



## Středoškolská technika 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

# EDUCO - SYSTÉM TŘÍDNÍ SPRÁVY

Vojtěch Mašek

Střední průmyslová škola elektrotechnická,  
Praha 10, V Úžlabině

## Úvod

Na nápad vytvořit takovouto aplikaci mě přivedla moje sestra, která potřebovala pro svou třídu nějaký nástroj, s jehož pomocí by mohla snadno organizovat třídu, ve které je pokladníkem. Napadlo mě tedy, že spojím povinnost s užitečným a přihlásím se touto prací k DMP. Dalším rozhodujícím faktorem bylo pochopitelně to, že mě vývoj webových stránek baví a zajímá mě vše okolo webových technologií.

Aplikace je psána pomocí jazyků HTML, PHP, Javascript a SQL.

## 1 Design

### 1.1 Pozadí

Při nastavování pozadí stránky jsem použil CSS vlastnost „background“ a rozhodl jsem se použít modré pozadí s lehkým abstraktním ornamentem. Pro modrou barvu jsem se rozhodl proto, že je lidskému oku nejpříjemnější a na rozdíl od barvy zelené nebo červené má uklidňující vliv na mozek. Pod obrázek na pozadí jsem také nastavil jednolitou barvu podobnou té na obrázku, aby se při dlouhém načítání obrázku nezobrazovala pouze bílá plocha. Obrázek jsem se sice snažil optimalizovat pro přenos po webu, ale při přijatelné kvalitě obrazu jsem docílil zmenšení jen asi na 600 KB, takže přenos obrázku může chvíli trvat, nicméně po uložení do paměti cache prohlížeče již funguje vše velmi svižně.



Obrázek 1: Vzhled stránky

## 1.2 Barvy

Na webu jsou použity následující barvy: světle modrá, jejíž použití bylo vysvětleno v předchozím bodě, bílá jako podkladová barva těla stránky a barva písma na modrém pozadí a nakonec tmavě šedá jako barva písma v těle stránky, která koresponduje s modrou a bílou lépe než čistě černá barva.

## 1.3 Rozložení

### 1.3.1 Hlavička

Zvolil jsem jednoduché a esteticky čisté rozložení prvků webové aplikace. V hlavičce stránky se nachází název aplikace a po přihlášení se zde také zobrazují informace o přihlášeném studentovi, to znamená: název jeho školy, jeho třída, jméno a příjmení.

Po přihlášení studenta do systému se zde také začne zobrazovat menu, pomocí kterého může uživatel procházet všechny příspěvky v aplikaci. Po kliknutí na nápis „eDUCO“ je uživatel přesměrován podle toho, jestli je přihlášen, buď na nástěnku, nebo na přihlašovací stránku.

Písmo v hlavičce i patičce vrhá stín, aby bylo lépe čitelné na světle modrém pozadí. K tomu jsem použil CSS3 vlastnost „text-shadow“.

### 1.3.2 Tělo stránky

Celé tělo stránky má podobu bílého pole vrhajícího lehký stín, na pozadí je proto umístěna opakující se poloprůhledná textura, která je pouze 1 pixel vysoká, kvůli optimalizaci rychlosti

nahrávání. Vrchní a spodní část těla zasahuje zčásti do hlavičky a patičky kvůli přirozenému ukončení stínu.

### 1.3.2.1 Nadpis stránky

Nadpis stránky je řešen pomocí obrázku, z důvodu složité implementace vlastních fontů do webových stránek, obrázek je použit ve vlastnosti „background“ aplikované na nadpis úrovně 1. Každý takovýto nadpis má definován vlastní styl v souboru kaskádových stylů. Nadpis je oddělen od zbytku stránky vodorovnou čarou.

```
111 h1.nastenka
112 {
113     margin: 0px auto 20px auto;
114     height: 37px;
115     background: url("../img/h_nastenka.png") no-repeat top center;
116 }
```

Obrázek 2: Styl nadpisu

### 1.3.2.2 Sub menu

Tento oddíl stránky se nachází na její levé straně a zobrazují se v něm dodatečné volby pro procházení jednotlivých záložek, jako jsou například: způsob řazení příspěvků nebo volba předmětu, ze kterého chce uživatel zobrazit výpisky. Opět je oddělen čarou, tentokrát však svislou, aby byl odlišen od vlastního obsahu zobrazené stránky. Tento oddíl se zobrazuje pouze na stránkách po přihlášení, s výjimkou nástěnky a technické podpory, kde není potřeba. Oddíl má ve svém kaskádovém stylu vlastnost „float: left;“, která zajišťuje jeho pozici na levém okraji stránky.

