

Webová aplikace pro podporu a evaluaci výuky

A Web Application for the Support and Evaluation
of Tuition Processes

Bc. Jiří Ančinec

Diplomová práce
2013



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jiří ANČINEC**
Osobní číslo: **A10477**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Učitelství informatiky pro střední školy**
Forma studia: **prezenční**

Téma práce: **Webová aplikace pro podporu a evaluaci výuky**

Zásady pro vypracování:

1. Zpracujte analýzu hlavních požadavků na webovou aplikaci pro podporu a evaluaci výuky.
2. Navrhněte strukturu databáze a webové aplikace.
3. Vytvořte aplikaci a popište postup řešení.
4. Věnujte pozornost zabezpečení aplikace.
5. Navrhněte a implementujte do aplikace příklady vzorových evaluačních dotazníků.
6. Implementujte a otestujte vytvořenou aplikaci na ukázkovém internetovém serveru či na serveru vybrané školy.
7. Vytvořte kompletní dokumentaci ke správě a používání aplikace.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. RESIG, John. JavaScript a Ajax: moderní programování webových aplikací. Vyd. 1. Překlad Ondřej Baše, Ondřej Žižka. Brno: Computer Press, 2007, 360 s. ISBN 978-802-5118-245.
2. LACKO, L'uboslav. PHP a MySQL: hotová řešení. 1. vyd. Brno: CP Books, 2005, 299 s. ISBN 80-251-0397-8.
3. BRÁZA, Jiří. PHP 5: začínáme programovat. 1. vyd. Praha: Grada Publishing, 2005, 244 s. ISBN 80-247-1146-X.
4. ASLESON, Ryan. AJAX: vytváříme vysoce interaktivní webové aplikace. Vyd. 1. Překlad Jakub Zemánek. Brno: Computer Press, 2006, 269 s. ISBN 80-251-1285-3.
5. VRÁNA, Jakub. 1001 tipů a triků pro PHP. Vyd. 1. Brno: Computer Press, 2010, 456 s. ISBN 978-802-5129-401.
6. VAŠŤATKOVÁ, Jana. Úvod do pedagogické evaluace a jejích metod: studijní texty pro distanční vzdělávání : projekt Školský management. 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 2010, 75 s. ISBN 978-802-4426-815.
7. CHRÁSKA, Miroslav. Metody pedagogického výzkumu: základy kvantitativního výzkumu. Vydání 1. Praha: Grada Publishing, 2007, 265 s. ISBN 978-802-4713-694.

Vedoucí diplomové práce: **Mgr. Lubomír Sedláček, Ph.D.**
Ústav matematiky

Konzultant: **Mgr. Michal Mikláš**

Datum zadání diplomové práce: **22. února 2013**

Termín odevzdání diplomové práce: **22. května 2013**

Ve Zlíně dne 22. února 2013


prof. Ing. Vladimír Vašek, CSc.
děkan




doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem této diplomové práce je vytvořit moderní webovou aplikaci, která má za úkol usnadnit učitelům činnosti ve výuce jako jsou prezentace nového učiva, sběr domácích úkolů či připomínání nejrůznějších událostí. Aplikace také poskytuje učitelům prostředky pro evaluaci výuky a autoevaluaci studentů. V teoretické části jsou přiblíženy pojmy evaluace a e-learning. Dále jsou zde popsány použité technologie. V praktické části je popsána analýza požadavků, řešení aplikace a konečná implementace na testovací server.

Klíčová slova: Evaluace, výuka, e-learning, PHP, MySQL, jQuery, Nette, webová aplikace.

ABSTRACT

The aim of this thesis is creating a modern web application which can facilitate the activities in tuition processes like presentation of new curriculum, collecting of homework or showing of reminders. The application provides tools for evaluation of tuition processes and autoevaluation of students. In theoretical part, expressions like evaluation and e-learning are explained. In this part, technologies which were used in this application are described. Requirements analysis, application solution and final implementation on the test server are described in the practical part.

Keywords: Evaluation, tuition process, e-learning, PHP, MySQL, jQuery, Nette, web application.

Děkuji vedoucímu mé diplomové práce panu Mgr. Lubomíru Sedláčkovi, Ph.D. za pomoc a připomínky k mé diplomové práci. Také děkuji panu Mgr. Michalu Miklášovi za pomoc s praktickou částí.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 EVALUACE	11
1.1 PEDAGOGICKÁ EVALUACE.....	11
1.1.1 Vymezení pojmu ‚pedagogická evaluace‘	11
1.1.2 Hodnocení vs. evaluace.....	12
1.2 DRUHY EVALUACE	13
1.2.1 Členění z hlediska hodnotitele	14
1.2.2 Rozdělení dle hlediska účelu.....	14
1.2.3 Rozdělení dle interpretace výsledků	14
1.2.4 Rozdělení z hlediska času	15
2 E-LEARNING	16
2.1 HISTORIE	17
2.2 FORMY E-LEARNINGU.....	17
2.3 ROLE UČITELE V E-LEARNINGU	17
3 POUŽITÉ TECHNOLOGIE	19
3.1 HTML.....	19
3.1.1 Historie	19
3.2 KASKÁDOVÉ STYLY.....	19
3.2.1 Twitter bootstrap	20
3.3 JAVASCRIPT	21
3.3.1 jQuery.....	21
3.3.2 Asynchronous Javascript and XML	22
3.3.3 Highcharts	23
3.4 PHP.....	24
3.4.1 Historie	24
3.4.2 Nette Framework.....	25
3.4.2.1 Presenter.....	26
3.4.2.2 Komponenty.....	27
3.5 MYSQL	27
3.5.1 PDO.....	28
3.5.2 Nette database a NotORM	28
II PRAKTICKÁ ČÁST	29
4 ANALÝZA POŽADAVKŮ	30
4.1 OBECNÉ POŽADAVKY	30
4.2 KURZY	31
4.3 SKUPINY.....	31
4.3.1 Studenti ve skupině	31
4.3.2 Připomínky	31
4.3.3 Vzkazy.....	32
4.3.4 Úkoly.....	32

4.4	SPRÁVA DOTAZNÍKŮ	33
4.4.1	Vytvoření.....	33
4.4.2	Odpovídání	33
4.4.3	Vyhodnocení	33
5	NÁVRH STRUKTURY APLIKACE	34
5.1	ADRESÁŘOVÁ STRUKTURA.....	34
5.1.1	app	34
5.1.2	www	35
5.1.3	libs	35
5.1.4	files	36
5.1.5	log.....	36
5.1.6	temp.....	36
5.2	NÁVRH DATABÁZE	36
6	ŘEŠENÍ APLIKACE	43
6.1	ÚVODNÍ STRÁNKA	43
6.2	SKUPINY.....	44
6.3	LEKCE	44
6.4	ÚKOLY	44
6.5	UPOZORNĚNÍ	45
6.6	ZPRÁVY.....	46
6.7	NASTAVENÍ	47
6.8	DOTAZNÍKY.....	47
6.9	VZOROVÉ EVALUAČNÍ DOTAZNÍKY	47
7	ZABEZPEČENÍ APLIKACE	49
7.1	CROSS-SITE SCRIPTING	49
7.2	SQL INJECTION	50
7.3	CROSS-SITE REQUEST FORGERY	50
7.4	OCHRANA APLIKACE VE FRAMEWORKU NETTE	52
8	IMPLEMENTACE APLIKACE NA INTERNETOVÝ SERVER.....	53
8.1	TESTOVÁNÍ APLIKACE	53
ZÁVĚR	55	
CONCLUSION	56	
SEZNAM POUŽITÉ LITERATURY.....	57	
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	59	
SEZNAM OBRÁZKŮ	60	
SEZNAM TABULEK.....	61	
SEZNAM PŘÍLOH.....	62	

ÚVOD

Při výuce se učitel potýká s řešením nejrůznějších problémů a otázek. Jedná se například o to, jak předat učivo žákům, jak jim vysvětlit novou látku, případně jak zjistit, že žáci danému učivu rozumí. V neposlední řadě učitele zajímá, jak studenti vyučovací hodinu vnímají a jaký z ní mají prožitek. Zkušený pedagog zajisté dokáže třídu pozorovat a na mnohé kladené otázky si takto odpovědět. Ovšem vždy existuje riziko, že někteří žáci se neprojeví dostatečně na to, aby učitel jejich postoje během výuky postřehl a mohl na ně náležitě reagovat.

K tomuto účelu je vhodné se žáků dotázat přímo – například pomocí dotazníkových formulářů. Pravdou je, že vyplnění takového formuláře zabere jistý čas. Další čas zabere také jeho vytvoření a následně i vyhodnocení. Tyto potíže při získávání cenných informací se snažím odbourat pomocí aplikace, která je cílem této diplomové práce. Jedním z jejích hlavních pilířů je dotazníkový systém, který učiteli maximálně zjednodušuje vytvoření dotazníku a jeho následné předložení žákům. Vše probíhá on-line v prostředí, ve kterém učitel může následně tento dotazník vyhodnotit – třeba i přímo během výuky a následně, pokud je toto vhodné, žáky seznámit s výsledky dotazníku.

V dnešní době je stále větší důraz kladen na moderní technologie a s tím spojený přesun informací a nejrůznějších činností na internet. Ne jinak je tomu i ve školství a výuce obecně. S postupem času vzniklo mnoho e-learning systémů. Spousta z nich je využívána celosvětově a mají i velkou uživatelskou podporu. Nejznámější z nich je asi systém Moodle, který používá i škola, pro kterou je určena aplikace, jež je cílem mé diplomové práce. Jednou z nevýhod tohoto systému je paradoxně jeho robustnost. Obsahuje velkou spoustu rozšíření a možností, ve kterých je těžké se vyznat. Nastavování je zdlouhavé a ne vždy se podaří docílit kýženého výsledku. Proto jsem se rozhodl vytvořit aplikaci přímo na míru dle potřeb dané školy. Aplikaci, která bude umět právě to, co učitel potřebuje – dle jeho požadavků. Což je mimo jiné požadavky na prezentaci učiva, na práci s odevzdanými úkoly nebo také k umožnění soukromé komunikace mezi učitelem a žákem přímo v hodině.

I. TEORETICKÁ ČÁST

1 EVALUACE

Jednou ze základních náplní aplikace, která je cílem této diplomové práce, je podpora evaluace výuky. Nejprve je potřeba přiblížit význam pojmu *evaluace* a *pedagogická evaluace*.

Pojem *evaluace* je v českém jazyce poměrně nový. Vznikl z latinského slovesa *valere* (v překladu: být silný, mít platnost), které bylo následně přeloženo do francouzštiny jako *évaluer* (hodnotit, oceňovat) a odtud do angličtiny jako *evaluation* (určení hodnoty, ocenění). V češtině se používal pojem *hodnocení*, který, jak je ukázáno dále, je s pojmem *evaluace* ve velké míře synonymní.

1.1 Pedagogická evaluace

V současné době je pojem pedagogická evaluace jedním z nejčastěji skloňovaných pojmů v pedagogické teorii i praxi.

„Evaluace je odborníky považována za vědní disciplínu. V posledních letech se rozvíjí vědní obor pedagogická evaluace, neboť současná pedagogika chápe teorii a praxi evaluace jako jednu ze svých součástí. Jedná se o vědecko-praktický obor zabývající se zjišťováním, porovnáváním, hodnocením a vysvětlováním kvality a efektivity edukace.“ (Vašátková, 2010, s. 7)

1.1.1 Vymezení pojmu ‚pedagogická evaluace‘

Průcha (1996, s. 10) tento pojem definuje pomocí souborů distinktivních rysů:

1. Pedagogická evaluace je především teoretický přístup. Je to koncepce, podle níž veškeré jevy vzdělávací reality (vzdělávací procesy, jejich programy a fungování, vzdělávací výsledky atd.) mohou a musí být určitými způsoby hodnoceny.
2. Pedagogická evaluace je současně metodologie, tj. soubor instrumentů (speciálních metod, technik) a profesních konvencí k aplikaci těchto instrumentů pro účely realizace přístupu.
3. Pedagogická evaluace je proces, tj. soubor aktivit zajišťovaných institucionální a organizační infrastrukturou výzkumu, kterými se realizuje v praxi přístup pomocí metodologie. Tento proces se zaměřuje na zjišťování a analýzu dat odrážejících stav či vývoj určitých jevů vzdělávací reality, tj. na monitorování a měření těchto jevů.

4. Pedagogická evaluace jakožto proces se uskutečňuje na různých úrovních vzdělávací praxe – od hodnocení jednotlivců či jednotlivých vzdělávacích programů až po evaluaci výsledků na úrovni celé národní vzdělávací soustavy či mezinárodní evaluaci vzdělávacích soustav mnoha zemí.
5. Pedagogická evaluace má různé způsoby využití, a to jak pro účely vědecké a výzkumné, tak pro účely praktické.

„Ve vědecké terminologii má *evaluace* obecný význam *hodnocení*. V pedagogice znamená zjišťování, porovnávání a vysvětlování dat charakterizujících stav, kvalitu, fungování, efektivnost škol, částí nebo celku vzdělávacího systému... Má důležitou roli pro korekce a inovace vzdělávacího systému, pro strategie plánování jeho rozvoje, priorit aj. Je samostatnou vědní disciplínou, opírá se o rozsáhlou a propracovanou teoretickou a metodologickou základnu“ (Průcha, Walterová, Mareš, 2003, s. 155).

