
2. Smazat třídu a založit novou:
   - bude smazána Vaše nynější třída se studenty a založena nová prázdná třída,
   - informace o nové třídě můžete kdykoliv změnit,
   - Váš profil zůstane v původní podobě.

Po smazání Vám bude zaslán email se jmény Vašich studentů a jejich zůstatky. Stejně tak bude odeslán email s konečným individuálním stavem fondu i jednotlivým studentům.

XLS

Tento odkaz Vám umožní vygenerovat XLS soubor s maticovým výpisem všech studentů, třídních akcí a informací o nich.

Po stisknutí budete vyzváni k výběru ze dvou možností. Váš XLS soubor můžete přímo otevřít, nebo si ho uložit pro pozdější zhlédnutí. XLS soubory se generují s názvem složeným z dnešního data a názvu třídy.

Microsoft Excel - P3B-28.03.2010.xls

Soubor Úpravy Zobrazit Vložit Formát Nástroje Data Okno Nápověda

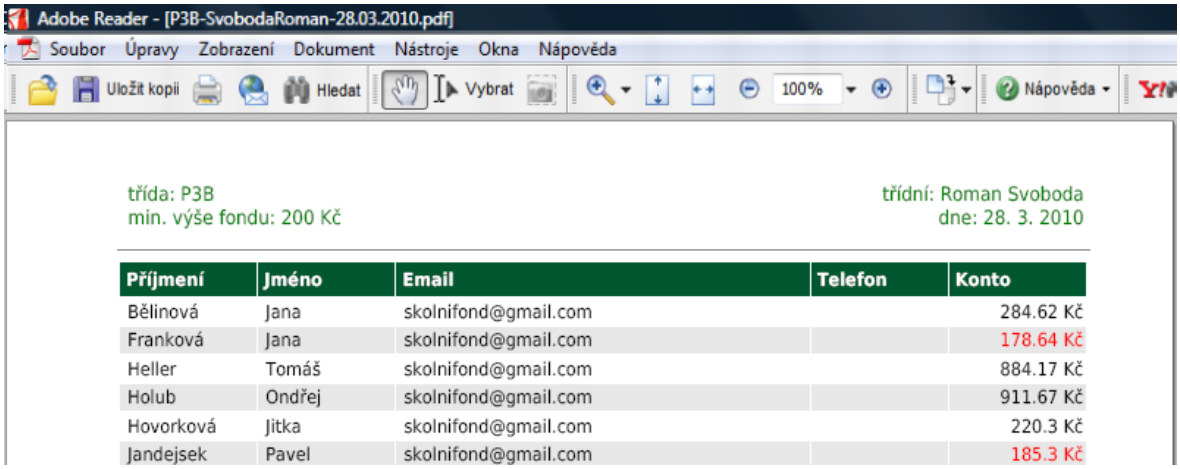
J20

	A	B	C	D	E	F	G	H
1		13.11.2010	19.12.2010	28.3.2011				
2		Divadlo (140 Kč/J)	Chata hory (5000 Kč/C)	Hory (80 Kč/J)	∑ indiv. příjmy	∑ indiv. čerpání	∑ indiv. výplaty	zůstatek
3	Bělinová Jana	140 Kč	454.55 Kč	80 Kč	1 500 Kč	0 Kč	100 Kč	284.62 Kč
4	Franková Jana	0 Kč	454.55 Kč	160 Kč	1 500 Kč	400 Kč	0 Kč	178.64 Kč
5	Heller Tomáš	0 Kč	0 Kč	160 Kč	1 500 Kč	150 Kč	0 Kč	884.17 Kč
6	Holub Ondřej	140 Kč	0 Kč	160 Kč	1 500 Kč	0 Kč	0 Kč	911.67 Kč
7	Hovorková Jitka	140 Kč	454.55 Kč	160 Kč	1 500 Kč	0 Kč	0 Kč	220.3 Kč
8	Jandejsek Pavel	140 Kč	454.55 Kč	160 Kč	1 500 Kč	0 Kč	0 Kč	185.3 Kč
9	Křečková Alena	0 Kč	454.55 Kč	80 Kč	1 500 Kč	0 Kč	200 Kč	222.8 Kč
10	Mazůrková Ilona	140 Kč	454.55 Kč	0 Kč	1 500 Kč	0 Kč	0 Kč	390.3 Kč
11	Michálková Simona	140 Kč	0 Kč	0 Kč	1 500 Kč	0 Kč	0 Kč	1025.68 Kč
12	Pátková Darja	0 Kč	0 Kč	80 Kč	1 500 Kč	0 Kč	0 Kč	949.85 Kč
13	Peterová Aneta	140 Kč	0 Kč	240 Kč	1 500 Kč	300 Kč	0 Kč	468.18 Kč
14	Rajčanyová Renata	0 Kč	454.55 Kč	160 Kč	1 500 Kč	0 Kč	0 Kč	444.62 Kč
15	Šedina Martin	140 Kč	454.55 Kč	0 Kč	1 500 Kč	0 Kč	80 Kč	282.8 Kč
16	Štětinová Markéta	140 Kč	454.55 Kč	80 Kč	1 500 Kč	0 Kč	0 Kč	582.12 Kč
17	Svoboda Karel	0 Kč	0 Kč	0 Kč	1 500 Kč	0 Kč	0 Kč	1 500 Kč
18	Vozáb Václav	0 Kč	454.55 Kč	160 Kč	1 900 Kč	0 Kč	0 Kč	662.8 Kč
19	Zicha Daniel	140 Kč	454.55 Kč	80 Kč	1 500 Kč	0 Kč	0 Kč	411.14 Kč
20	Celkem:	1 400 Kč	5 000 Kč	1 760 Kč	25 900 Kč	850 Kč	380 Kč	1 500 Kč

Obrázek 14 – Třídní učitel – XLS

PDF

Tímto odkazem můžete vygenerovat PDF soubor, ve kterém bude shrnutí Vašeho fondu. Na úvodní straně jsou informace o studentech a jejich zůstatcích a na dalších stranách přehled všech akcí jednotlivých studentů. Toto PDF je v tištěné podobě velice příjemným pomocníkem například při třídních schůzkách.