### 1.3.2.3 Vlastní obsah stránky

V tomto oddíle se uživateli zobrazuje obsah stránky, kterou otevřel, a to jak při procházení příspěvků, tak v administraci (podrobnější popis různých typů obsahu je uveden v části „Sekce Aplikace“). Oddíl má ve svém kaskádovém stylu vlastnost „float: right;“, která zajišťuje jeho pozici na pravém okraji stránky. Oba plovoucí oddíly jsou poté ukončeny prázdným oddílem, který má nastavenou vlastnost „clear: both;“, což znamená, že se čeká s ukončením těla stránky, dokud neskončí tyto dva prvky, a teprve pak se smí vypsát patička.

### 1.3.3 Patička

Patička v nepřihlášeném stavu obsahuje odkaz na registraci nového žáka a odkaz na formulář technické podpory. V přihlášeném stavu se zobrazují odkazy na administraci obsahu, nastavení a také technickou podporu, aby se v případě potíží nemusel uživatel odhlašovat.

Dále obsahuje jméno autora, a informaci o autorských právech vztahujících se k aplikaci.

## 1.4 Testování

Funkčnost a vzhled stránek jsem testoval v následujících prohlížečích: Mozilla Firefox, Google Chrome, Opera, Safari, Internetový prohlížeč pro Symbian, Android browser, Mobile Internet Explorer, Internetový prohlížeč herní konzole Sony Playstation 3 a Internet Explorer verze 8, 9 a 10.

Ve všech zmíněných prohlížečích se obsah zobrazuje správně s výjimkou Internet Exploreru 8 a 9, kde se kvůli nekompatibilitě s CSS3 nezobrazuje stín pod písmem, to ale není žádná závažná chyba.

## 2 Sekce aplikace

### 2.1 Registrace

Pro používání aplikace se nejprve uživatel musí zaregistrovat. Proces registrace je rozdělen do tří částí, v první uživatel vybírá školu, kterou navštěvuje, pokud ji v seznamu nenalezne, klikne na možnost „jiná...“ a objeví se textové pole, kam může zadat nový název. V druhé části vybírá z obdobného seznamu svou třídu, opět má možnost, v případě že svou třídu v seznamu nenajde, zadat její název ručně. Ve třetí části registrace už uživatel zadává své osobní údaje a volí si přihlašovací jméno a heslo. Po odeslání nastává vyhodnocení celé registrace, všechny tři části registrace jsou ošetřeny proti odeslání prázdných polí, ve třetí části se navíc testuje správnost zadaného hesla porovnáním s kontrolním políčkem pro heslo, pokud se zadaná hesla neshodují, stránka vrátí chybu a uživatel musí heslo zadat znovu. Testuje se také správný tvar e-mailové adresy, to obstarává PHP funkce „preg\_match()“, která využívá regulárních výrazů. Pokud adresa není ve správném tvaru, stránka opět vrátí chybu. Kromě správnosti hesla a e-mailové adresy se testuje také unikátnost přihlašovacího jména (loginu), podle něčeho se totiž musí vyhodnotit, kdo se vlastně přihlašuje.

Pokud jsou všechna pole vyplněna správně, formulář se odešle a skript začne zapisovat do databáze. První na řadě je škola, v případě, že již škola existuje, je do proměnné uloženo její identifikační číslo (ID) z databáze, pokud uživatel vytvořil novou, musí se nejprve do databáze vložit a poté se také načte její ID do pomocné proměnné. Pokud uživatel v seznamu přehlédl svoji školu a zadal stejný název de facto podruhé, nic se neděje. Před zápisem do databáze se testuje, zda už škola se stejným názvem nebyla zapsána dříve. Pokud ano, do databáze se nic nezapisuje a vybere se ID školy se stejným názvem, toto opatření slouží k ochraně proti duplicitě. Stejně akce jako při výběru školy se provedou i při výběru třídy.