Slavík (1999, s. 36-37) uvažuje, že v praxi českých středních škol se pojem evaluace používá pro systémové hodnocení výuky. Přesto se tento pojem nevžil pro jakékoli všednodenní výsledky sumativního hodnocení. Uvádí, že pro tyto případy se v češtině používají spíše výrazy *slovní (o)hodnocení, známkování, poznámka* apod.

1.1.2 Hodnocení vs. evaluace

„Popravdě řečeno pojmy *hodnocení* a *evaluace* jsou v běžné řeči zaměňovány (*zhodnocení* je vlastně doslovným překladem slova *evaluace*) a také v odborných textech není jejich rozlišování ustáleno“ (Slavík, 1999, s. 35)

Průcha (1996, s. 11) uvádí, že v obecném významu jsou výrazy *hodnocení* a *evaluace* synonymní. Mezi těmito dvěma slovy ale přesto existují drobné rozdíly a to v české i zahraniční odborné pedagogické terminologii.

- Evaluace pokrývá širší, komplexní význam výše zmíněných distinktivních rysů. Termín evaluace tedy vyjadřuje teorii, metodologii a praxi veškerého hodnocení a je především používán ve vědeckém a výzkumném okruhu komunikace.
- Termín *hodnocení* se používá v širších kontextech běžné školní praxe, zejména v souvislosti s evaluací určitých subjektů vzdělávání. Hovoří se například o hodnocení žáků, hodnocení práce učitelů apod.

Rozdíly mezi hodnocením a evaluací jsou přehledně znázorněny v následující tabulce (Tab. 1). Z tabulky je zřejmé, že hlavním rozdílem je absence nutnosti přípravy při hodnocení. K evaluaci je naopak potřeba důsledný, vědecký přístup.

Tab. 1: Rozdíly mezi hodnocením a evaluací podle Nezvalové (2006)

HODNOCENÍ (neřízené hodnocení)	EVALUACE (řízené hodnocení)
Kritéria: <ul style="list-style-type: none"> ○ nejsou vymezena, ○ indikátory výkonu nejsou stanoveny explicitně, ○ nejsou sdílána mezi partnery. 	Kritéria: <ul style="list-style-type: none"> ○ jsou vymezena explicitně a odsouhlasena, ○ jsou stanoveny specifické oblasti priorit založené na vymezených cílech, ○ jsou formulovány indikátory výkonu.
Evaluační plán: <ul style="list-style-type: none"> ○ není přesně stanoven, ○ není jasné, co kdo bude dělat, ○ není konzistentní s cíli, ○ není připravován záměrně, je použit v případě potřeby. 	Evaluační plán: <ul style="list-style-type: none"> ○ je strukturovaný, ○ je daná jasná odpovědnost, ○ jsou definovány explicitní vztahy s cíli, ○ vyžaduje detailní plánování.
Metody: <ul style="list-style-type: none"> ○ nejsou předem stanoveny, ○ metody jsou nekonzistentní, ○ nepromyšlená analýza dat. 	Metody: <ul style="list-style-type: none"> ○ systematické, ○ přesně určené zdroje dat, ○ použit reprezentativní vzorek, ○ evaluační nástroje odpovídají metodám použitým pro sběr dat, ○ systematická analýza dat, ○ vypracování zprávy.

1.2 Druhy evaluace

Evaluaci můžeme rozlišovat podle mnoha kritérií a hledisek. Starý (2005, s. 6) uvádí členění dle tří hledisek. Jedná se o:

- hledisko hodnotitele,
- hledisko účelu,

- hledisko interpretace výsledků.

1.2.1 Členění z hlediska hodnotitele

Toto členění klade především důraz na osobu, která evaluaci provádí.

Z pohledu školy ji Vašátková (2010, s. 10) rozlišuje na:

- **Externí** (vnější), která je uskutečňována orgány inspekce, zřizovateli, jinými agenturami na základě národních srovnávacích zkoušek, výzkumnými pracovišti atd.
- **Interní** (vnitřní), kdy si ji škola zajišťuje sama.

Slavík (1999, s. 24) rozděluje hodnocení také na **vnitřní a vnější**. Toto dělení ale uvažuje z pohledu výuky. Tvrdí, že jedinými vnitřními posuzovateli (hodnotiteli) výuky a nejsilněji zainteresovanými účastníky jsou učitelé a žáci. Za vnější posuzovatele považuje všechny ostatní – tedy ředitele, hospitující učitele, inspektory či laické posuzovatele – v první řadě rodiče žáků.

1.2.2 Rozdělení dle hlediska účelu

„Účelem **sumativní** evaluace je zachytit konečný stav výsledků, např. vysvědčení žáka nebo počet absolventů přijatých ke studiu na vysokou školu, aby se mohly výsledky porovnávat s jinými subjekty, např. žáky nebo středními odbornými školami. Oproti tomu účelem **formativního** hodnocení je poskytnout subjektu (žákovi či škole) zpětnou vazbu, aby se mohl dále zlepšovat. V praxi se tyto dva účely zpravidla vyskytují současně. Zda se jedná spíše o sumativní nebo formativní evaluaci lze rozhodnout podle toho, jaký účel převažuje.“ Starý (2005, s. 6)

1.2.3 Rozdělení dle interpretace výsledků

Do tohoto rozdělení Starý (2005, s. 6) zařazuje:

- **Normativní evaluaci**, která se vyznačuje možností porovnat výsledky hodnocených subjektů mezi sebou.
- **Kriteriální hodnocení**, které je především vztaženo k předem stanoveným a zpravidla explicitně vyjádřeným kritériím.

1.2.4 Rozdělení z hlediska času

Další členěním evaluace je z hlediska času. Uvažují se zde období „před“, „během“ a „po“ realizaci programu či procesu, který je předmětem evaluace (Seberová, Malčík, 2009, s. 28-29):

- Evaluace **ex ante** neboli předběžnou evaluaci. Jejím předmětem je posouzení vnitřní konzistence programu, vazby na jeho účely, problémů, které má řešit. Je posuzován soulad mezi cíli a strategiemi.
- Evaluace **on going** neboli průběžná evaluace. Probíhá během realizace programu či procesu. Předmětem jsou především pokroky v naplňování stanovených cílů, kvalita a adekvátnost strategií.
- Evaluace **ex post** neboli následná evaluace. Používá se po skončení programu či procesu. Předmětem posouzení je hlavně reálný stav naplnění cílů. Posuzují se zde i další okolnosti, jako například dopad na účastníky či prostředí, kvalita zdrojů aj. Mohou zde být i formulována případná korektivní opatření a doporučení k modifikaci programu.

2 E-LEARNING

„Termín *e-learning* se u nás používá v této anglické podobě nebo v překladu jako ‚elektronické učení/vzdělávání‘. Označuje různé druhy učení podporovaného počítačem, zpravidla s využitím moderních technologických prostředků, především CD-ROM.“ (Průcha, Walterová a Mareš, 2003, s. 57)

Podle Květoně (2007) není pojetí e-learningu ustálené. Představuje ale tři základní varianty:

- **Pedagogické pojetí**, kde přístup k multimédiím za pomoci internetu výrazně obohacuje obsah výuky. Podle tohoto pojetí slouží internet také k výměně informací a ke spolupráci vzdělávací komunity.
- **Technologické pojetí** tvrdí, že e-learning je spektrum aplikací a procesů jako je Web-based training (WBT), Computer-based training (CBT), virtuální třídy nebo digitální spolupráce.
- **Sít'ové pojetí**, které Květoň (2007) zmiňuje, spočívá v užití sítí pro přenos dovedností a znalostí. Také ale uvádí, že toto pojetí je velmi úzké vymezení e-learningu, poněvadž nezahrnuje např. výuku pomocí CD-ROM či jiných datových nosičů.

Podle Kopeckého (2006) je e-learning v širším pojetí aplikací multimediálních technologií a internetu za účelem zvýšení kvality a dostupnosti vzdělávání. Jedná se o multimediální podporu vzdělávacího procesu. V užším pojetí se jedná o vzdělávání realizované prostřednictvím internetu.

V souvislosti se zaváděním e-learningu ve školách zmiňuje Kopecký (2007) pojem ‚blended learning‘. Jedná se o smíšenou formu vzdělávání, která představuje ideální kombinaci výhod frontální výuky, která se běžně používá ve školství, a e-learningu.

„Blended learning je prezenční výuka s e-learningovou podporou, kterou lze využít v běžných hodinách, ale také v domácí přípravě. Tedy počítačově gramotný učitel běžně využívá ve svých hodinách funkčně a efektivně techniku tak, aby mělo její použití žádoucí vzdělávací efekt.“ (Kopecký, 2007)

2.1 Historie

Zounek (2009, s. 24) uvádí, že první náznaky e-learningu se objevily už v 60. letech minulého století. Přestože v té době nebyla k dispozici takřka žádná multimédia ani neexistovaly žádné síťové prostředky, existovala snaha využít moderní technologie ve vyučování a učení.

Koncem 60. let se začalo experimentovat se stroji na učení, kterým se říkalo vyučovací automaty. U nás byl vyvinut automat s názvem Unitutor. Vykládaná látka byla rozdělena na jednotlivé stránky. Na konci každé stránky byla uvedena kontrolní otázka s výběrem odpovědí. Díky tomu se mohl program větvit. Vyučovací automaty byly příliš složité a málo účinné. Proto se neujaly. (Stříteská, 2003)

Samotný e-learning, jak jej známe dnes, mohl začít až s rozvojem internetu Podle Květoně (2007) vznikl e-learning až v roce 1993. Uvádí, že samotný pojem ‚e-learning‘ nebyl do roku 1999 zaveden. Do té doby se používal výraz ‚online learning‘.

2.2 Formy e-learningu

Výčet forem dle Kopeckého (2006):

- **CBT** (Computer-based training) – za použití osobních počítačů a přenosných médií, jako jsou například CD, DVD, flash disky apod.
- **WBT** (Web-based training) – jedná se o formu, která používá pro distribuci obsahu internet.
- **LMS** (Learning Management Systems) – jeho základní podstatou je organizovat a řídit výuku a kompetence (Pejša, 2004).
- **LCMS** (Learning Content Management System) – tímto termínem lze označit jakýkoli nástroj či systém, který slouží k tvorbě či sestavování výukového obsahu (Pejša, 2004).
- **Active worlds, 3D svět, virtuální realita.**
- **Blended learning**, který je kombinací e-learningu a běžné výuky.

2.3 Role učitele v e-learningu

„Role učitele v e-learningu (především pak v ucelených on-line kurzech) je specifická, a proto se používá i specifický výraz ‚tutor‘. Z hlediska koncepce e-learningu je podstatné,

zda student bude s e-learningovým materiálem pracovat ve třídě pod dohledem učitele (blended learning) zcela samostatně (plně distanční studium bez průběžné podpory učitele), nebo zda se bude jednat o samostudium na dálku řízené učitelem (tutorovaný kurz). Tutor v e-learningovém kurzu je především facilitátor a jeho základním úkolem je pedagogicko-psychologická podpora žáků a studentů. Tutor usnadňuje a řídí studium, zpřístupňuje doplňující zdroje, upozorňuje na mezipředmětové vztahy a souvislosti teoretických poznatků se zkušenostmi z praxe a podporuje žáky při překonávání studijních překážek.

Od slova tutor je odvozen název pro prezenční setkání v e-learningu - tzv. „tutoriál“. V e-learningu se obvykle konají tutoriály úvodní, synchronizační (v průběhu kurzu) a závěrečné.“ (Rohlíková, Vejvodová, 2010, s. 12)

3 POUŽITÉ TECHNOLOGIE

V této kapitole budou popsány techniky a technologie, které jsem použil k vytvoření aplikace.

3.1 HTML

HTML (Hypertext Markup Language) je textový značkovací jazyk, který se používá k tvorbě webových stránek. „Každá webová stránka se zakládá na minimálním množství kódu jazyka HTML; jinak by to nebyla webová stránka.“ (Castro, 2012, s. 15)

Jazyk HTML se používá především k definici významu obsahu stránek. Slouží k tomu značky neboli tagy a jejich atributy. Všechny značky jsou tvořeny dvojicí lomených závorek, mezi kterými je název značky a případně atributy. Například ``

Tagy se dělí na párové a nepárové. Párových tagů je většina a jsou tvořeny otevírací a ukončovací značkou. Ukončovací značku symbolizuje lomítko, které se nachází před názvem značky.

Nepárové značky jsou spíše výjimkou. Patří mezi ně například značka pro obrázek („``“) nebo značka pro nový řádek („`
`“).

3.1.1 Historie

V roce 1991 vytvořil Tim Berners-Lee první definici jazyka HTML jako součást projektu WWW v CERNu. Tento projekt měl sloužit vědcům komunikaci a sdílení výsledků výzkumu po celém světě. (Kosek, 1998, s. 20)

Když tento jazyk vznikal, měl podobu stručného dokumentu popisujícího jen několik elementů, které popisovaly různé části stránky, jako například záhlaví, seznamy nebo odstavce. (Castro, 2012, s. 16)

S postupem času se požadavky na HTML zvyšovaly, a proto vznikaly nové verze tohoto jazyka. Aby tento vývoj byl kontrolovaný, založil Tim Berners-Lee v roce 1994 World Wide Web Consortium (W3C).

3.2 Kaskádové styly

Jestliže se jazyk HTML používá k definici významu obsahu stránek, pak kaskádové styly (CSS) se používají k definici vzhledu tohoto obsahu.