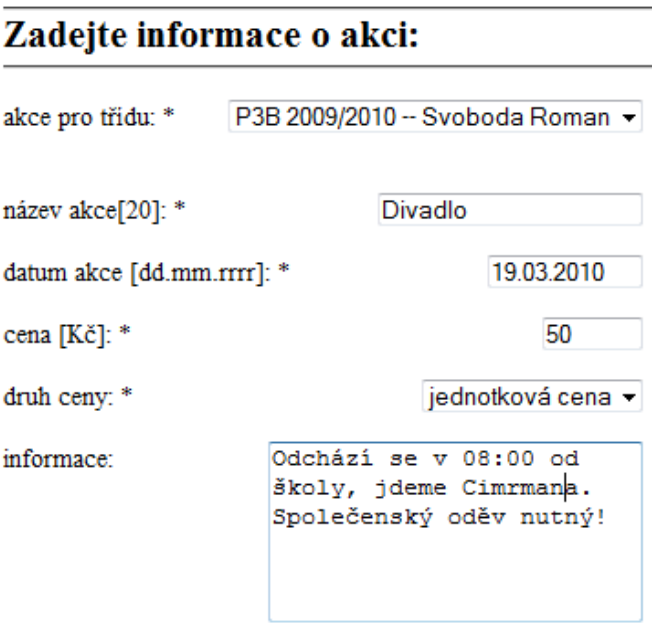


Příjmení	Jméno	Email	Telefon	Konto
Bělinová	Jana	skolnifond@gmail.com		284.62 Kč
Franková	Jana	skolnifond@gmail.com		178.64 Kč
Heller	Tomáš	skolnifond@gmail.com		884.17 Kč
Holub	Ondřej	skolnifond@gmail.com		911.67 Kč
Hovorková	Jitka	skolnifond@gmail.com		220.3 Kč
Jandejsek	Pavel	skolnifond@gmail.com		185.3 Kč

Obrázek 15 – Třídní učitel – PDF

PŘIDAT AKCI

Zde lze přidávat hromadné akce. Jak si v dalším kroku vysvětlíme, můžete na ně poté přihlašovat studenty. Nejprve vyberte třídu, která se akce zúčastní, vyplňte název, datum (vždy je Vám nabídnuto primárně dnešní datum), cenu, druh ceny a nepovinně doplňující informace o akci.



Obrázek 16 – Třídní učitel – Přidání akce

Jsou dány dva druhy ceny:

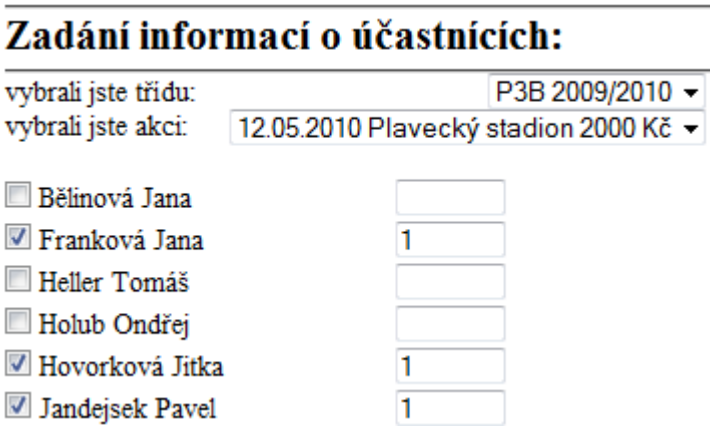
- **jednotková** – např. návštěva kina, divadla atd.,
- **celková** – např. cena za autobus, celkový pronájem plaveckého stadionu a jiné.

PŘIHLÁSIT NA AKCI

Zde můžete přiřazovat studenty k akcím, které jsou registrovány pro jejich třídu.

Jakým způsobem přihlašovat:

- vyberte Vaší, nebo z Vámi delegovaných tříd,
- vyberte akci této třídy,
- zaškrtněte studenty, kteří se akce budou účastnit nebo se jí účastnili (k tomuto kroku máte přístup kdykoliv, pokud někdo nepřišel do školy, nezoufejte, jednoduše můžete účastníky měnit).



Obrázek 17 – Třídní učitel – Informace o účastnících akce

Sloupec s čísly značí poměrný počet odebraných jednotek. Jedete-li např. na hory, kde je cena akce zadána za jednu noc (např. 150 Kč – **jednotková cena**), vyplňte zde počet nocí, které bude student na horách spát a aplikace za Vás sama dopočítá cenu pro studenta. U akcí s **celkovou cenou** se závěrečná suma za studenta rozpočítá procentuálně.

Po odeslání údajů aplikace zkontroluje **aktuální zůstatky** na kontech studentů. V případě, že jsou menší než Vámi nastavená mez, vypíše jejich jména a odešle **varovné emaily**.