Jako další se provede zápis žáka do databáze. Ke správnému zařazení žáka do tříd se použijí hodnoty získané při volbě školy a třídy. Zadané heslo se „zahashuje“ pomocí funkce „md5()“, která neumožňuje zpětný převod na znaky (více o této funkci v kapitole „Přihlášení“). V případě, že uživatel právě založil novou třídu, zapíše se jeho ID do tabulky administrátorů a tím získá administrátorská práva (více o uživatelských právech v kapitole „Administrace“). Také se vygeneruje odkaz, pomocí kterého se mohou ostatní registrovat o něco snadněji, protože mohou úplně přeskočit výběr školy a třídy a přejít rovnou k zadání svých údajů. Tento odkaz se dá rozeslat všem členům třídy.

### 2.2 Přihlášení

Formulář pro přihlášení se zobrazí hned po vstupu na stránky. Uživatel se přihlašuje pomocí uživatelského jména a hesla. Přihlášení probíhá tím způsobem, že když uživatel odešle své přihlašovací údaje a spočítá se, kolik záznamů v databázi má stejnou položku „login“. Pokud je nulový, aplikace zahlásí chybu „Nesprávné uživatelské jméno nebo heslo“. Pokud je výsledné číslo rovno právě jedné, ale není správné heslo, také vypíše chybu. Na heslo se napřed aplikuje funkce „md5()“ a toto heslo poté porovná s tím uloženým v tabulce uživatelů. Pokud uživatel zadal správné údaje a jméno a heslo se shodují s těmi v databázi a ID uživatele

se nenachází v tabulce zablokovaných účtů (viz kapitola „Správa uživatelů“), načtou se veškeré informace o uživateli včetně jeho nastavených práv do superglobální proměnné „\$\_SESSION[]“ a tam zůstanou, dokud se uživatel neodhlásí, nebo dokud zcela nevypne prohlížeč. Pokud je uživatel přihlášen, nelze se na přihlašovací stránku dostat, protože se na jejím začátku testuje, zda je proměnná „\$\_SESSION[]“ použita a pokud ano, je uživatel přesměrován na stránku „Nástěnka“. To samé platí pro ostatní stránky, pokud je proměnná prázdná, uživatel je přesměrován na přihlašovací stránku. Je to proto, aby nedocházelo k chybám v běhu aplikace, jelikož celá její činnost se řídí právě údaji uloženými v proměnné „\$\_SESSION[]“.

## 2.3 Procházení příspěvků

### 2.3.1 Nástěnka

Tato stránka slouží k přehlednému zobrazení upozornění na domácí úkoly, ohlášené testy a nezaplacené finance, její obsah se až na všeobecná upozornění pro celou třídu, generuje automaticky. Vychází z údajů zadaných v tabulkách „ukoly“, „testy“, „finance“ a „finance\_stav“, kde hledá zápisy, jejichž datum platnosti je o jeden den vyšší než dnešní, nebo jako v případě financí, mají datum někdy v budoucnosti a přihlášený uživatel má u nich evidováno, že ještě nezaplátil. Zvláštní položkou na této stránce jsou již zmíněná všeobecná upozornění, která edituje uživatel. Jejich doba platnosti, tedy doba, po kterou se zobrazují, je 7 dní od data zveřejnění, toho jsem docílil pomocí SQL funkce „DATE\_SUB“, která od stávajícího data odečte 7 dní a pokud je datum zveřejnění upozornění v intervalu mezi výsledným a dnešním datem, pak se zobrazí, pokud ne, systém ho ignoruje. V případě, že se v jedné z těchto čtyř částí stránky nenachází žádné upozornění, vypíše se příslušná informativní hláška. To je ošetřeno SQL funkcí „COUNT“.

### 2.3.2 Výpisky

V záložce výpisky se zobrazují výpisky, které studenti dané třídy nahráli. Když uživatel otevře tuto záložku, je vyzván k výběru předmětu, seznam předmětů se nachází v levém menu. Seznam je vytvořen pomocí SQL funkcí „DISTINCT“, která vyřadí duplicitní položky, to je potřeba udělat, protože každý zápis má v sobě definován předmět, ze kterého výpisky jsou. Kvůli variabilitě aplikace není dán žádný pevný seznam vyučovaných předmětů.