„První verze jazyka CSS se objevily až několik let po vzniku jazyka HTML, přičemž oficiálně se tento jazyk prosadil v roce 1996.“ (Castro, 2012, s. 16)

Dříve se pro formátování dokumentu používaly tzv. formátovací tagy. Mezi takové se řadil třeba tag „“, který sloužil k určení, jaký font se má použít, jakou bude mít text barvu apod. Od této praxe se ale opustilo. V této době se takové vlastnosti definují výlučně v CSS.

Kaskádové styly můžeme do HTML vložit třemi základními způsoby:

- Pomocí atributu style.
- Pomocí tagu „<style>“, který by měl být umístěn v hlavičce dokumentu.
- Pomocí externího souboru.

Nejvyšší prioritu z výše zmíněných možností mají styly definované v atributu „style“. Při práci se styly platí několik pravidel:

- Později definované styly přepisují styly definované dříve.
- Styly s podrobnějším selektorem mají přednost.
- Příznak „!important“ umožňuje, aby nebyl daný styl později přepsán.
- Priorita vnitřního elementu. V případě konfliktu je použit styl toho elementu, který je nejbližší formátovanému obsahu.

Je potřeba také upozornit na několik nedostatků souvisejících s momentálním stavem vývoje specifikace CSS a současnými webovými prohlížeči. V současné době je vyvíjena třetí verze. Jelikož není tato verze ještě dokončena, podporují ji jednotlivé prohlížeče odlišně. Bohužel toto platí i pro stávající verzi CSS2.1.

3.2.1 Twitter bootstrap

Twitter bootstrap je sada kaskádových stylů a javascriptových funkcí, které lze velmi snadno použít při vytváření www stránek. Tato sada obsahuje velké množství předdefinovaných stylů, jako vzhled formulářových prvků, tlačítek, nadpisů. Navíc jsou tyto styly přizpůsobeny pro zobrazení stránek na různých zařízeních. Této vlastnosti se říká responzivní design a projevuje se tak, že prvky na stránce se přizpůsobují šířce stránky. Díky tomu se třeba na mobilním telefonu prvky poskládají pod sebe nebo na tabletu s menší úhlopříčkou se prvkům lehce upraví šířka.

Twitter bootstrap navíc obsahuje javascriptové knihovny, které umožňují mimo jiné zobrazovat modální okna, vyskakovací menu či bublinkovou nápovědu.

3.3 Javascript

Jak uvádí Aslesson (2006, s. 21–22) javascript vznikl za účelem dynamicky měnit tagy ve stránkách za účelem obohatit zážitek klienta. Posléze se zjistilo, že je možné považovat celou stránku za objekt, a tak se zrodil Document Object Model (DOM). Zpočátku byly javascript a DOM úzce propojeny, ale později se rozvinuly v odlišné koncepty. DOM tedy představuje stránku v její plně objektové reprezentaci, která může být modifikována skriptovacím jazykem, mezi které patří například i javascript.

O standardizaci DOM se zasloužilo konsorcium W3C. O standard javascriptu se stará asociace ECMA (European Computer Manufacturers Association). ECMA ratifikovala javascript jako specifikaci s označením ECMAScript. Skripty v souladu s touto specifikací by měly vypadat a chovat se stejně ve všech prohlížečích podporujících tyto standardy.

Javascript je vykonáván výhradně v prohlížeči, neboli na straně klienta. Je tedy potřeba mít prohlížeč, který javascript podporuje. Zároveň je také potřeba, aby uživatel ve svém prohlížeči javascript nezakázal. V současné době je na většině zařízení javascript funkční. Přesto by při návrhu aplikace mělo být myšleno na možnost, že javascript na klientském zařízení nebude k dispozici. Měl by se tedy spíše používat jako doplněk funkčnosti.

Veškerý javascript se do stránky vkládá pomocí párového tagu „<script>“. V případě, že chceme použít externí javascriptový soubor, přidáme do tagu atribut „src“, pomocí kterého specifikujeme cestu k danému souboru.

Syntaxe tohoto jazyka je podobná jazykům Java nebo C++. Zde ale podobnost s těmito jazyky končí. Javascript je jazyk bez typové kontroly. Do proměnné můžeme uložit řetězec, následně jej nahradit číslem, objektem či dokonce funkcí. Další odlišností je, že Javascript je interpretovaný jazyk.

3.3.1 jQuery

jQuery je javascriptová knihovna, která má za úkol usnadnit práci s nativním javascriptem. Mezi její největší výhody patří podpora selektorů a optimalizace pro všechny hlavní současné prohlížeče. Další velkou výhodou je její rozšířenost. Díky tomu je na internetu velká komunita programátorů, kteří mezi sebou sdílí zkušenosti s tímto frameworkem. Další výhodou je velké množství volně dostupných rozšíření (pluginů).

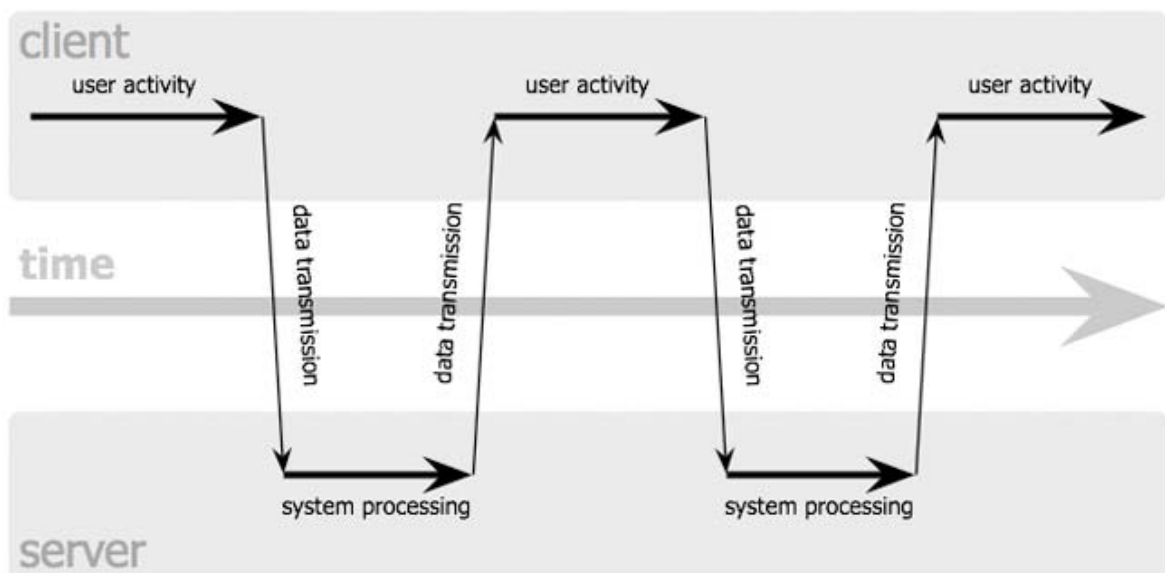
V nativním javascriptu je programátor odkázán na základní práci s DOM, kde se jednotlivé elementy jen velmi obtížně vyhledávají. Toto omezení jQuery řeší pomocí podpory

selektorů, které jsou podobné selektorům, jenž známe z CSS. Díky tomu můžeme pomocí jediného příkazu získat přístup ke všem tagům, které mají konkrétní třídu, nebo dokonce konkrétní hodnotu nějakého atributu. Samozřejmostí je vyhledávání podle identifikátorů, ale také rozšířené možnosti práce s DOM.

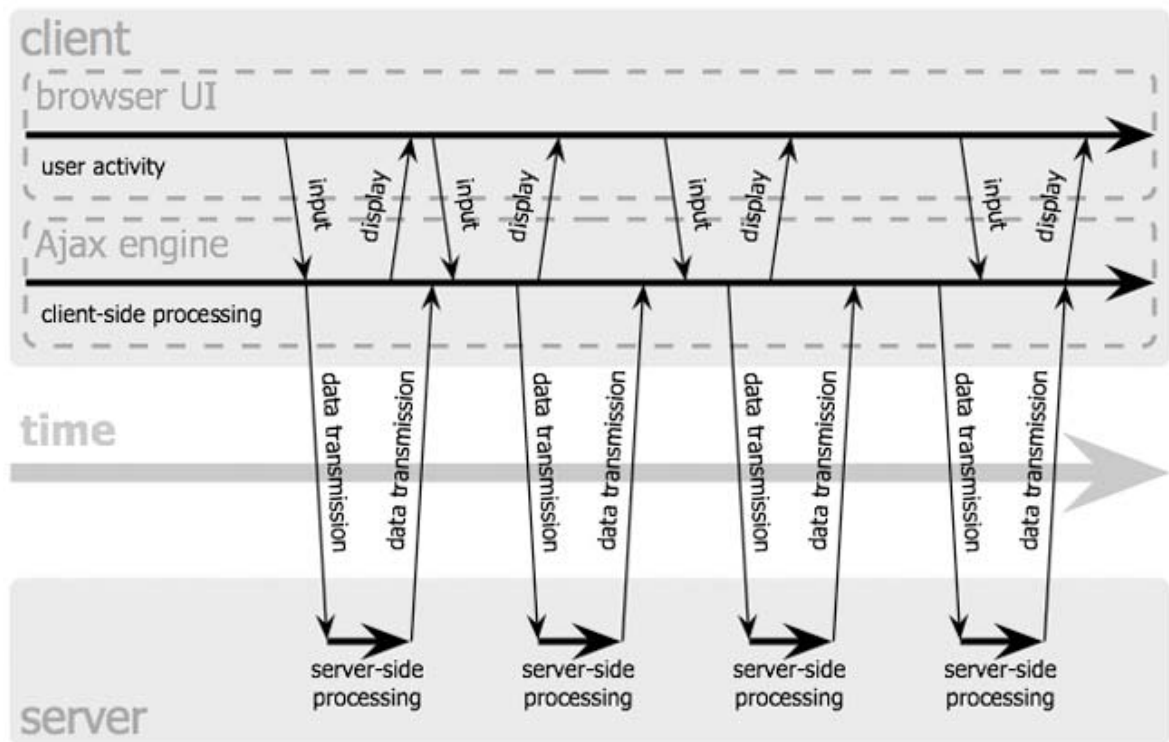
3.3.2 Asynchronous Javascript and XML

Asynchronous Javascript and XML (AJAX) je termín, který se poprvé objevil v roce 2005 v článku „Ajax: A new Approach to Web Application“ od Jesse James Garretta. Samotný termín AJAX zahrnuje tisíce různých datových komunikací, které mají ale společný předpoklad – požadavky probíhají směrem od klienta k serveru i v případě, že stránka je již kompletně načtena. (Resig, 2007, s. 20-21)

Díky tomu je možné stahovat ze serveru data, aniž by bylo nutné znovu načítat celou stránku. Rozdíl mezi klasickým (synchronním) modelem a asynchronním modelem webové aplikace ukazují následující obrázky (Obr. 1: Klasický (synchronní) model webové aplikace. (Garrett, 2005) a Obr. 2: Model ajaxové webové aplikace (Garrett, 2005))



Obr. 1: Klasický (synchronní) model webové aplikace. (Garrett, 2005)



Obr. 2: Model ajaxové webové aplikace (Garrett, 2005)

AJAX lze velmi snadno aplikovat pomocí jQuery. Tato knihovna obsahuje pro asynchronní komunikace následující funkce:

- jQuery.post() vykoná požadavek pomocí HTTP metody POST.
- jQuery.get() vykoná požadavek pomocí HTTP metody GET.
- jQuery.ajax() slouží k univerzálnímu volání asynchronních požadavků. Kombinuje obě výše zmíněné metody.
- .load() slouží k asynchronnímu načtení obsahu. Tímto obsahem se nahradí obsah elementu, nad kterým byla tato funkce zavolána.

Ačkoli je ve významu zkratky AJAX použit termín XML, není pro předávání dat tento formát přímo vyžadován. S úspěchem lze využít také formáty JSON, JSONP, HTML, script nebo i běžný text.

3.3.3 Highcharts

Highcharts je knihovna pro zobrazování grafů napsaná v čistém HTML5 a javascriptu. Tato knihovna je dostupná pro veřejné školy a univerzity dostupná pod licencí Creative Commons Attribution-NonCommercial 3.0 License. Jelikož licenční podmínky umožňují použití této knihovny v mé aplikaci, rozhodl jsem se ji použít. Grafy, které tato knihovna generuje, jsou velmi pěkné a přitom je snadné je vytvořit.

3.4 PHP

Zkratka PHP byla původně interpretována jako Personal Home Page. V průběhu času se význam zkratky změnil na Hypertext Preprocessor.

PHP je skriptovací jazyk, který běží na straně serveru. Z počátku se PHP používalo jako doplněk webových stránek, kdy se do HTML vkládaly tzv. ostrůvky PHP kódu. V současné době se začíná PHP používat čím dál více jiným způsobem. Nyní se vytvářejí komplexní aplikace, které celou HTML stránku přímo generují.

Ať se PHP používá jakýmkoli přístupem, vždy platí, že PHP kód se nachází mezi značkami „<?php“ a „?>“. Přičemž koncová značka není povinná. Toho se využívá především v aplikacích, které jsou tvořeny mnoha PHP soubory. Za ukončovací značkou býval často omylem přidán nějaký bílý znak (mezera, tabulátor, nový řádek apod.), který vedl následně k chybovým hlášením. Pokud vkládáme PHP přímo do HTML kódu, je ukončovací značka nutná.

PHP je velmi rozšířený jazyk. Jeho obliba spočívá ve velké podpoře na nejrůznějších hostingových serverech. Dá se říct, že použití PHP na hostingových serverech se stalo jednou ze základních poskytovaných služeb. Díky jeho rozšířenosti existuje velká komunita programátorů, kteří sdílí své zkušenosti na nejrůznějších fórech, vytvářejí četné tutoriály a návody.