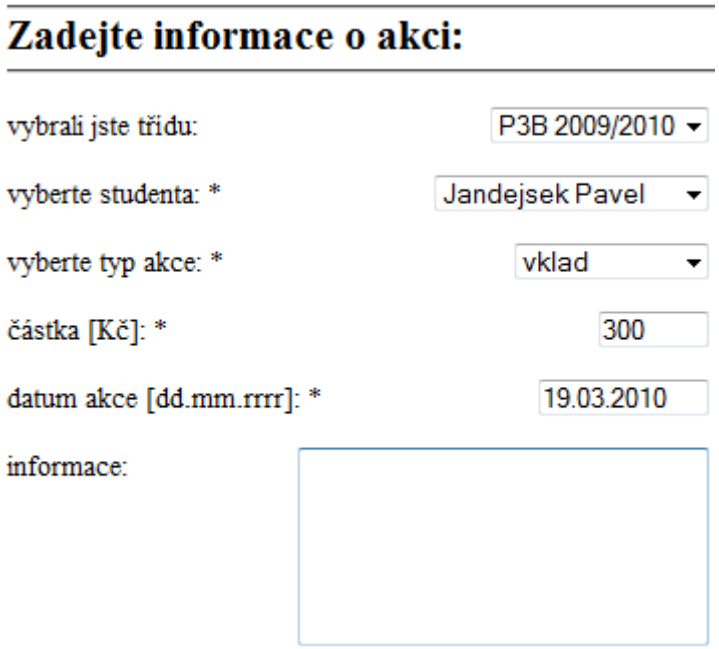
ZMĚNIT / SMAZAT AKCI

Zde lze měnit informace o akcích, nebo akce přímo mazat. Provedené změny se ihned projeví na kontech studentů. Všechny částky za tuto akci budou znovu přepočítány a připraveny k nahlédnutí.

Aplikace zkontroluje zůstatky na kontech zúčastnivších se studentů a v případě malého kreditu vypíše jejich jména a odešle varovné emaily.

PŘIDAT INDIV. AKCI

Zde lze přidávat záznamy o individuálních akcích. Přednastavené akce jsou: **vklad, výplata a čerpání**. Tuto část aplikace využijete například ve chvíli, kdy Vám student přinese hotovost, nebo naopak vyplácíte finanční obnos konkrétnímu studentovi.



Obrázek 18 – Třídní učitel – Individuální akce

Nejprve vyberte třídu, ze které je onen student a poté vyplňte všechna potřebná data. Po přidání akce bude zkontrolován stav konta studenta. **V případě malého zůstatku Vám bude vypsáno varovné hlášení a studentovi zaslán email**.

SMAZAT INDIV. AKCI

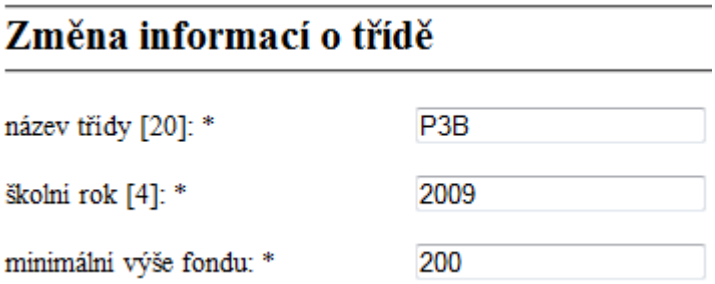
Pokud jste se spletli při zadávání individuální akce nebo z jakéhokoliv důvodu chcete akci smazat, můžete to udělat zde.

OSOBNÍ ÚDAJE

V této sekci můžete měnit své osobní údaje. Ihned po změně Vám bude zaslána informační zpráva, pokud ji neobdržíte, znovu zkontrolujte zadaný email. Mobilní telefon není povinný údaj. V případě změny hesla, vyplňte všechny pole pro něj určené. Pokud heslo měnit nechcete, nevyplňujte naopak ani jedno.

ÚDAJE O TŘÍDĚ

Zde lze měnit informace týkající se Vaší třídy, po odeslání budou ihned všem zobrazovány aktuální údaje.



Obrázek 19 – Třídní učitel – Změna údajů třídy

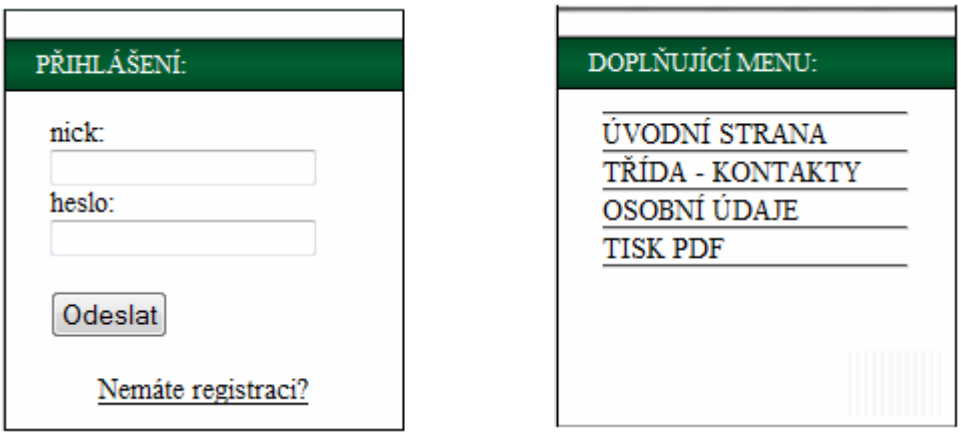
Položka „minimální výše fondu“ určuje, při jakém stavu financí na studentských účtech budou odesílány informační emaily s žádostí o navýšení. **Při změně částky minimální výše fondu, aplikace odešle všem studentům ve třídě informaci o této události.** V případě, že student má menší zůstatek, než je požadován, bude v emailu upozorněn.