Po výběru předmětu se upraví adresa a přidá se do ní část „GET“, na základě této proměnné se zobrazí tabulka se seznamem zápisů z příslušného předmětu. U každého z nich je možné stáhnout si zápis ve formátu HTML do počítače, také je možné stáhnout všechny zápisy najednou. Stahování zajišťuje skript download.php, který na základě proměnné „\$\_GET“ stáhne buď jeden, nebo všechny zápisy. Při stažení souboru používám hlavičku pro vynucené stažení.

```
12 header("Content-Description: File Transfer");
13 header("Content-Type: application/force-download");
14 header("Content-Disposition: attachment; filename=\"Výpisky ".$predmet.".html\");
```

Obrázek 3: Vynucené stažení

Po rozkliknutí konkrétního zápisu se opět změní v adrese stránky část „GET“ na ID příslušného zápisu a zobrazí se samotný zápis, který je opět možno nejen stáhnout, ale tentokrát i rovnou vytisknout. V souvislosti s tiskem jsem musel vytvořit tiskový styl, ve kterém jsou pomocí CSS vlastnosti „display: none;“ skryty všechny prvky stránek, které by v tisku nějakým způsobem překážely, což ve výsledku znamená, že zůstal zobrazen pouze

oddíl s vlastním obsahem stránky. Tisk dokumentu je zajištěn pomocí javascriptu, konkrétně metodou „window.print()“.

### 2.3.3 Domácí úkoly

Jednoduchý seznam zadaných domácích úkolů. Dělí se na dvě části, seznam aktuálních úkolů a archiv těch, které již měly být odevzdány. Seznamy se aktualizují vždy ve 12:00, to je zajištěno tím, že se každému uloženému úkolu nastaví kromě data odevzdání také čas 12:00. Záznamy v databázi se pak porovnávají s aktuálním datem a časem a podle toho se řadí buď do archivu, nebo do seznamu aktuálně platných zadání. Standardně jsou úkoly řazeny podle data vzestupně, takže se nejbližší úkoly zobrazují jako první, v archivu se pak řadí obráceně, tedy od těch, které měly být odevzdány jako poslední. V obou seznamech se dá stránkovat, stránkovací seznam se generuje v podstatě do nekonečna, takže velikost archivu je omezena pouze maximální velikostí databáze.

### 2.3.4 Testy

Seznam ohlášených testů je realizován obdobným způsobem jako domácí úkoly. Témata testů se stejně jako zadání domácích úkolů archivují. Mezi těmito dvěma kategoriemi je vlastně pouze jediný rozdíl a to ten, že k tématu testu je možno přidat ještě odkaz na výpisky nebo jiné studijní materiály k danému tématu. Tento odkaz se v seznamu sám zaktivní.

### 2.3.5 Finance

Seznamy financí slouží hlavně pokladníkům tříd pro evidenci zaplacených a nezaplacených položek. V levém menu se zobrazuje seznam položek a po kliknutí na jednu z nich se zobrazí stručný popis toho, za co se vybírají peníze, konkrétní částka a celá přehledná tabulka s dvěma sloupci, v jednom je příjmení a jméno a ve druhém stav platby u příslušného žáka.

V této části se využívají dvě tabulky v databázi. V jedné jsou uloženy informace o dané položce a ve druhé informace o stavu platby všech uživatelů, kterých se tato položka týká. Tabulky jsou de facto propojeny skrze číslo položky, protože v druhé tabulce má každý řádek dáno, ke kterému číslu položky se váže. Výpis tabulky probíhá tak, že se nejprve z databáze vyberou informace o položce včetně jejího čísla a následně se vybere seznam všech žáků ze třídy. Poté se v cyklu hledají v tabulce „finance\_stav“ záznamy, které vyhovují podmínce, že patří ke konkrétní položce a zároveň ke konkrétnímu žákovi ze třídy. Na základě hodnoty v tomto záznamu se pak do tabulky vypíše zatržítko, křížek nebo pomlčka. Tyto symboly udávají, zda žák zaplatil, nezaplatil nebo se od něj peníze na tuto položku v seznamu financí nevybírají.



Obrázek 4:

Symboly pro označení stavu platby

### 2.3.6 Soubory

Zprovoznit datové úložiště byl trochu problém, původně jsem chtěl pro nahrávání souborů použít nějaký předpřipravený zásuvný modul (plug-in), ale nakonec se ukázalo, že bude jednodušší naprogramovat si vlastní, než implementovat nějaký již hotový systém (více v kapitole administrace).