Podobně, jako bylo zmíněno u javascriptu, i syntaxe PHP je blízká jazykům C nebo Java. PHP je také netypový jazyk, jehož kód se provádí interpretací. Velkou oblibu již od jeho vzniku zapříčinila jeho jednoduchost a zároveň velké množství vestavěných funkcí. Množství těchto funkcí se každou verzí značně zvětšovalo. Bohužel při takovém překotném tempu vznikla spousta nekoncepčních pochybení, která se projevují v názvosloví funkcí, nebo pořadí argumentů u některých funkcí.

Poslední stabilní verze je v době psaní této práce verze PHP5.4. Pátá verze se od čtvrté liší především rozšířením podpory objektového programování. Nyní již není problém v PHP používat základní vymoženosti objektového programování, mezi které patří například dědičnost, rozhraní, abstraktní třídy, klonování objektů apod.

3.4.1 Historie

Historie jazyka PHP se datuje od roku 1995, kdy Rasmus Lerdorf pracoval na vývoji nástroje PHP/FI (Personal Homepage Tools/Form Interpreter). Následně si tohoto nástroje

všimli Zeev Suraski a Andi Gutmans, kteří se rozhodli tento skriptovací jazyk přepracovat. Společně s Rasmusem Lerdorfem vytvořili novou verzi, která měla již dnes známý název PHP3 (PHP již znamená Hypertext Preprocesor). Krátce po uvolnění této verze se odhadovalo, že PHP je nainstalováno na 50 000 doménách.

Na konci roku 1998 se Zeev a Andi rozhodli jazyk znovu přepracovat. Verze PHP4 přišla s novým přístupem, kdy nejprve PHP kód přeložen do bajtového kódu, který je následně zpracován pomocí Zend engine (slovo Zend symbolizuje jména **Zeev** a **Andi**). Díky tomuto přístupu se výkon jazyka výrazně zvýšil, ale zároveň byla zachována zpětná kompatibilita s předchozí verzí. Vývoj pokračoval dále. Následně se začaly množit požadavky na obecnější objektově orientovaný přístup. Toto podnítilo vznik další v této chvíli poslední verze s označením PHP5. (Gutmans, 2005, s. 27-29)

3.4.2 Nette Framework

Nette je český framework napsaný v jazyce PHP. Plně využívá možností PHP5, momentálně je podporována i poslední verze PHP5.4. Jeho autorem je David Grudl.

Nette jsem do své aplikace zvolil z mnoha důvodů. Díky četné komunitě se rychle vyvíjí. Jak jsem již zmínil, vyvíjí jej především čeští programátoři, takže dokumentace i technická podpora je dostupná i v českém jazyce.

V tomto frameworku je implementována podpora návrhového vzoru MVC (Model-View-Controller). Cílem tohoto návrhového vzoru je oddělit uživatelské rozhraní od aplikační logiky a datové části.

Model je datový i funkční základ aplikace. Obsahuje tedy aplikační logiku a všechny akce uživatele jsou zpracovávány zde. Model si spravuje svůj vnitřní stav a ven nabízí pevně dané rozhraní. Model by neměl vědět o existenci view ani kontroleru.

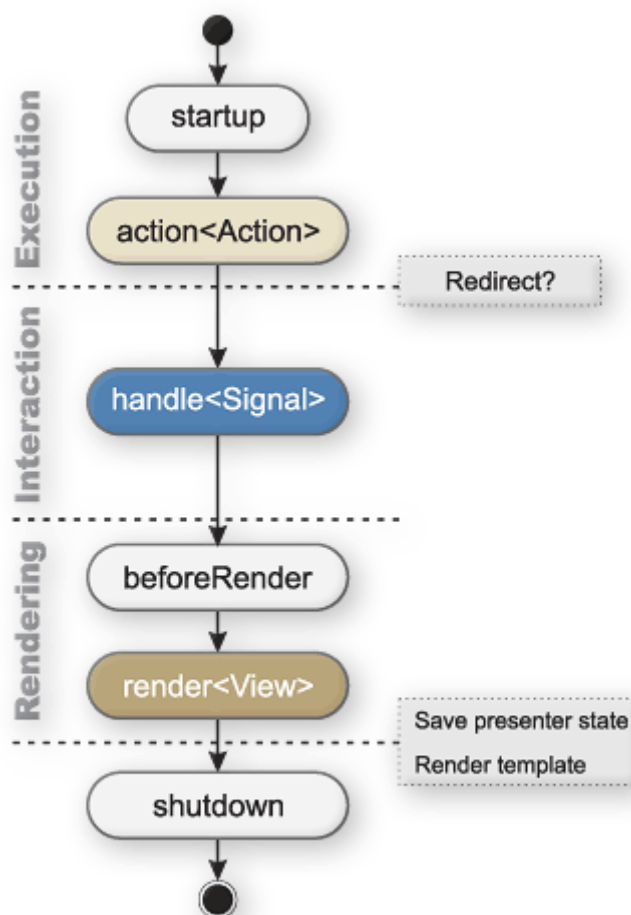
View je vrstva aplikace, která má na starosti zobrazení výsledku požadavku. V Nette se k tomu používá například šablonovací systém latte.

Controller je vrstva, která zpracovává požadavky uživatele. Na základě těchto požadavků vyvolá akci v modelu (aplikační logice) a následně požádá view o vykreslení získaných dat. V Nette frameworku se kontrolerům říká presentery.

3.4.2.1 Presenter

Presenter je řadič, který má za úkol zpracovat požadavek uživatele. Na obrázku (Obr. 3: Životní cyklus presenteru (Nette Framework: Dokumentace)) je znázorněno, jak je každý presenter postupně zpracováván. Po inicializaci objektu presenteru se spustí metoda `startup()`. Následně je vykonána metoda `action`, ve které se obvykle iniciují formuláře, nebo se zde rozhoduje o případném přesměrování aplikace. Následuje zpracování signálů (metoda `handle`). Signály slouží převážně ke změně dat, aniž by se změnilo view. S úspěchem se dá použít třeba při ajaxových požadavcích, kdy ukládáme do databáze změnu hodnot, nebo když mažeme nějaký záznam.

Následuje volání metody `render`, která by měla sloužit k získání dat z databáze a jejich předání do view. Získávat data z databáze bychom měli zde především proto, že v dříve popsaných metodách se tato data mohou měnit.



Obr. 3: Životní cyklus presenteru (Nette Framework: Dokumentace)

3.4.2.2 Komponenty

Komponenty jsou další silnou stránkou Nette frameworku. Jejich hlavním účelem je znovupoužitelnost kódu. Na stránkách nette frameworku (<http://addons.nette.org>) je veden seznam volně dostupných komponent, které slouží k rozšíření možností Nette.

Ve své aplikaci jsem použil následující komponenty:

- ConfirmationDialog – slouží k potvrzování uživatelských akcí.
- DatePicker – rozšiřuje formuláře o prvek umožňující výběr datumu.
- Grido – umožňuje vložit do stránek datagrid.
- CheckboxList – rozšiřuje formuláře o seznam zaškrtačacích políček.
- BootstrapFormRenderer – který formátuje formuláře ve stylu twitter bootstrap.
- InputList – formulářová komponenta pro práci se zaškrtačacími políčky.
- Plupload – umožňuje nahrávat soubory pomocí javascriptové knihovny Plupload.
- ReCaptchaControl – ochrana proti spamovacím robotům od google.
- Replikátor – umožňuje replikovat části formulářů.
- VisualPaginator – slouží k rozšíření možností stránkování.

3.5 MySQL

MySQL je volně šiřitelný multiplatformní relační databázový server. Jeho velká obliba je způsobena tím, že je nejčastěji používaným databázovým serverem na webech, které jsou založeny na technologii PHP. Ve srovnání s ostatními databázovými servery se řadí MySQL mezi ty jednodušší, avšak pro běžné operace je dostatečně vyhovující.

MySQL bylo vytvořeno ve švédskou firmou MySQL AB. V současné době ji vyvíjí společnost Oracle Corporation.

Komunikace s databází, jak vyplývá již ze samotného názvu, probíhá pomocí jazyka SQL (Structured Query Language). MySQL obsahuje několik typů úložišť. Tato úložiště se od sebe mírně odlišují.

MyISAM je nejpoužívanější typ tabulek v MySQL. Tyto tabulky nepodporují transakce, ale na druhou stranu jsou velmi rychlé. Tento typ tabulek je optimalizován pro dotazy SELECT.

InnoDB na rozdíl od MyISAM využívají transakce. Jsou optimalizovány pro větší rychlost dotazů INSERT a UPDATE a pro práci s velkým objemem dat. Ve své aplikaci jsem

se rozhodl použít právě úložiště InnoDB. Hlavním důvodem pro volbu tohoto úložiště je použití Nette\Database, kterou popíšu později.

3.5.1 PDO

PDO (PHP Data Objects) je podle Vrány (2005) nejvýznamnější novinkou v PHP 5.1. Jedná se o rozšíření, které umožňuje jednotným způsobem pracovat s rozličnými databázemi. Na rozdíl od jiných knihoven se nesnaží vytvořit kompletní abstrakci skrývající rozdíly mezi jednotlivými databázovými servery, ale snaží se poskytnout nízkou úroveň přístupu pomocí jednotného rozhraní.

3.5.2 Nette database a NotORM

V Nette Frameworku je implementována databázová vrstva, jejímž základem je PDO. Implementace v Nette umožňuje navíc uvádět v příkazech „exec“ a „query“ výčet parametrů.

Další nástroj, který rozšiřuje práci s PDO a který je v nette implementován, je knihovna NotORM od Jakuba Vrány.

„Knihovna NotORM se dá použít pro pohodlné procházení propojených dat uložených v databázi. Její největší předností je intuitivní API připomínající extenzi SimpleXML a vysoká efektivita. Efektivita je založena především na tom, že se všechna propojená data z databáze získávají najednou a přitom pouze k těm záznamům, ke kterým jsou potřeba.“ (Vrána, 2010, s. 229–230)

Díky této knihovně se mohou dotazy do databáze skládat postupně a za pomoci objektů. Není přitom nutné znát přímo syntaxi SQL.

II. PRAKTICKÁ ČÁST

4 ANALÝZA POŽADAVKŮ

V této části práce se budu zabývat konkrétními požadavky na aplikaci, která má být hlavním přínosem této diplomové práce.

4.1 Obecné požadavky

Navrhovaná aplikace má v první řadě za úkol umožnit učitelům prezentovat informace v co nejširší multimediální podobě. Očekává se její nasazení v kurzech grafiky, programování apod. Proto musí být učitelé schopni vkládat do obsahu obrázky, videa, ale i ukázky zdrojových kódů, které je kvůli jejich čitelnosti potřeba formátovat.

Druhým účelem aplikace je šetřit učitelům čas. Učitel musí být schopný dělat některé úpravy přímo během hodiny. Jedná se například o opravy v textu, založení nového domácího úkolu či sledování odevzdávání prací přímo v hodině. Je důležité, aby tyto informace nemusel nikterak složitě hledat, aby mu byly přímo nabídnuty. Toto je důležité především k relativně krátké době vyučování.

Dalším požadavkem je možnost položit studentům v kurzu krátký dotazník. Takové dotazníky jsou pro učitele cenným prostředkem pro zpětnou vazbu nebo nástrojem pro evaluaci. Mnohdy se stane, že učitelé napadnou další otázky přímo během výuky a proto je důležité, aby editace takových dotazníků byla maximálně jednoduchá. Dotazníky mohou sloužit nejen k evaluaci výuky, ale také jako jistá forma rozhodování v rámci skupiny. Učitel může studentům položit dotazník, jenž má spíše podobu anketní otázky či několika otázek, a zjistit tak případný zájem studentů o daná témata. Jelikož studenty takový výsledek může bezprostředně zajímat, mělo by učitelům být umožněno tyto výsledky studentům publikovat. A to v přehledné a graficky příjemné formě.

Mezi nejzákladnější požadavky většiny webových aplikací se řadí omezení funkčnosti jednotlivým uživatelům v závislosti na uživatelských rolích. V aplikaci jsou použity čtyři role. Jedná se o anonymního návštěvníka, studenta, učitele a administrátora. Již z názvů rolí je patrná jejich hierarchie a případná oprávnění. Účet administrátora a učitele se v mnohém neodlišuje. Administrátor má navíc pouze rozšířené pravomoci v oblasti správy uživatelů a nastavení aplikace.

4.2 Kurzy

Veškerá činnost podpory výuky bude v aplikaci prováděna prostřednictvím kurzů. Pod výrazem „kurz“ rozumějme pojem, který může odpovídat i pojmu „předmět“. Kurzy obsahují definice jednotlivých lekcí a úkolů.

K samotnému kurzu není možné přiřadit studenty či učitele. Kurzy nemají ani rozvrh. K těmto účelům budou sloužit od kurzů odvozené skupiny.

Díky odlišení kurzů od skupin můžeme s kurzy pracovat tak, aby toto suplovalo skutečnost. Například ve škole je předmět neboli kurz Informační a výpočetní technologie (IVT). Z důvodů kapacity počítačové učebny není možné, aby všichni studenti navštěvovali tento předmět ve stejné skupině. Proto pro tento kurz jsou vytvořeny dvě skupiny (IVT1 a IVT2), které mají společné učivo, ale rozdílný rozvrh hodin a třeba i různé učitele. Právě tento stav by měl být promítnut i do aplikace.