ÚDAJE O ŠKOLE

Možnost změny údajů o škole se zobrazuje jen „zakladateli“ školy.

7.3 Student (rodič)

Po registraci studenta třídním učitelem obdržíte emailovou zprávou přihlašovací údaje. Pokud je ve své poště nemůžete nalézt, zkontrolujte prosím i spam složku. Je možné, že Váš email nesprávně zařadil naši zprávu. Po obdržení kontaktních údajů se přihlaste.



Obrázek 20 – Student – a) Přihlašovací formulář, b) Menu

Jako rodič studenta máte díky aplikaci plně pod kontrolou výdaje svého syna (dcery). Můžete shlédnout všechny jeho vklady na účet a informace o akcích, kterých se v rámci školy účastní. Dále v případě potřeby máte přístup ke kontaktu na třídního učitele a rodiče ostatních studentů. V neposlední řadě je Vám umožněna úschova všech důležitých informací ve formátu PDF.

ÚVODNÍ STRANA

Tato položka menu odkazuje na Vaši úvodní stránku, kde jsou poskytovány informace o všech akcích studenta a výši jeho konta. Pokud chcete, můžete se na ni kdykoliv vrátit.

Celkový zůstatek na kontu

Na Vašem kontu je: **475.22 Kč**

Vaše čerpání a vklady

Datum	Název	Doplňující informace	Cena	Zůstatek
02.06.2010	Zříceni Choustník	Pojedeme se podívat do jižních Čech :-)	-142.86 Kč	475.22 Kč
28.05.2010	Plavecký bazén	Máme rezervován na 2h plavecký bazén KH.	-125 Kč	618.08 Kč

Obrázek 21 – Student – Úvodní stránka

TŘÍDA – KONTAKTY

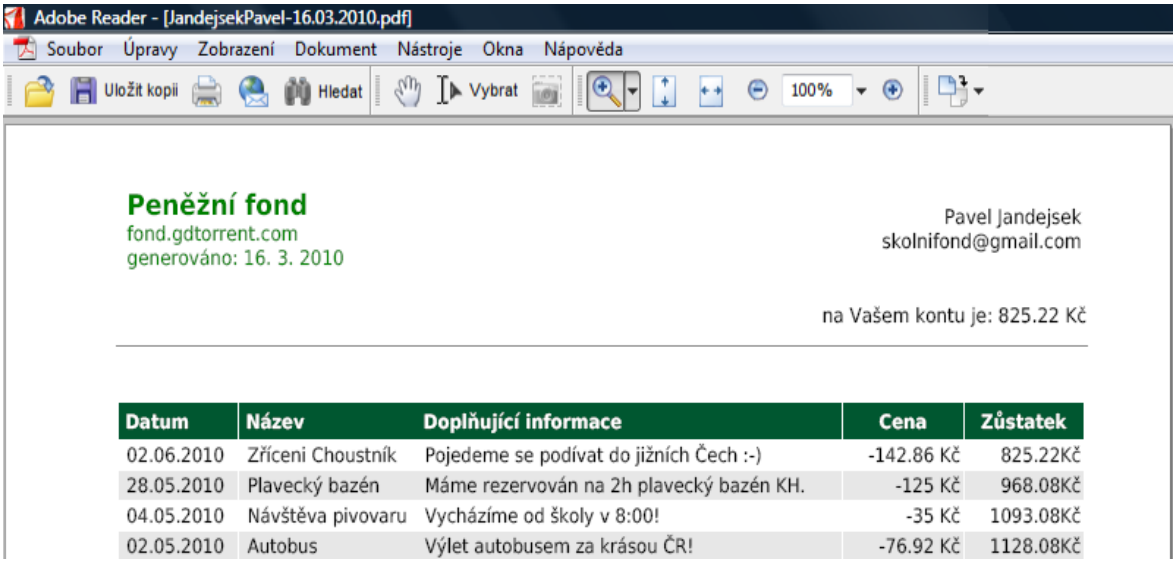
Zde naleznete podrobné informace o škole, třídě, třídním učiteli a všech spolužácích.

OSOBNÍ ÚDAJE

V této sekci můžete měnit své osobní údaje. Ihned po změně Vám bude zaslána informační zpráva, pokud ji neobdržíte, znovu zkontrolujte zadaný email. Mobilní telefon není povinný údaj. V případě změny hesla, vyplňte všechny pole pro něj určené. Pokud ho měnit nechcete, nevyplňujte naopak ani jedno.

TISK PDF

Po stisknutí toho odkazu, budete vyzváni k výběru ze dvou možností, Váš PDF soubor můžete přímo otevřít, nebo si ho uložit. PDF soubory se generují s názvem, který se skládá z dnešního data a jména studenta.



Obrázek 22 – Student – PDF

8 Závěr

Cílem bakalářské práce bylo vytvořit WWW aplikaci pro správu třídního fondu. Při tvorbě byly využity technologie: HTML, CSS, PHP a databázový server MySQL. Konečná webová aplikace je plně validní a odpovídá tedy pravidlům pro tvorbu webu. Aplikace byla testována a je plně funkční v internetových prohlížečích: Mozilla Firefox 3.5.9, Opera 10.10., Google Chrome 4.1. a Internet Explorer 7.