Nicméně, v této záložce má uživatel – student přístup jak k souborům, které nahráli jeho spolužáci, tak ke svým soukromým studijním materiálům. Soubory jsou řazeny do přehledné tabulky, která obsahuje název souboru, datum nahrání, velikost souboru v KB a odkaz ke stažení. O stažení souboru se stará opět skript „download.php“, který na základě části „GET“ v adrese stáhne příslušný soubor. Pro stahování netextového souboru jsem ale musel pozměnit hlavičku ve zmíněném skriptu (viz obrázek 5). Soubory je možné řadit podle názvu, velikosti data nahrání a podle typu.

```
35 $soubor = $_GET["soubor"];
36 $tabulka = mysql_query("SELECT NAZEV, TYP FROM soubory WHERE ID = '$soubor'");
37 $radek = mysql_fetch_assoc($tabulka);
38 header("Content-Description: File Transfer");
39 header("Content-type: octet/stream");
40 header("Content-Disposition: attachment; filename=\"".$radek["NAZEV"]."\"".$radek["TYP"]."\"");
41 header("Content-Length: ".$filesize("files/".$soubor));
```

Obrázek 5: Stažení souboru

## 2.3.7 Odkazy

V tomto seznamu jsou odkazy na osvědčené studijní materiály a stránky z webu. Eviduje se datum přidání, název a odkaz samotný. Záznamy se opět dají řadit podle názvu a data přidání. Řazení je realizováno pomocí rozbalovacího seznamu, který má na sobě aplikovaný javascript, konkrétně při změně vybrané hodnoty v seznamu se vyvolá událost „onchange“ a pomocí proměnné „\$\_GET“ se upraví způsob řazení.

## 2.4 Administrace

### 2.4.1 Upozornění

Tento formulář obsahuje pouze jedno vstupní pole (textarea) a pomocí metody POST se odešlou zpět na ten samý skript (tento způsob používám u všech formulářů kvůli úspoře místa a snížení počtu souborů, zpracování tímto způsobem také umožňuje, aby v případě chybného zadání nezmizely z formuláře všechny zadané údaje). Na začátku skriptu je podmínka, která testuje stisknutí tlačítka odeslání, pokud je stisknuto, zkontrolují se vstupy, zda nejsou prázdné a pokud jsou všechny v pořádku, provede se dotaz „INSERT INTO“, který do databáze запиše příslušný řádek, v tomto případě všeobecné upozornění.

Upozornění se samozřejmě dají upravovat a mazat, tato část je ale vždy stejná a vychází z formuláře pro vložení, proto je popsána v dalším bodě a dále už je opomíjena.

### 2.4.2 Výpisky

Administrace všech položek v aplikaci se dělí na dvě části, část kde můžete přidat nový příspěvek a část sloužící k editaci nebo mazání starších příspěvků.

Pro vkládání výpisků používám v aplikaci WYSIWYG editor TinyMCE. Ten se po jednoduché implementaci sám aplikuje na všechny prvky typu „textarea“, které se na stránce vyskytují. Umožňuje tak formátovat text přímo v prohlížeči a není nutné ho nijak složitě převádět do HTML, které pokud je vygenerováno nějakým textovým procesorem, má většinou podobu nesrozumitelných shluků všech možných parametrů. TinyMCE generuje čistý kód, pak stačí na stránce pouze vypsát (nepotřebuje žádné dodatečné formátování) a který v databázi nezabere zdaleka tolik místa jako výše zmíněný kód z textových procesorů typu MS Word apod.



Pro správné zařazení zápisu je potřeba zadat zkratku předmětu (např.: MAT, FYZ, ČJL, atd.). Dále musí uživatel zadat datum, kdy byly výpisky pořízeny. Automatické datum se nezadává z toho důvodu, že by mohlo docházet k chybnému řazení výpisků. Kromě samotného zápisu látky je potřeba zadat ještě nadpis. Ten slouží k identifikaci zápisu v seznamu výpisků.

Formulář je pochopitelně ošetřen proti odeslání prázdných polí a zobrazuje, jak dopadlo odeslání. To samé platí pro druhou (editační) část administrační stránky.

Ve spodní části stránky se zobrazuje tabulka s odkazy na úpravu nebo smazání daného příspěvku. Pro obě akce se volá skript z jiné stránky. Pokud chce uživatel příspěvek smazat, je ještě jednou dotázán, zda si je jistý. Toto je vyřešeno opět pomocí javascriptu, funkce „confirm()“ vyvolá otevření potvrzovacího dialogu, který pokud je zamítnut, vrátí se na původní stránku, pokud je potvrzen, pokračuje dále na skript pro smazání. Ten nemá žádný textový výstup, proto si jeho existence běžný uživatel nemá šanci všimnout, vidí až návrat zpět do administrace. Po smazání se pochopitelně zobrazí hláška „Zápis byl smazán“.