4.3 Skupiny

Skupiny slouží již pro samotnou práci se systémem při výuce. Ke skupině se budou přiřazovat studenti, učitelé. V detailu předmětu budou také přístupné jednotlivé lekce a úkoly. Právě tyto lekce si budou moci studenti během výuky zobrazovat, prohlížet a jejich prostřednictvím se i učit. Učitel bude mít možnost lekce editovat přímo ze své skupiny, přičemž se každá změna v lekcích zároveň projeví ve všech ostatních skupinách odvozených od daného kurzu.

4.3.1 Studenti ve skupině

Učitel by měl být schopný jednoduše přiřadit studenty do své skupiny. K tomu bude mít k dispozici jednoduchý formulář, pomocí kterého bude moci studenty vyhledat a následně i přiřadit do své skupiny. Jelikož na škole mají veškerou evidenci vedenou pomocí systému Bakaláři, bude mít učitel možnost provést import studentů do své skupiny pomocí vygenerovaného seznamu ze systému Bakaláři. Importovací skript bude schopný přiřadit studenty na základě jména, příjmení a třídy.

4.3.2 Připomínky

V rámci své skupiny bude mít učitel k dispozici systém pro zobrazování připomínek. Tato funkcionality je zamýšlena kvůli tomu, aby se nestalo, že učitel něco slíbí a pak na to

zapomene. Tyto připomínky budou zobrazovány pouze učiteli a všem žákům v jeho skupině.

4.3.3 Vzkazy

Při diskuzi požadavků na aplikaci padla žádost na možnost komunikovat se studentem pomocí krátkých zpráv, které se mu v systému zobrazí. Tento požadavek vznikl především proto, že učitel se může přímo během výuky daného studenta diskrétně dotázat. Tento student mu následně může odpovědět, aniž by o tom zbytek třídy musel vědět. Tento přístup může odbourat jistou komunikační bariéru způsobenou ostychem některých studentů.

4.3.4 Úkoly

Jeden z nejdůležitějších požadavků na novou aplikaci je systém úkolů. Jak jsem napsal výše, úkoly budou moci být definovány k jednotlivým lekcím. Úkoly budou přístupné pouze pro přihlášené uživatele, aby bylo pro učitele zjevné, jaký úkol byl odevzdán kterým studentem.

Ke každému úkolu bude učitel mít možnost přidat termín odevzdání, název úkolu a volitelně i popis, který může studentům sloužit jako návod. Termín odevzdání nemá studenty omezovat pro případné pozdní odevzdání. Má sloužit hlavně jako informace pro učitele, že nějaký student svou práci odevzdal pozdě.

Pro potřeby aplikace jsme se shodli na třech základních typech úkolů:

- odevzdání souboru
- odevzdání obrázku
- odevzdání textu.

Na první pohled by se mohlo zdát, že odevzdání souboru a obrázku je prakticky totéž. Typ úkolu pro obrázky bude mít umožněnu další práci s odevzdanými úkoly. Učitel bude mít možnost z odevzdaných obrázků vytvořit prezentaci, kterou bude moci studentům ještě v hodině předvést. Také bude mít možnost vytvořit pro studenty z vybraných obrázků krátký dotazník, pomocí kterého si studenti své práce budou moci prohlédnout a následně se mezi sebou i ohodnotí.

Následně bude mít učitel možnost z odpovědí vytvořit „hitparádu“ nejlépe ohodnocených prací, které bude moci studentům prezentovat.

4.4 Správa dotazníků

Jednou z nejdůležitějších částí aplikace bude systém pro evaluaci výuky neboli dotazníkový systém. Každý učitel bude moci vytvořit dotazník, který posléze pomocí odkazu nebo interních zpráv rozešle studentům dotazník k vyhodnocení. Proto jedním z požadavků učitelů je umožnit vyplnění dotazníku i anonymním uživatelům. Mohlo by se stát, že čas strávený registrací bude posléze chybět při vyplňování dotazníku. Nepředpokládá se tedy, že všichni studenti budou v systému registrovaní.

4.4.1 Vytvoření

Vytváření dotazníků bude umožněno pouze učitelům, kteří jsou v systému zaregistrovaní s příslušnými právy. Při vytváření dotazníku bude učitel moci zadat základní informace k dotazníku jako je jeho název, úvod k dotazníku, základní informace o dotazníku a závěrečné poděkování. Také bude moci nastavit, zda je dotazník veřejný (odkaz na něj se zobrazí na úvodní stránce) a zda je již připraven k vyplňování. Učitel také bude moci nastavit období, ve kterém bude dotazník přístupný, případně heslo, které bude nutné zadat, aby mohl být dotazník spuštěn.

4.4.2 Odpovídání

Odpovídat bude moci kdokoli, kdo zná odkaz na konkrétní dotazník. Po načtení odkazu se uživateli zobrazí název dotazníku a základní informace. Po samotném spuštění dotazníku se zobrazí úvodní text a poté již všechny otázky. Otázky, které učitel označí jako povinné, budou zvýrazněny. Během vyplňování dotazníku může uživatel tento dotazník průběžně ukládat. V případě odeslání bude zkontrolováno vyplnění všech povinných otázek. Pokud budou některé z nich nevyplněny, budou tyto otázky podbarveny červeně.

4.4.3 Vyhodnocení

Učitel bude mít k dispozici systém pro vyhodnocení dotazníku. U otázek s uzavřenými odpověďmi bude k dispozici možnost zobrazení grafu. U otevřených otázek bude k dispozici možnost exportu jednotlivých odpovědí. Samozřejmě bude export všech odpovědí do souboru, který je možné otevřít v programu MS Excel.

Může se stát, že dotazník vyplní někdo, kdo jej vyplňovat neměl. Takového respondenta může učitel z výsledků dotazníku vyřadit. K dispozici bude také možnost zobrazit si odpovědi jednotlivých respondentů.

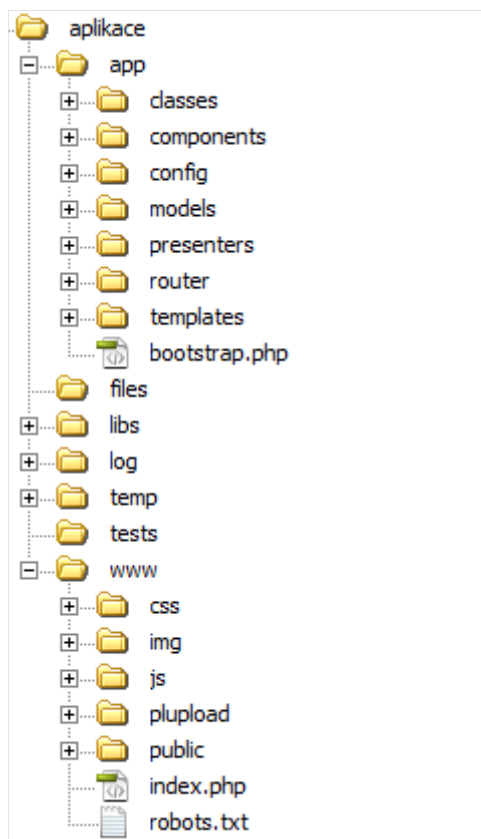
5 NÁVRH STRUKTURY APLIKACE

V této kapitole stručně vysvětlím návrh struktury aplikace ze dvou pohledů. Jsou jimi návrh rozložení aplikace na úrovni souborového systému a návrh struktury databáze.

5.1 Adresářová struktura

Struktura souborového systému odpovídá doporučenému rozložení adresářů v sandboxu (Obr. 4. Adresářová struktura), jenž je dodáván v distribuci frameworku Nette. Základními rysy tohoto rozložení jsou:

- Aplikace a knihovny jsou mimo adresář, který je veřejně přístupný z webu.
- Knihovny, které aplikace bude využívat, jsou v jiném adresáři, než ve kterém je samotná aplikace.
- Struktura adresáře s aplikací odpovídá návrhovému vzoru MVC.



Obr. 4. Adresářová struktura

5.1.1 app

Adresář app je zřejmě tím nejdůležitějším umístěním v aplikaci. Nachází se zde veškerá aplikační logika, šablony, konfigurační soubory. Tento adresář by měl být umístěn mimo

viditelnou část webu. Za předpokladu, že adresář `www` je touto viditelnou složkou, je tato podmínka splněna. Hlavním důvodem, proč by tento adresář neměl být vidět z webu, jsou právě konfigurační soubory.

V tomto adresáři se nachází několik složek, jejichž význam je vhodné vysvětlit:

- `models`,
- `presenters`,
- `templates`,
- `components`,
- `config`.

Ve složce `models` se nachází třídy pro práci s databází. Jednotlivé třídy zde obsahují metody pro získání dat z databáze, pro ukládání, změnu či mazání záznamů.

Složka `presenters` obsahuje definici všech zobrazitelných stránek aplikace. Nachází se zde prakticky celá aplikační logika.

V adresáři `templates` se nachází šablony, které jsou nutné pro zobrazení aplikace. Struktura tohoto adresáře odpovídá seznamu presenterů. Pro každý presenter zde existuje složka s definovanými šablonami pro jednotlivé akce presenteru.

Ve složce `config` se nachází konfigurační soubory. Jsou zde mimo jiné definované nastavení připojení k databázi, uživatelská oprávnění, seznam modelů apod.

5.1.2 `www`

Toto je jediný adresář, který je přístupný z internetu. Nachází se zde tedy klientská část aplikace. Tím jsou myšleny javascriptové knihovny, kaskádové styly, obrázky apod.

Jediný `php` soubor, který by se zde měl nacházet, je `index.php`, jenž má za úkol spustit aplikaci.

5.1.3 `libs`

Do tohoto adresáře jsem umístil samotný Nette Framework a rozšíření, která jsem se v aplikaci rozhodl použít.

5.1.4 files

Do této složky nahrávají studenti své úkoly. Pro tento účel jsem zvolil složku mimo viditelnou část webu z toho důvodu, aby studenti neměli volný přístup k pracím ostatních studentů.

5.1.5 log

Do tohoto adresáře se ukládají všechny logy aplikace. V případě potřeby je zde možné najít několik typů záznamů:

- access.log,
- error.log,
- html soubory popisující chyby (soubory s prefixem „exception“ + datum).

Při získávání informací o problémech s aplikací je nejužitečnější právě html soubor s detailním popisem chyby či výjimky. V těchto souborech se dají najít téměř všechny informace o stavu aplikace v chvíli, kdy k problému došlo.

5.1.6 temp

Adresář pro dočasné soubory a cache. Všechny soubory v této složce slouží převážně k urychlení běhu aplikace. Rozhodl jsem se také ukládat do této složky vygenerované soubory, mezi které patří například zip archivy s odevzdanými úkoly nebo csv soubory s výsledky dotazníků.

5.2 Návrh databáze

V této kapitole budu popisovat návrh databáze. Jelikož databáze obsahuje mnoho tabulek, rozdělil jsem je podle základních celků. Tyto celky následně popíši.

V prvním diagramu (Obr. 5: Struktura tabulek pro kurzy) je vyobrazeno schéma, které znázorňuje část databáze, popisující systém kurzů a skupin a lekcí.

Ve schématu se nachází několik číselníků:

- **Classes** – zde se nachází seznam všech tříd. O jeho naplnění se stará administrátor, který má k dispozici nástroj pro import ze systému Bakaláři.
- **Hours** – zde se nachází seznam vyučovacích hodin, který se používá k zobrazení rozvrhu hodin na úvodní stránce

- **Rooms** – tato tabulka obsahuje seznam učeben. Tento seznam může aktualizovat administrátor. Tato tabulka je použita pro přiřazení skupiny do konkrétní učebny.

Schedules je tabulka, pomocí které je v systému zaznamenávána informace o rozvrhu skupin. Je zde uložena informace kdy a kde probíhá výuka jednotlivých skupin.