Třídní fond splňuje všechny požadované funkcionality. Aplikace je tvořena tak, aby třídní učitel i rodič studenta mohli bez jakýchkoliv odborných znalostí plně využívat její funkčnost. Dále byl splněn požadavek na automatické generování varovných zpráv při poklesu financí na kontu studenta pod nastavenou mez, čímž aplikace usnadňuje třídnímu učiteli správu fondu. Pro úschovu informací, přípravu třídních schůzek nebo jako podklad pro peněžní deník je uživateli umožněno ukládat souhrnné informace do XLS a PDF souborů.

Při tvorbě layoutu www stránek byl kladen důraz na co nejjednodušší, nejpřehlednější a zároveň uživatelsky neodpudivý design. Rozvržení bylo uzpůsobeno přehlednosti výpisů a intuitivnosti uživatelského ovládání.

Možný budoucí rozvoj aplikace vidím v zakomponování každoroční uzávěrky třídního fondu. Vždy na konci školního roku by byla provedena sumarizace všech dat do souborů a ty odeslány emaily konkrétním uživatelům, popřípadě uloženy na server a nabídnuty, v případě zájmu, k pozdějšímu stažení. Zároveň s tímto krokem by bylo nutné pozměnit databázovou strukturu a vhodně přidat uživatelům parametr „převod“, který by udával konečný zůstatek na účtu studenta po minulém školním roce. S tímto parametrem by bylo nutné počítat při všech výpisech a dotazech na zůstatky studentů. Změna by vedla k určitému zpřehlednění aplikace a v konečném důsledku i k úspoře místa na databázovém serveru a snížení datového toku mezi aplikací a tímto serverem.

Při tvorbě bakalářské práce jsem se naučil mnoho nových programátorských dovedností a rozšířil si teoretické i praktické znalosti o zabezpečení www aplikací. Cíl bakalářské práce považuji za splněný.

Literatura

[1] *Web Hacking – PHP Injection* [Online]. [200-?]. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://stoyan.cz/hacking-php-injection/>.

[2] *SQL Injection* [Online]. 24. 1. 2005. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://security-portal.cz/clanky/sql-injection>.

[3] *SQL Injection (Full Paper)* [Online]. 5. 11. 2009. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://security-portal.cz/clanky/sql-injection-full-paper>.

[4] *Jak ukládat hesla do databáze* [Online]. 6. 1. 2010. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://miho.blog.zive.cz/2010/01/jak-ukladat-hesla-do-databaze/>.

[5] **MALÝ, J.; KACÁLEK, J..** *Zabezpečení webových aplikací II. – databáze* [Online]. 15. 8. 2007. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://access.feld.cvut.cz/rservice.php?akce=tisk&cisloclanku=2007080002>.

[6] *Zabezpečení hesla z pohledu programátora* [Online]. 21. 2. 2009. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://www.it-joker.cz/Pocitace-weby/93-Zabezpeceni-hesla-z-pohledu-programatorap.2.html>.

[7] **BITTO, Ondřej**. *Lámání hesel v praxi (1.)* [Online]. 12. 7. 2005. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://www.lupa.cz/clanky/lamani-hesel-v-praxi-1/>.

[8] **ŽÁK, David**. *Databázové systémy II – Zálohování dat* [Přednáška]. Pardubice: Univerzita Pardubice. 2009. [Citace: 17. 4. 2010].

[9] *Soubor .htaccess* [Online]. [200-?]. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://www.jakpsatweb.cz/server/htaccess.html>.

[10] **ČEGAN, Lukáš**. *Návrh a tvorba WWW – Úvod do vývoje webových aplikací* [Přednáška]. Pardubice: Univerzita Pardubice. 2009. [Citace: 17. 4. 2010].

[11] **KOFLER, Michael; ÖGGL, Bernd**. *PHP 5 a MySQL 5: Průvodce webového programátora*. Vyd. 1. [s.l.]: Computer press, 2007. 607 s. ISBN 978-80-251-1813-9.

[12] *What is FPDF?* [Online]. [200-?]. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://www.fpdf.org/>.

[13] *mPDF* [Online]. 21. 4. 2010. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://mpdf.bpm1.com/>.

[14] **ŠEDO, Jan**. *Soubory MS Excel a MS Word v PHP, ASP či Notepadu* [Online]. 6. 10. 2002. [Citace: 17. 4. 2010]. Dostupný z WWW: <http://interval.cz/clanky/soubory-ms-excel-a-ms-word-v-php-asp-ci-notepadu/>.

Příloha A – Podrobný popis tabulek databáze

okres

V této tabulce jsou vloženy názvy všech okresů České republiky. Využívá ji tabulka MESTO.

název sloupce	Typ	constraint	key
idOkresu	Int	autoincrement	PK
nazevOkresu	Varchar(40)	NN, UNIQ	

mesto

V této tabulce jsou vloženy názvy všech měst České republiky. Tabulka využívá tabulku OKRES.

název sloupce	Typ	constraint	key
idMesta	Int	autoincrement	PK
idOkresu	Int		FK
nazevMesta	Varchar(40)	NN	

skola

Zde jsou uloženy všechny informace týkající se jednotlivých škol. Tabulka využívá tabulky MESTO, DRUH a TRIDNI a je využívána tabulkou TRIDA.

název sloupce	Typ	constraint	key
idSkoly	Int	autoincrement	PK
idMesta	Int		FK
idDruhu	Int		FK
idVytvorilTridni	Int		FK
nazevSkoly	Varchar(20)	NN	
upresnujiInformace	Text		
vytvoreno	Datetime		
posledniUprava	Timestamp		

druh

Tato tabulka obsahuje informace o druhu školy (např. školka, základní škola, střední škola atd.). Využívá ji tabulka SKOLA.