Pro úpravu stávajícího zápisu slouží samostatný formulář, který je totožný s tím pro vložení, jen s tím rozdílem, že se do něj automaticky načtou data v závislosti na tom, který zápis si uživatel přeje upravit. Tato data se nejprve načtou z databáze a poté se vloží do vstupních polí pomocí jejich atributu „value“. Samotná změna dat v databázi se provede SQL dotazem „UPDATE“. Pokud se z nějakého důvodu změna nepodaří (nejčastěji vinou nevyplněného pole), je uživatel vrácen na formulář, kde zůstanou vyplněny hodnoty, které zapsal a může je upravit a formulář znovu odeslat. Tato část je u všech položek v administraci stejná, proto už ji nebudu v dalších bodech popisovat.

Uživatel může upravovat a mazat pouze výpisky, které sám vytvořil. Výjimku tvoří administrátoři, kteří mají přístup ke všem výpiskům od všech žáků ve třídě. Z toho jasně vyplývá, že v databázi se uchovává i autor výpisků.

### **2.4.3 Domácí úkoly**

Sekce domácí úkoly je přístupná všem uživatelům ve třídě, každý tedy smí zadávat do systému nové úkoly, editovat je či odstraňovat. Databázová věta obsahuje především zkratku předmětu, ze kterého je úkol zadán, datum odevzdání a samotné zadání domácího úkolu. K zadání data je zde (stejně jako ve všech ostatních formulářích) používán javascriptový kalendář Tigr Calendar. Ten má ošetřeny všechny zákonitosti, které při výběru data musí platit, jako například že měsíc únor nemůže mít nikdy třicet dní. Na vstupní pole data je aplikován parametr „readonly“, aby uživatel nemohl nedopatřením přepsat datum a způsobit tím chybu.

### **2.4.4 Testy**

Administrace testů je velmi podobná administraci domácích úkolů, jen s tím rozdílem, že se do databáze ukládá i odkaz na příslušné studijní materiály, ten je ale nepovinný.

### **2.4.5 Finance**

Do této části administrace mají přístup pouze uživatelé s administrátorskými a pokladnickými právy, jinak by si každý student mohl libovolně upravovat, stav svých plateb a to je nežádoucí. Pokladník do formuláře zadává následující údaje: název položky, datum, do kdy je potřeba finance vybrat, stručný popis položky a konkrétní částku, kromě těchto zadává ještě počáteční stav u každého studenta. Tento seznam se generuje automaticky ze seznamu žáků



třídy a stav zaplacení se udává pomocí přepínačů („radio button“). Celý seznam se vypisuje v cyklu a výchozí hodnota u všech žáků je „Nezaplaceno“.

Jako vždy je formulář ošetřen proti odeslání prázdných vstupních polí a je možnost upravit nebo smazat stávající seznam.

## 2.4.6 Soubory

Nahrávání souborů do datového úložiště jsem řešil pomocí formuláře se vstupním polem typu „file“. U nahrávaného souboru lze také nastavit práva pro čtení, buď může být soubor veřejný, což znamená, že si ho může stáhnout kdokoli, nebo může být označen jako soukromý a v tom případě se zobrazí jen tomu, kdo jej nahrál. Administrátor třídy má sice právo soubor přejmenovat, nebo odstranit, ale ani on se nedostane k jeho obsahu. Je tak zaručena určitá forma diskrétnosti pro každého uživatele. Nastavení soukromého souboru můžou studenti využít, například pokud si takto posílají do školy svou práci, ale nechtějí, aby ji někdo okopíroval.

Při odeslání vyplněného formuláře se obslouží událost „onsubmit“ a zobrazí se vyčkávací kolečko (viz obr. 6). Kolečko se zobrazí pomocí změny jeho CSS vlastnosti „display“ z „none“ na „block“.



Obrázek 6: Vyčkávací kolečko

Po nahrání souboru do paměti serveru je přepsán na pevný disk a údaje o něm, jako jsou název, velikost, typ a datum nahrání jsou zapsány do databáze. Název souboru na disku se změní na jeho identifikační číslo v databázi, které je unikátní, to proto, aby nedocházelo ke kolizím vinou stejných názvů souborů (při stažení se soubor přejmenuje zpět).