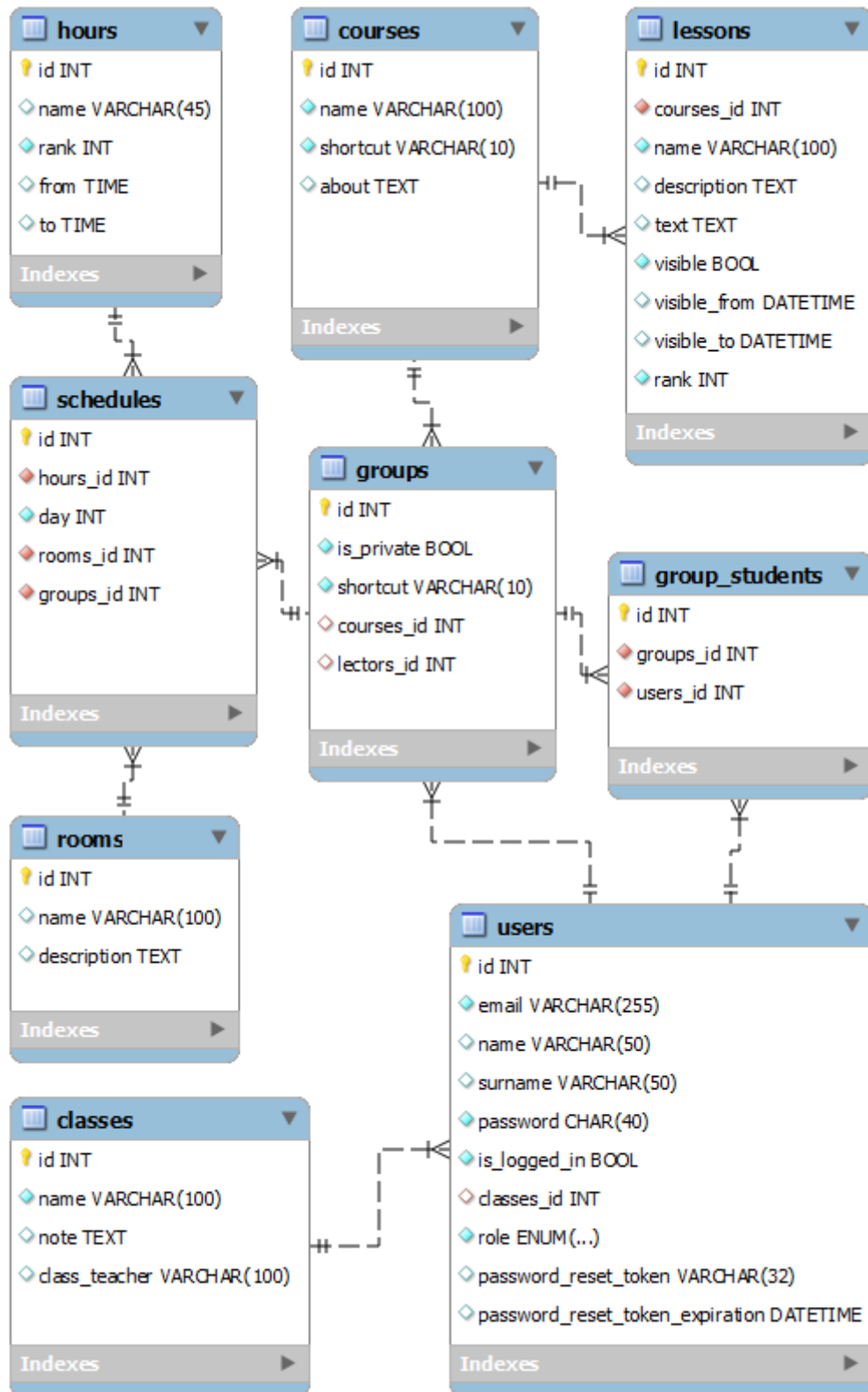
Courses – v této tabulce se nacházejí obecné informace týkající se jednotlivých kurzů. V základu to jsou pouze název kurzu, zkratka a popis.

Na tabulku s kurzy je napojena tabulka **groups**, která reprezentuje skupiny. Jak jde vidět z její definice, skupina může mít právě jeden kurz a jednoho učitele. Studentů může mít každá skupina více a to díky tabulce **groups_students**.

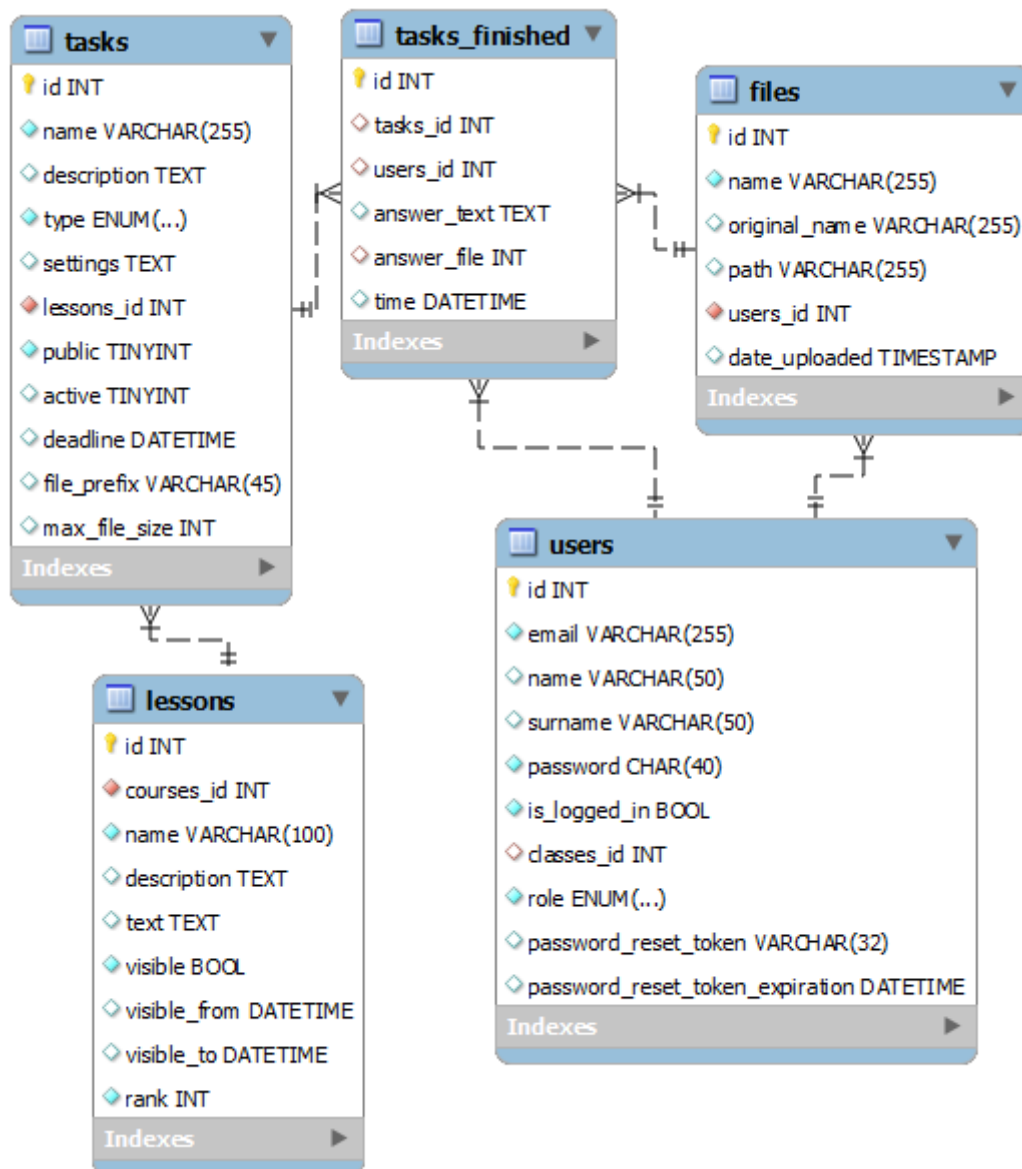
S tabulkou kurzů je také svázána tabulka **lessons**, která obsahuje informace o jednotlivých lekcích. Na tabulku s lekcemi jsou dále navázány úkoly. Vazbu mezi tabulkami pro lekce a úkoly znázorňuje další diagram (Obr. 6: Struktura tabulek pro úkoly)

Do tabulky **tasks** se ukládají informace o úkolech. Nachází se zde mimo jiné informace o názvu úkolu, popis, typ úkolu či termín odevzdání.

V tabulce **tasks_finished** jsou uchovávány odevzdané úkoly. Za povšimnutí stojí, že soubory odevzdané v rámci úkolů, jsou ukládány do zvláštní tabulky – **files**. Všechny tyto informace jsou propojeny s tabulkou **users**, ve které se nachází informace o všech uživateli aplikace. Díky tomu víme, který soubor či úkol student odevzdal.



Obr. 5: Struktura tabulek pro kurzy

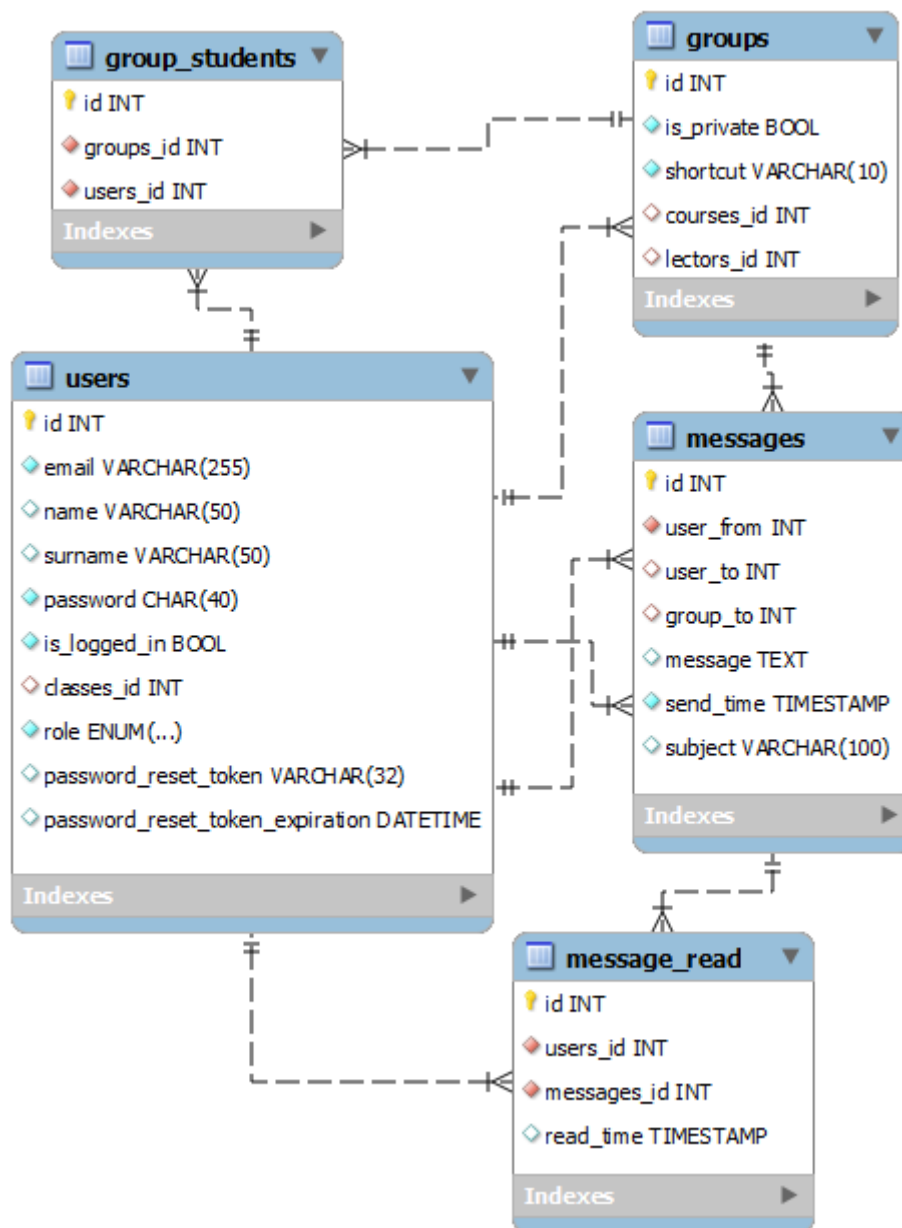


Obr. 6: Struktura tabulek pro úkoly

Dalším diagramem, který je potřeba popsat, je schéma pro zprávy (Obr. 7: Struktura tabulek pro zprávy). V tomto diagramu je znázorněno, že zprávy (tabulka **messages**) jsou vázány buď na jednotlivé příjemce (sloupec **user_to**) nebo na skupiny. Zprávy určené pro skupiny budou zobrazeny správným studentům právě díky spojovací tabulce **group_students**.

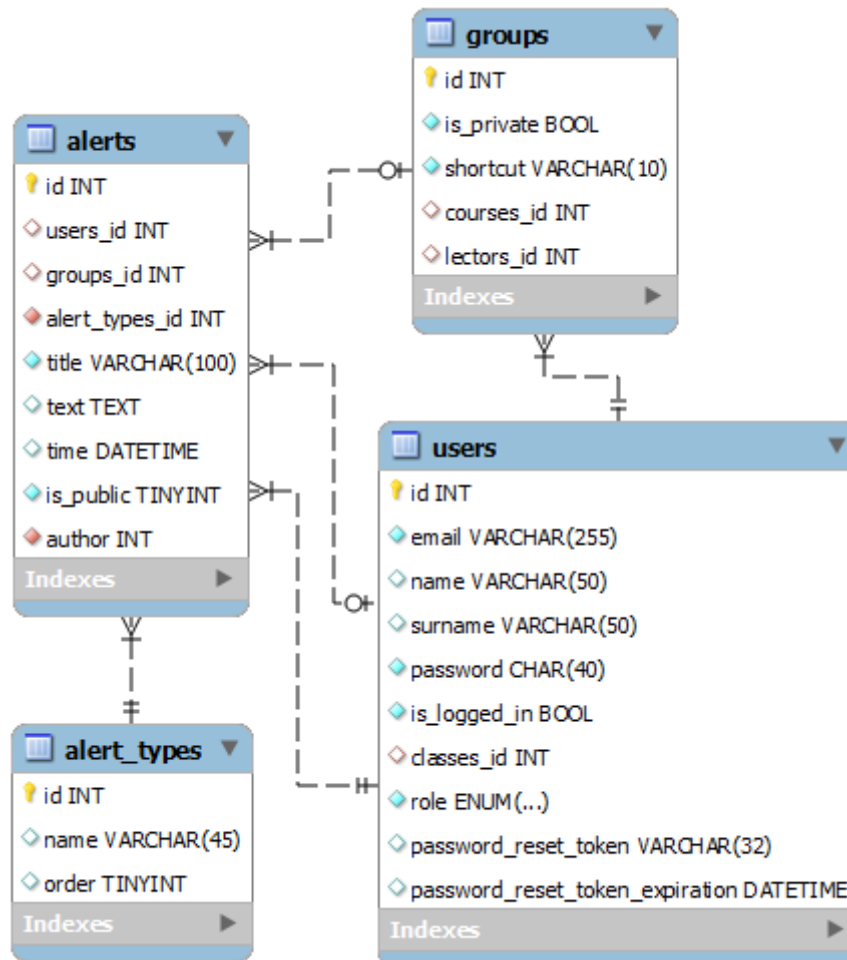
Samotná zpráva obsahuje kromě příjemců také předmět, text, odesílatele a také čas odeslání.