název sloupce	Typ	constraint	key
idDruhu	Int	autoincrement	PK
idMesta	Varchar(20)	UNIQ	

tridni

Tabulka TRIDNI obsahuje informace o třídních učitelích. Data z ní využívají tabulky AKCE, UCAST, INDIVIDUALNIAKCE, PRAVANATRIDU a SKOLA.

název sloupce	Typ	constraint	key
idTridniho	Int	autoincrement	PK
nick	Varchar(7)	NN, UNIQ	
heslo	Char(32)	NN	
jmeno	Varchar(20)	NN	
prijmeni	Varchar(25)	NN	
email	Varchar(70)	NN	
mobil	Int(9)		
vytvoreno	Datetime		
posledniUprava	Timestamp		

trida

Do této tabulky se zaznamenávají všechny informace o třídách. Využívá tabulku SKOLA a je využívána tabulkou STUDENT a PRAVANATRIDU.

název sloupce	typ	constraint	key
idTridy	Int	autoincrement	PK
idSkoly	Int		FK
nazevTridy	Varchar(20)	NN	
skolniRok	Year(4)	NN	
minimalniVyseFondu	Int	NN	
vytvoreno	Datetime		
posledniUprava	Timestamp		

pravaNaTridu

Zde se zaznamenávají údaje o vlastnictví tříd, popřípadě o delegovaných právech. Hodnota „1“ ve sloupci [vlastnik] znamená, že třídní učitel je vlastníkem třídy a hodnota „2“ znamená, že má pouze delegovánu pravomoc na manipulaci s ní. Využívá dat z tabulek TRIDNI a TRIDA.

název sloupce	typ	constraint	key
idTridy	Int		PFK
idTridniho	Int		PFK
vlastnik	Varchar(20)		
vytvoreno	Year(4)		

student

Zde jsou uloženy osobní údaje všech studentů. Sloupec [smazano] obsahuje informaci o tom, zda byl student individuálně odstraněn ze třídy (vyplněno datum) či ne (hodnota „null“) Tato tabulka využívá tabulku TRIDA a je využívána tabulkami UCAST a INDIVIDUALNIAKCE.

název sloupce	Typ	constraint	Key
idStudenta	Int	autoincrement	PK
idTridy	Int		FK
nick	Varchar(7)	NN, UNIQ	
heslo	Char(32)	NN	
jmeno	Varchar(20)	NN	
prijmeni	Varchar(25)	NN	
email	Varchar(70)	NN	
mobil	Int(9)		
vytvoreno	Datetime		
smazano	Date		
posledniUprava	Timestamp		

akce

Tato tabulka obsahuje informace o hromadných akcích. Hodnota „1“ v druhu ceny značí jednotkovou cenu, hodnota „2“ značí cenu za celou akci. Využívá tabulky TRIDNI a TRIDA a je využívána tabulkou UCAST.

název sloupce	Typ	constraint	Key
idAkce	Int	autoincrement	PK
idTridy	Int		FK
idZmenilTridni	Int		FK
nazev	Varchar(20)	NN	
doplnujiciInformace	Text		
datumKonani	Date	NN	
druhCeny	Int	NN	
cena	Float	NN	
vytvoreno	Datetime		
posledniUprava	Timestamp		

ucast

Tabulka UCAST zachycuje účast studentů na hromadných akcích a počet jednotek, které tímto odčerpávají. Využívá tabulky AKCE, STUDENT a TRIDNI.

název sloupce	Typ	constraint	Key
idStudenta	Int	autoincrement	PFK
idAkce	Int		PFK
idZmenilTridni	Int		PFK
pocet	Int	NN	
posledniUprava	Timestamp		

individualniAkce

Tato tabulka obsahuje informace o individuálních akcích jednotlivých studentů. Čerpá z tabulek STUDENT, TYPINDIVIDUALNIAKCE a TRIDNI.

název sloupce	Typ	Constraint	Key
idIndividualniAkce	Int	Autoincrement	PK
idStudenta	Int		FK
idTypu	Int		FK
idZmenilTridni	Int		FK
castka	Float	NN	
doplnujiciInformace	Text		
datumPlatby	Date	NN	
posledniUprava	Timestamp		

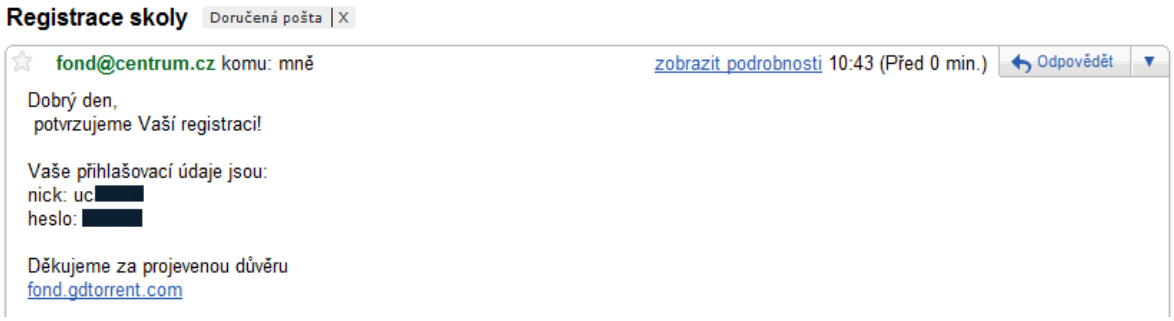
typIndividualniAkce

Tato tabulka obsahuje informace o druhu individuální akce (vklad, výplata, čerpání). Využívá ji tabulka INDIVIDUALNIAKCE.

název sloupce	Typ	Constraint	key
idTypu	Int	Autoincrement	PK
typ	Varchar(10)	NN, UNIQ	