## 2.4.7 Odkazy

Vkládání odkazů se řeší formulářem s dvěma vstupními poli: nadpis odkazu a odkaz samotný. Při zápisu do databáze se jedná o triviální úkon, provede se dotaz „INSERT INTO“ a odkaz se uloží do tabulky. Není zde co jiného řešit.

## 2.5 Nastavení

### 2.5.1 Změna hesla

Pokud by chtěl uživatel změnit své heslo, například při podezření, že mu někdo jeho přihlašovací údaje zcizil, může si ho změnit právě zde. Pro změnu je potřeba zadat staré heslo a dvakrát stejné nové heslo. Formulář tedy obsahuje tři vstupní pole typu „password“, ošetření prázdných vstupů je samozřejmostí. Pokud staré heslo nesouhlasí, změna se neprovede a vypíše se chybová hláška. Na vstupní pole jsem aplikoval atribut „autocomplete=off“ proto, aby se heslo nedoplňovalo samo z paměti prohlížeče a uživatel ho musel skutečně sám napsat.

## 2.5.2 Změna osobních údajů

Tuto volbu uživatel použije, změní-li se některý z jeho osobních údajů, jako je jméno nebo e-mailová adresa. Změna jména je sice nepravděpodobná a e-mailová adresa se využívá jen ve formuláři technické podpory, ale v případě potřeby je dobré mít aktuální kontaktní údaje. Navíc ve všeobecných podmínkách používání je napsáno, že pokud se tyto podmínky změní, musí být uživatelé informováni pomocí elektronické pošty.

Ve vyhodnocení formuláře jsou ošetřeny prázdné vstupy a také se funkcí „preg\_match()“ kontroluje správnost tvaru e-mailové adresy.

## 2.5.3 Změna uživatelských práv

Zde může administrátor třídy, jednoduše pomocí zaškrtačkových políček přidávat a odebírat administrátorská a pokladnická práva a také může zablokovat přístup uživatelům, kteří nedodržují všeobecné podmínky používání (viz příloha 1 – Všeobecné podmínky používání a prohlášení o ochraně osobních údajů). Vyhodnocení formuláře pak probíhá velmi jednoduše, nejprve se všechna práva daných uživatelů odeberou a poté se opět do příslušných tabulek v databázi zapisou, těm uživatelům, kteří mají zaškrtnutá políčka. Jediný, u koho administrátor nemůže tato práva odstranit nebo měnit, je on sám. Vždy totiž musí být ve třídě alespoň jeden administrátor. Pokud se chce této funkce vzdát, musí nejprve (pokud to už neudělal) jmenovat jiného administrátora a poté ho požádat aby mu jeho práva odebral.

## 2.5.4 Změna označení třídy

Primitivní skript pro změnu názvu třídy, když začne nový školní rok. Uživatel pouze přepíše v textovém políčku název třídy, odešle formulář, a pokud není pole prázdné, v databázi se přepíše hodnota SQL dotazem „UPDATE“ a upraví se i hodnota v „\$\_SESSION[]“, aby se změna projevila ihned a ne až při dalším přihlášení. Pokud je pole prázdné, v databázi se nic neděje a formulář zahlásí chybu „Nebylo zadáno nové označení třídy“. Tuto změnu však může udělat pouze administrátor.

## 2.5.5 Správa uživatelů

Zde může uživatel, pokud má administrátorská práva, spravovat uživatele ve své třídě. Pokud například někdo přestupuje na jinou školu, nebo propadl do nižšího ročníku, je potřeba ho ze stávající třídy odstranit. Žádný uživatel bohužel nemůže žáka jen tak „přesunout“ do jiné třídy, proto je potřeba celý jeho účet smazat, aby si následně mohl vytvořit nový ve svojí nové třídě. Odstraněním uživatele nedojde ke ztrátě dat, která vytvořil nebo nahrál. Této funkce se dá využít i tehdy, když se do třídy přihlásí někdo, kdo do ní nepatří (ať už se přihlásil nedopatřením, nebo schválně za účelem poškodit data, což by se mu ale stejně nepodařilo). Odstranění uživatele vyžaduje potvrzení heslem, jako ochranu proti nechtěnému „kliknutí“ anebo útoku někoho, kdo chce třídu poškodit.