Informace, zda byla zpráva přečtena, je uchovávána v tabulce **message_read**. Zde se také ukládá čas přečtení.



Obr. 7: Struktura tabulek pro zprávy

Dalším diagramem je struktura tabulek pro systém upozornění (Obr. 8: Struktura tabulek pro systém upozornění). Opět i z tohoto diagramu je zjevné, že tabulka **alerts** je vázána. Pro každé upozornění je ukládán název, text upozornění a čas, ke kterému se dané upozornění váže. Druhy typů upozornění jsou uloženy v číselníku **alert_types**.



Obr. 8: Struktura tabulek pro systém upozornění

Posledním schématem je struktura tabulek pro dotazníky (Obr. 9: Struktura tabulek pro dotazníky) V tomto diagramu se nachází tři číselníky:

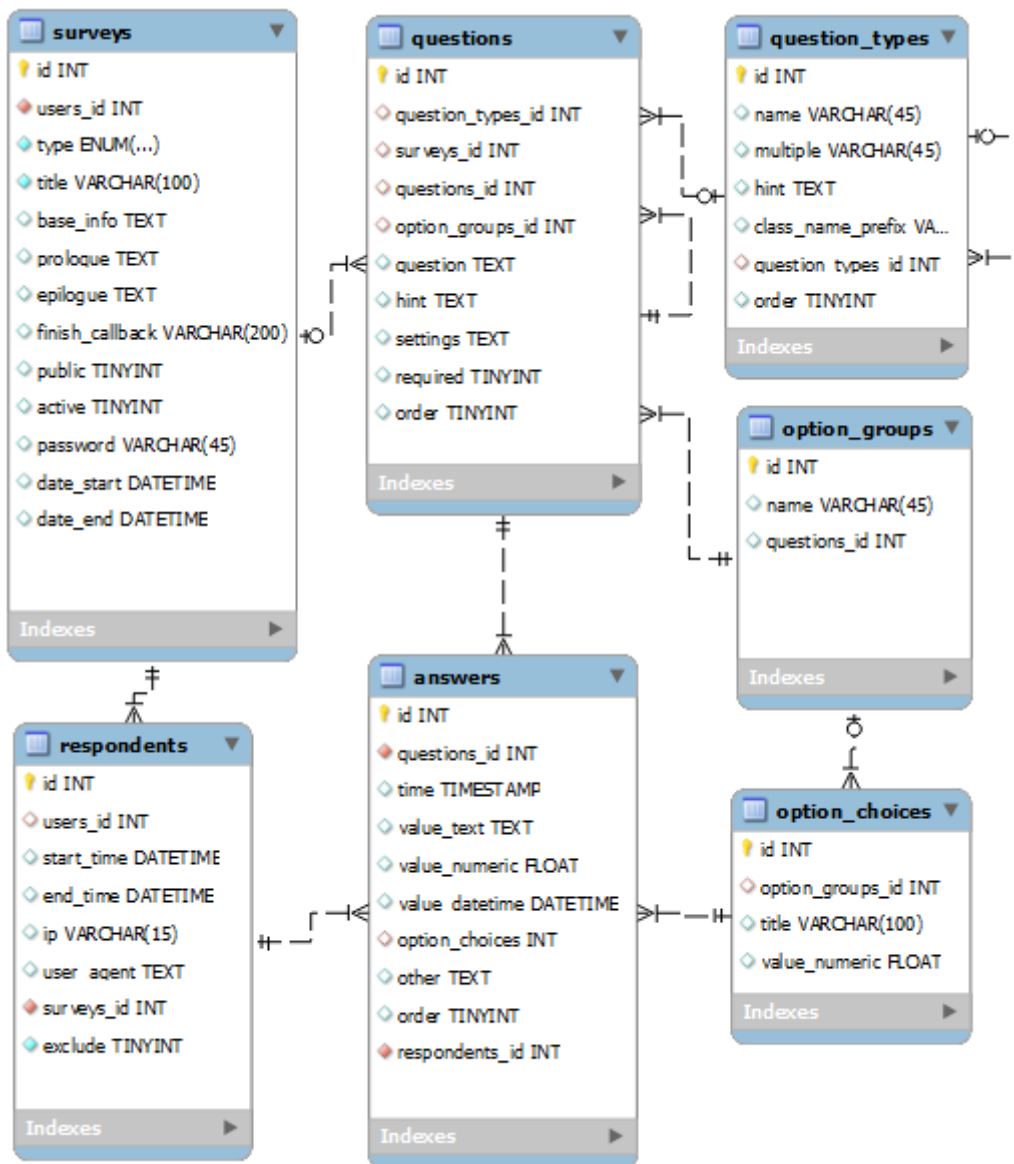
- **Question_types** – obsahuje seznam všech typů otázek. Je zde také uložen prefix názvu třídy, která v aplikaci daný typ otázky popisuje.
- **Option_groups** – v této tabulce jsou uloženy jednotlivé skupiny pro uzavřené odpovědi.
- **Option_choices** – v této tabulce jsou uloženy všechny možnosti pro skupiny uzavřených odpovědí. Primární klíč z této tabulky je pak použit pro uložení odpovědi na danou otázku.

Tabulka **surveys** obsahuje informace o jednotlivých dotaznících. Jak je z jejich sloupců patrné, ukládá se zde kromě základních informací o dotazníku také přístupové heslo či datum pro spuštění či ukončení dotazníku.

Na tabulku s dotazníky jsou navázány další dvě tabulky. Jsou to tabulky **questions** a **respondents**. Tabulka questions obsahuje informace o jednotlivých otázkách. Patří sem typ otázky, nápověda, zda je otázka povinná. Pokud je zvolen typ otázky, který vyžaduje výběr uzavřené skupiny je tato také uložena zde.

Jak vyplývá z názvu tabulky **respondents**, jsou v ní ukládány informace ohledně respondentů. U každého respondenta se ukládá čas spuštění a dokončení dotazníku, pokud je to přihlášený uživatel, tak je to zde také uloženo.

Poslední tabulkou je tabulka **answers**, kam se ukládají jednotlivé odpovědi.



Obr. 9: Struktura tabulek pro dotazníky

6 ŘEŠENÍ APLIKACE

Pro vývoj jsem se rozhodl použít následující technologie:

- Na straně serveru jazyk PHP, databázový server MySQL, vše s podporou PHP frameworku Nette.
- Na klientské straně je to především javascriptový Framework jQuery a základní design je založen na twitter bootstrap.

V dalších podkapitolách se budu zabývat jednotlivými částmi aplikace tak, jak byly ve výsledku implementovány.

6.1 Úvodní stránka

Úvodní stránka má především sloužit k nasměrování uživatele. Proto existují tři různé varianty této stránky a to pro nepřihlášené neboli anonymní uživatele, pro studenty a pro učitele.

Na úvodní stránce uvidí nepřihlášený uživatel základní pokyny, přihlašovací formulář a seznam veřejných dotazníků.

Po přihlášení vidí studenti na úvodní stránce především svůj rozvrh hodin (Obr. 10: Rozvrh hodin na úvodní stránce). V této verzi aplikace se zde zobrazují pouze ty předměty, na jejichž kurzy je student v systému zapsán. Obdobný rozvrh hodin vidí i učitelé.

Rozvrh

	1	2	3	4	5	6	7	8	9	10	11
Po						G (G2)					
Út											
St		G (G1)									
Čt											
Pá											

Obr. 10: Rozvrh hodin na úvodní stránce

Přihlášení uživatelé zde také vidí kalendář s upomínkami, které mohou učitelé ve svých skupinách vkládat. Tento systém bude popsán později.

6.2 Skupiny

Hlavní činnost během výuky, při které učitel využívá tuto aplikaci, se odehrává právě v části aplikace, kterou jsem nazval skupiny. Právě zde se nachází lekce a systém pro práci s úkoly.

V rámci editace skupiny může učitel upravovat:

- Základní informace o skupině.
- Rozvrh skupiny.
- Zapsat studenty pomocí formuláře, či je hromadně importovat ze systému Bakaláři.

6.3 Lekce

Lekce by měla obsahovat veškeré elektronické podklady, které učitel plánuje při hodině použít. Student si podle pokynů učitele danou lekci vyhledá a následně zde bude mít veškeré materiály, které při výuce upotřebí.

6.4 Úkoly

V případě, že se učitel rozhodne k lekci přidat úkol, může tak udělat v editaci dané lekce. Zde se nachází jednoduchý formulář, pomocí kterého může učitel přidat nový úkol. Jediný povinný parametr úkolu je jeho název. Všechny ostatní hodnoty jsou volitelné a dostatečně vysvětlené přímo ve formuláři.

Výsledný úkol je zobrazen na konci každé lekce. Učitel u jednotlivých úkolů vidí navíc některé ovládací prvky. Rozdíl mezi učitelským a studentským pohledem na formulář pro odevzdání úkolu ilustrují následující obrázky (Obr. 11 a Obr. 12).

Úkoly

Úprava barvy a kontrastu pomocí automatických korekcí

Odevzdané práce (1)

Použijte svou fotografii dle vlastního uvážení.

Nastavit deadline pouze pro tuto skupinu:

dd.mm.rrrr

hh:mm

Uložit

Obr. 11: Formulář pro odevzdání úkolu z pohledu učitele.

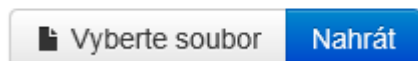
Úkoly

Úprava barvy a kontrastu pomocí automatických korekcí

Použijte svou fotografii dle vlastního uvážení.

Váš úkol byl odevzdán 17.5.2013 13:08

Uložený soubor: boty-final.jpg



Obr. 12: Formulář pro odevzdání úkolu z pohledu studenta.

V systému je umožněno pracovat se třemi druhy úkolů, které se odlišují typem odpovědi:

- Text – zde se očekává jednoduchá slovní odpověď.
- Soubor – učitel může zvolit maximální velikost odevzdávaného souboru a příponu, již daný soubor musí mít.
- Obrázek – toto je speciální typ souboru, se kterým může učitel po odevzdání dále pracovat.

V případě, že daný úkol je typu *obrázek* má učitel k dispozici možnost provést prezentaci obrázků přímo z přehledu odevzdaných souborů. Také může vybrat práce studentů a vygenerovat pro skupinu speciální dotazník pro hodnocení těchto prací. Poté, co studenti tyto práce ohodnotí, má učitel k dispozici tzv. hitparádu nejlepších prací, kterou může studentům ještě v téže hodině prezentovat.

6.5 Upozornění

Účelem tohoto modulu je, aby se nestávalo, že učitel studentům něco slíbí, a pak na to zapomene. Učitel má zde tedy možnost pomocí jednoduchého formuláře přidat upozornění, které se zobrazuje na všech stránkách dané skupiny (Obr. 13) a na úvodních stránkách uživatelů (Obr. 14).

Upozornění

Písemka 22. 5. 2013

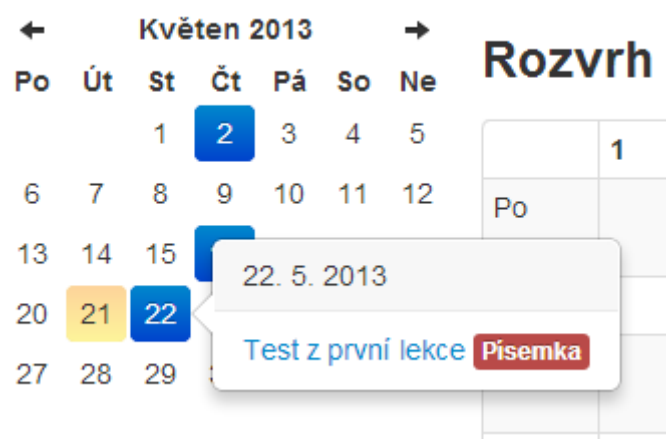
Test z první lekce

Zkoušení 30. 5. 2013

Opakování

Budu zkoušet z....

Obr. 13: Upozornění na stránkách skupiny



Obr. 14: Upozornění na úvodní stránce

6.6 Zprávy

Systém zpráv má sloužit k případné komunikaci mezi učitelem a studenty případně mezi učitelem a celou skupinou. Komunikace mezi studenty navzájem byla záměrně zakázána. Zprávu lze odeslat pomocí speciálního formuláře nebo vygenerovat automaticky. Tato možnost je dostupná u pozvánek k dotazníkům.

V případě, že uživatel obdrží pozvánku, je o tom ihned informován pomocí ikonky v hlavním menu (Obr. 15).



Obr. 15: Ikona znázorňující novou zprávu

6.7 Nastavení

Do nastavení aplikace mají přístup pouze uživatelé s administrátorským oprávněním.

V rámci nastavení lze provádět:

- Import ze systému Bakaláři. Tento import vytvoří seznam tříd a studentů školy.
- Správu tříd. Třídy jsou důležité při registraci studentů a jejich následném vyhledávání (filtrování) v seznamech.
- Správu učeben, které se zobrazují v rozvrhu hodin.
- Správu uživatelů.

Jedinou výjimkou, kterou mají učitelé, je přístup k seznamu studentů.