Příloha B – Příklady emailů zasílaných aplikací

Registrace školy (třídy)



Registrace skoly Doručená pošta | X

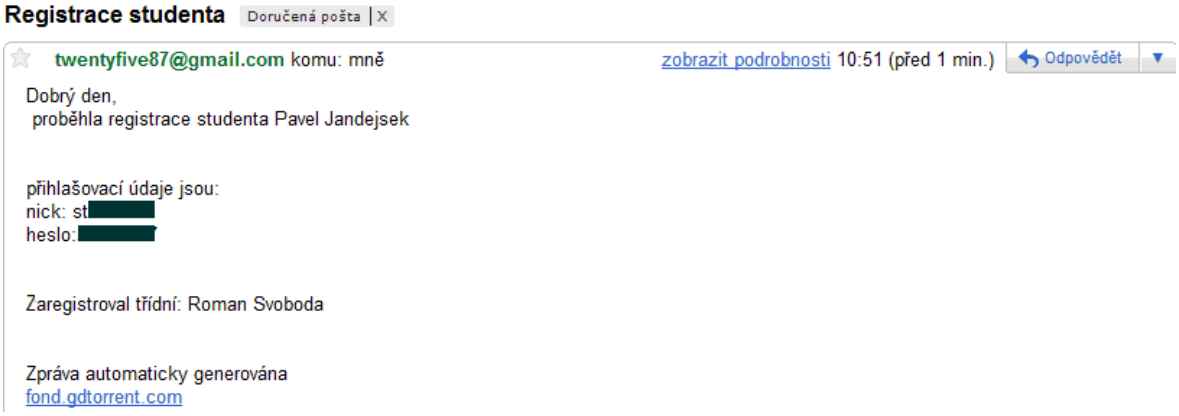
fond@centrum.cz komu: mně — zobrazit podrobnosti 10:43 (Před 0 min.) — Odpovědět

Dobrý den,
potvrzujeme Vaší registraci!

Vaše přihlašovací údaje jsou:
nick: uc[redacted]
heslo: [redacted]

Děkujeme za projevenou důvěru
fond.gdtorrent.com

Přidání studenta



Registrace studenta Doručená pošta | X

twentyfive87@gmail.com komu: mně — zobrazit podrobnosti 10:51 (před 1 min.) — Odpovědět

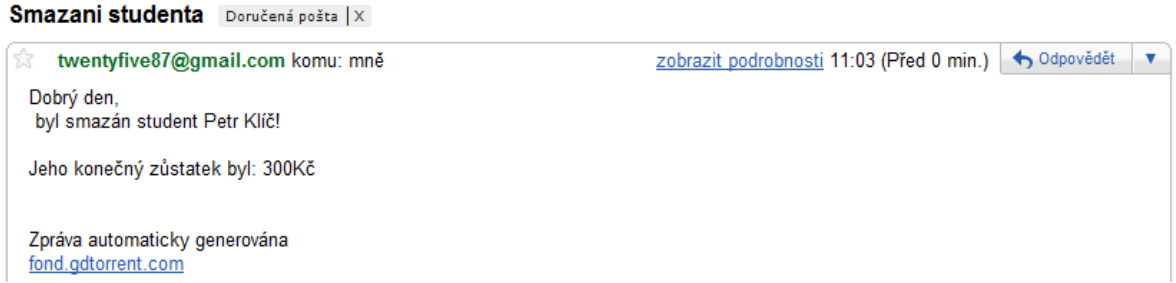
Dobrý den,
proběhla registrace studenta Pavel Jandejsek

přihlašovací údaje jsou:
nick: st[redacted]
heslo:[redacted]

Zaregistroval třídní: Roman Svoboda

Zpráva automaticky generována
fond.gdtorrent.com

Smazání studenta



Smazani studenta Doručená pošta | X

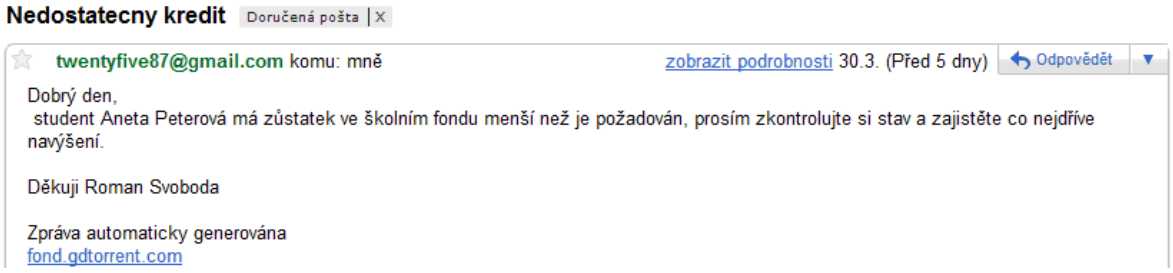
twentyfive87@gmail.com komu: mně — zobrazit podrobnosti 11:03 (Před 0 min.) — Odpovědět

Dobrý den,
byl smazán student Petr Klíč!