## 2.6 Technická podpora

Tato část aplikace slouží pro kontakt s hlavním administrátorem nebo provozovatelem aplikace. Pokud je uživatel přihlášen, automaticky se vyplní jeho e-mailová adresa. Do textového pole pak napíše popis problému, který nastal, a ten se následně pomocí funkce „mail()“ odešle na nastavenou adresu.

```

47     $headers = "Content-type: text/html; charset=utf-8" . "\r\n";
48     $headers .= "From: eDUCO <".$_POST['email'].>" . "\r\n";
49     mail("m.games@email.cz","Technická podpora","<p>Z portálu eDUCO bylo odesláno
50     chybové hlášení:<br>".$_POST['text'].<br>E-mail odesílatele:
51     ".$_POST['email'].</p>",$headers);

```

Obrázek 7: Odeslání e-mailu

Formulář lze využít i pro žádost o vygenerování nového hesla, ale další postup už záleží na tom, jestli uživatel, který toto žádá, potvrdí svou identitu, či nikoliv. Pokud ano, dá se uvažovat o vytvoření nových přihlašovacích údajů namísto těch zapomenutých, ale jednodušší způsob je požádání administrátora své třídy o smazání stávajícího účtu aby si uživatel mohl vytvořit nový a nebyl ve třídě registrován dvakrát.

### 3 Nápověda

Nápověda je realizována pomocí bublin, které se zobrazí po najetí myši na ikonku otazníku. Tyto ikonky jsou umístěny v sekci „Administrace“ a poskytují vždy uživateli nápovědu k danému poli. Pokud si uživatel neví rady s jejich vyplněním, jen najede na otazníček a zobrazí se nápověda. Pro zobrazení bublin je použit alternativní popisek a titulek obrázku (parametry „alt“, „title“).



Obrázek 8: Ikona pro zobrazení nápovědy

### 4 Validita

Všechny stránky v aplikaci jsem testoval, zda je jejich HTML výstup (pokud nějaký mají) validní. W3C HTML validátor však hlásí chyby typu „zastaralý způsob zápisu“ u některých prvků stránky, například u buněk tabulky, kde nastavuji jejich šířku pomocí atributu „width“ a ne pomocí kaskádových stylů. Přesto se na funkčnosti stránek ani na jejich vzhledu nic nemění a vše se zobrazuje v pořádku. Tato chyba má své opodstatnění, jelikož jsem stránky nadeklaroval jako HTML5 a v této verzi jazyka by se opravdu měly používat kaskádové styly, stejně si ale myslím, že by validátor neměl tento způsob zápisu vyhodnocovat jako závažnou chybu, ale měl by „vyhodit“ pouze upozornění na zastaralý způsob zápisu. Kromě této chyby prošla většina dokumentů bez poskvrny a v HTML 4.01 Transitional by se jevíly jako plně validní.

Kaskádové styly jsem také zkontroloval CSS validátorem a všechny (styl pro zobrazení v prohlížeči i pro tisk) byly vyhodnoceny jako CSS3 validní.

### Závěr

Z mého pohledu jsem zadání splnil takřikajíc do puntíku. Během tvorby jsem se přiučil několika novým věcem, které se mi jistě ještě někdy budou hodit, například upload souborů nebo práce se systémem práv. Mám v plánu tento systém dále upravovat a vylepšovat, aby snad mohl konkurovat dalším podobným webovým aplikacím. Snad se mi v dohledné době podaří naprogramovat i mobilní verzi, možná i aplikaci pro systém Android.

# Seznam použité literatury

TinyMCE javascript WYSIWYG Editor. [online] [cit. 14. 10. 2012]. Dostupné z:  
<<http://www.tinymce.com/download/download.php>>

Tigra Calendar. [online] [cit. 14. 10. 2012]. Dostupné z:  
<[http://www.softcomplex.com/products/tigra\\_calendar/](http://www.softcomplex.com/products/tigra_calendar/)>

PHP File upload. [online] [cit. 3. 1. 2013]. Dostupné z:  
<[http://www.w3schools.com/php/php\\_file\\_upload.asp](http://www.w3schools.com/php/php_file_upload.asp)>

Javascript – událost onsubmit. [online] [cit. 5. 1. 2013]. Dostupné z:  
<<http://www.jakpsatweb.cz/javascript/formular-zruseni.html>>

IBU, Changing visibility using javascript. [online] [cit. 5. 1. 2013]. Dostupné z:  
<<http://stackoverflow.com/questions/6205148/changing-visibility-using-javascript>>

\*LOOK\*, Select - odkazovani - onchange. [online] [cit. 10. 2. 2013]. Dostupné z:  
<<http://www.jaknaweb.com/ukazuj.php?f=1&i=166068&t=166064&v=f>>