Jeho konečný zůstatek byl: 300Kč

Zpráva automaticky generována
fond.gdtorrent.com

Nedostatečný kredit



Nedostatecny kredit Doručená pošta | X

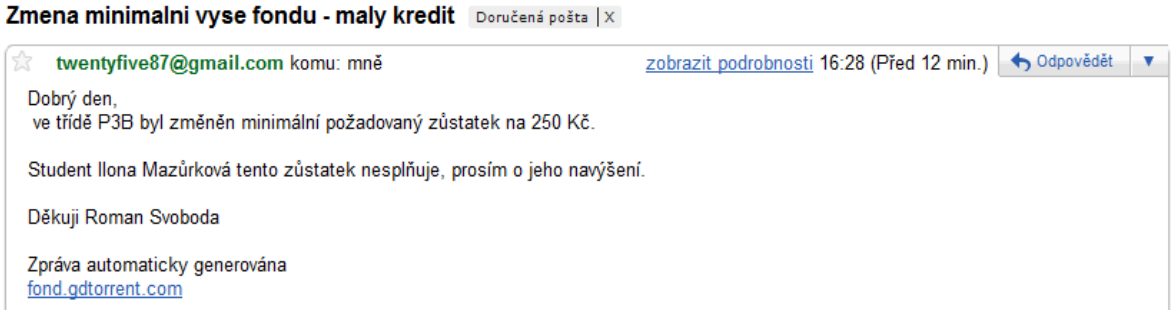
twentyfive87@gmail.com komu: mně zobrazit podrobnosti 30.3. (Před 5 dny) Odpovědět

Dobrý den,
student Aneta Peterová má zůstatek ve školním fondu menší než je požadován, prosím zkontrolujte si stav a zajistěte co nejdříve navýšení.

Děkuji Roman Svoboda

Zpráva automaticky generována
fond.gdtorrent.com

Změna minimální výše fondu studenta



Zmena minimalni vyse fondu - maly kredit Doručená pošta | X

twentyfive87@gmail.com komu: mně zobrazit podrobnosti 16:28 (Před 12 min.) Odpovědět

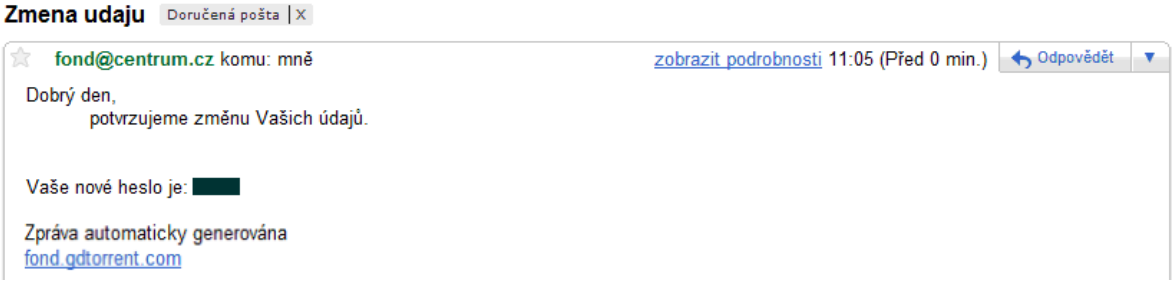
Dobrý den,
ve třídě P3B byl změněn minimální požadovaný zůstatek na 250 Kč.

Student Ilona Mazůrková tento zůstatek nesplňuje, prosím o jeho navýšení.

Děkuji Roman Svoboda

Zpráva automaticky generována
fond.gdtorrent.com

Změna osobních údajů



Zmena udaju Doručená pošta | X

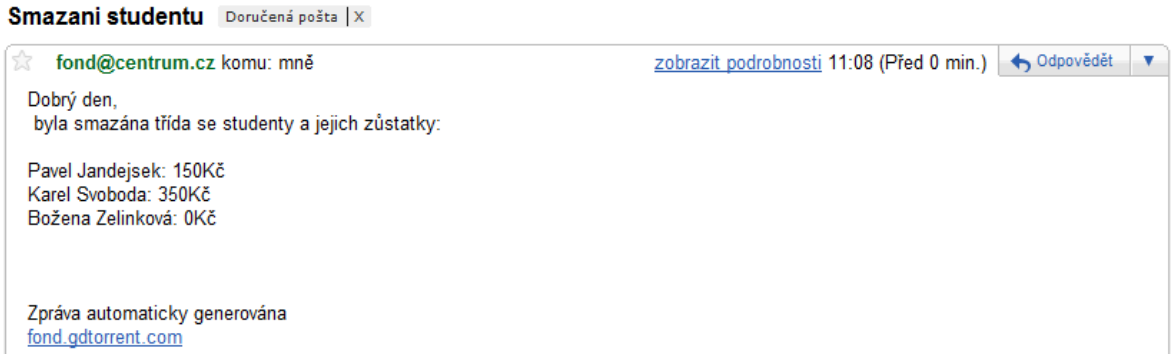
fond@centrum.cz komu: mně zobrazit podrobnosti 11:05 (Před 0 min.) Odpovědět

Dobrý den,
potvrzujeme změnu Vašich údajů.

Vaše nové heslo je: [redacted]

Zpráva automaticky generována
fond.gdtorrent.com

Smazání třídy



Příloha C – Instalační příručka

1. Nejprve je nutné zajistit hosting, který podporuje minimálně PHP verze 5, MySQL 4.1 a vyšší. Doporučuji hostingové služby na www.cesky-hosting.cz, popřípadě nalézt free hosting splňující tyto požadavky.
2. Do souboru password.php, který se nachází ve složce s aplikací /include/_private vyplňte vlastní přihlašovací údaje k databázi.
3. Do souboru kontakt.php, který se nachází v hlavním adresáři aplikace, vepište vlastní kontakt do označeného pole.
4. Vložte databázovou strukturu do své databáze. Toho docílíte zkopírováním scriptů ze souboru db_fond_startovaci.sql, například za využití MySQL Command Line Clienta.
5. Zkopírujte vše z hlavní složky aplikace na Váš hosting.
6. Ověřte funkčnost.