6.8 Dotazníky

Ke správě dotazníků mají přístup učitelé a administrátoři. Mohou zde vytvářet nové dotazníky, případně pracovat s těmi stávajícími.

Při vytváření dotazníku je nutné zadat pouze jeho název. Po vytvoření následuje přesměrování na editaci.

Editace dotazníku je rozdělena pomocí záložek na editaci otázek a editaci základních informací (podrobností), které byly ve formuláři při vytváření dotazníku.

Editace otázek je také rozdělena na dvě části. V levé části se nachází seznam otázek, u kterých lze uchopením myši měnit jejich pořadí. V pravé části se nachází místo pro formulář.

V této verzi aplikace jsem implementoval 12 různých typů otázek.

V případě, že se jedná o typ otázky s uzavřenou odpovědí, je zde na výběr ze 13 předdefinovaných sad odpovědí. Lze také vybrat 14. možnost – vlastní. V tomto případě se objeví pole pro zadání vlastních odpovědí. Odpovědi se zde píšou na samostatný řádek.

6.9 Vzorové evaluační dotazníky

Jedním z úkolů, které jsou v zadání této práce, je navrhnout a implementovat do aplikace příklady vzorových evaluačních dotazníků.

Navrhl jsem tedy dva ukázkové dotazníky:

- Hodnocení učitele.

- Názor žáků na dnešní hodinu.

Tyto dotazníky jsem pomocí aplikace vytvořil a následně naplnil testovacími daty. Ukázka formulářů vygenerovaných aplikací a ukázka výsledných grafů je v příloze P I.

7 ZABEZPEČENÍ APLIKACE

Internetová služba WWW zaznamenala za posledních 20 let velký rozmach. Toto velké rozšiřování se neprojevilo pouze v množství jednotlivých webových stránek, ale hlavně v jednotlivých technologiích a na nich postavených aplikacích. Vzhledem k dynamickému tempu vzniku nových technologií je logické, že tyto technologie obsahují mnoho bezpečnostních děr. Ačkoli se jejich vývojáři snaží tyto nedostatky odstranit, vždy se objeví nějaké nové riziko. Webové aplikace jsou na těchto technologiích závislé, a tedy jsou náchylné na rizika, která dané technologie přináší.

V této kapitole chci čtenáře seznámit s nejnámějšími technikami útoků na webové aplikace. Pokusím se přiblížit příčiny, proč k těmto útokům může dojít, a také jak jim předcházet.

7.1 Cross-Site Scripting

Cross-Site Scripting (XSS) je jeden z nejčastějších útoků. Je způsoben neošetřením uživatelských vstupů. Útočník díky tomu může do stránky podstrčit kus kódu, se kterým aplikace nepočítá, následkem čehož může dojít k poškození vzhledu stránek, jejich znefunkčnění, nebo dokonce k získání citlivých údajů ostatních návštěvníků stránek.

Útočník do stránek může podstrčit kód několika způsoby. Nejčastěji to je z:

- formuláře,
- URL adresy,
- databáze,
- cookies.

Data předávaná výše uvedenými způsoby mohou být útočníkem pozměněna, a proto bychom měli předpokládat, že pozměněna jsou. V jazyce PHP existuje několik funkcí, které se dají k ošetření vstupů použít.

Addslashes – tato funkce přidává zpětné lomítko před speciální znaky. Díky tomu se takto ošetřená mohou ukládat do databáze. V PHP existuje direktiva `magic_quotes_gpc`, která plní stejnou funkci. Pokud je aktivní a použijeme funkci `addslashes`, dojde k dvojitmu přidání zpětného lomítka.

urlencode - tato funkce slouží k ošetření dat, předávaných URL řetězcem. `urlencode` nahradí všechny nealfanumerické znaky jejich URL reprezentací (znakem „%“ a dvojicí hexadecimálních číslic).

htmlspecialchars – tuto funkci bychom měli používat všude tam, kde víme, že vypisovaný text neobsahuje žádné HTML značky. Pokud se nějaká HTML značka v textu objeví, tato funkce přemění řídicí znaky na odpovídající entity, tedy místo značky „<“ se vypíše „<“ apod.

7.2 SQL injection

Tento útok může být použit pro získání dat z databáze, ke kterým nemáme přístup či dokonce ke smazání samotných dat. Útočník může tohoto docílit pomocí vsunutí svého kódu do SQL dotazu. K tomuto dochází nejčastěji v případech, kdy do SQL dotazu vkládáme data z URL adresy či z formuláře a tato data neošetříme.

Ukázka dotazu, náchylného k útoku:

URL adresa: `page.php?id=1`

```
$dotaz = "SELECT * FROM tabulka WHERE id=".$_GET["id"];
```

Za předpokladu, že útočník pozmění adresu na „`page.php?id=1 OR 1=1`“, bude vypadat výsledný dotaz, který se provede v databázi následovně: „`SELECT * FROM tabulka WHERE id=1 OR 1=1`“

Útočník tedy provede výběr všech řádků v tabulce, ne jen jednoho konkrétního, jak zamýšlel programátor.

Ochranou proti SQL injection je ověřovat správnost předávaných parametrů. Pokud očekáváme číselnou hodnotu, tak je potřeba toto ověřit pomocí funkce `is_numeric`. V případě textových hodnot je vhodné použít výše zmíněnou funkci `addslashes`.

7.3 Cross-Site Request Forgery

Tento typ útoku zneužívá stálého přihlášení do aplikace. Tento útok umožňuje samotný protokol HTTP. Tento protokol je bezstavový a bylo nutné vymyslet techniky k zapamatování identity uživatele. Těmito technikami jsou především session a cookies.

Uživatel, který je tedy přihlášen do aplikace, má ve svém prohlížeči v cookies uložen session identifikátor, který se s každým požadavkem posílá na server. Útočník následně

může v uživatelské prohlídce vynutit načtení konkrétní stránky aplikace, aniž by o tom oběť musela vědět. Díky tomu může útočník například mazat příspěvky, odesílat zprávy či provádět jakékoli akce, ke kterým má oběť oprávnění.

Aby byl tento útok úspěšný, musí útočník znát strukturu napadené aplikace a musí nějakým způsobem navázat kontakt s obětí.

Útok se často provádí z cizích stránek, které si oběť zobrazí. Zde může být vložen například neviditelný obrázek s URL adresou, která vede na napadenou aplikaci.

Příklad takového obrázku:

```

```

Může se zde také vyskytovat neviditelný formulář, který je po načtení stránky odeslán pomocí javascriptu.

Ukázka takového formuláře:

```
<form action="http://aplikace.cz/clanek.php" id="form" method="post">
<input type="hidden" name="action" value="delete" />
<input type="hidden" name="id" value="123" />
</form>
<script type="text/javascript">
document.getElementById('form').submit();
</script>
```

Obrana proti CSRF není nejjednodušší. Jednou z možností je kontrolovat při každé akci http hlavičku „Referer“. V této hlavičce se vyskytuje informace o tom, z jaké stránky je na cílovou adresu odkazováno. Nevýhodou je, že ne všichni uživatelé mají odesílání této hlavičky ve svém prohlížeči povoleno. Navíc takovou hlavičku lze snadno podvrhnout. K tomu může sloužit třeba funkce `fsockopen`. Uživatel také může do aplikace vstoupit z emailu, který aplikace sama odeslala.

Nejosvědčenejší ochranou proti CSRF se zdá být použití autorizačního tokenu. Tento způsob ochrany je technologicky náročný, ale má také nejlepší výsledky. Ke každému formuláři je přiřazen token, který je zároveň uložen v session. Při odeslání formuláře se kontroluje existence daného tokenu. Pokud je vše v pořádku, může se formulář zpracovat. V opačném případě se jedná zřejmě o útok a aplikace na to již může zareagovat. Bohužel, jak je vidět z popisu této metody, má i ona své nedostatky. Jedná se například

o omezení práce s aplikací ve více oknech. Aby tato metoda byla bezpečná, musí autorizační token splňovat následující požadavky:

- nesmí být odvoditelný,
- musí být unikátní,
- dostatečně dlouhý,
- časově omezený.

Při každé příležitosti by měl být vygenerován nový.

7.4 Ochrana aplikace ve frameworku Nette

Jak jsem již zmínil výše, svou aplikaci jsem postavil na frameworku Nette. Tento Framework se pyšní svým přístupem k zabezpečení odvozených aplikací. Ve výchozím nastavení je programátor chráněn proti všem výše zmíněným útokům.

Context-Aware Escaping je technologie, kterou Nette obsahuje ve svém šablonovacím systému proti riziku XSS útoků. Tato technologie automaticky ošetřuje všechny výstupy. Pokud chce programátor vypsat neošetřený výstup, musí toto v šabloně explicitně vyznačit.

Ve frameworku je také implementována ochrana proti CSRF. Každý formulář lze ošetřit pomocí příkazu **\$form->addProtection()**. Tento příkaz vloží do vygenerovaného formuláře kontrolní token. Tento token také uloží do session. Po odeslání tohoto formuláře zároveň automaticky tento token vyhodnotí.

Ve své aplikaci jsem se snažil všechny formuláře takto ochránit.

SQL injection je v Nette ošetřena díky databázové vrstvě Nette\Database. Důležité je, aby všechny vstupní hodnoty byly vkládány do dotazu standardní cestou. Této praxe jsem se držel, a tedy by aplikace neměla být náchylná ani proti tomuto typu útoku.

8 IMPLEMENTACE APLIKACE NA INTERNETOVÝ SERVER

Aplikaci jsem vyvíjel na svém počítači, na kterém mám následující konfiguraci vývojového prostředí:

- Server Apache/2.0.64 (Win32).
- PHP Version 5.3.18.
- MySQL v. 5.5.28.

Pro účely implementace a testování aplikace jsem zvolil umístění na vlastní doméně **dp.ancinec.cz**. Tato doména i hosting se nachází na serverech hostingové společnosti Wedos Internet, a.s.

Na testovacím serveru se nachází následující konfigurace prostředí:

- Server Apache.
- PHP Version 5.3.21.
- MySQL v. 5.5.29.

Z výše uvedených přehledů je patrné, že konfigurace na mém počítači a na testovacím serveru jsou téměř identické. Hlavní rozdíl je pouze v operačním systému, ve kterém jsou nainstalovány. V mém počítači je nainstalován Windows8 a na testovacím serveru je distribuce Linuxu.

8.1 Testování aplikace

Během vývoje jsem aplikaci průběžně testoval lokálně a také na výše popsaném internetovém serveru. Tento postup se mi zdál vhodný z toho důvodu, že v případě odlišného nastavení PHP, serveru Apache či databázového serveru MySQL budu schopen ihned reagovat na zjištěnou situaci a přizpůsobit se jí. Tento postup je vhodnější, než zjišťovat možné problémy až po nasazení aplikace.

Jak se během vývoje a průběžného testování ukázalo, rozdílná prostředí způsobovala drobné problémy. Jednalo se například o to, že přísnější bezpečnostní politika na internetovém serveru mi nedovolovala vytvářet dočasné soubory tak, jak jsem toho byl schopen na svém počítači. Tento problém jsem vyřešil ukládáním souborů do složky temp, kterou mimo jiné využívá i Framework Nette.

Finální verzi aplikace jsem na výše zmíněné adrese otestoval z funkčního hlediska. Nenašel jsem žádnou část systému, která by se chovala nestandardně.

Zkoušel jsem také chování aplikace na nejrůznější neočekávaná vstupní data, která se uživatel může pokusit podstrčit (třeba za účelem získání přístupu do části aplikace, ke které nemá oprávnění). Ani zde se mi nepodařilo odhalit žádný problém.

Pro testovací účely jsem vytvořil tyto tři účty:

Administrátor:

- Email: admin@ancinec.cz
- Heslo: Dp_2013

Učitel:

- Email: ucitel@ancinec.cz
- Heslo: Dp_2013

Student:

- Email: student@ancinec.cz
- Heslo: Dp_2013

ZÁVĚR

Hlavním cílem této práce bylo vytvořit webovou aplikaci, která bude učitelům pomáhat při výuce i při její evaluaci. Pro podporu výuky byl do aplikace zakomponován systém pro prezentaci učiva v kurzech a pro odevzdávání úkolů. Dále byla aplikace rozšířena o možnost komunikace prostřednictvím zpráv a také o zobrazování upomínek.

Za velmi užitečnou funkci považuji možnost vytvořit z odevzdaných úkolů dotazník, který si studenti sami bodově ohodnotí. Učitel následně může studenty s nejlepšími pracemi seznámit. Jedná se o velmi užitečný motivační nástroj.

V rámci podpory evaluace je část aplikace věnována dotazníkovému systému. Učitelé mají k dispozici moduly pro vytváření dotazníků, pro jejich následné zpřístupnění i vyhodnocení.

V teoretické části jsem se zabýval přiblížením pojmů evaluace a e-learning. Také jsem zde popsal jednotlivé technologie, jež byly použity při vytváření aplikace.

Praktická část byla věnována analýze požadavků na aplikaci a bližšímu seznámení s jednotlivými částmi aplikace. Dále je zde popsán návrh struktury aplikace, a to jak adresářová struktura, tak i návrh databáze. Praktická část také obsahuje popis řešení aplikace a ukázkou vzorových dotazníků.

Další kapitola se zabývá zabezpečením aplikace. Jsou zde vyjmenovány a blíže popsány nejčastější typy útoků, se kterými se webové aplikace nejčastěji potýkají. U každé bezpečnostní hrozby jsou popsány i způsoby, kterými se lze proti těmto útokům chránit.

Aplikace je v této době nasazena na testovacím serveru <http://dp.ancinec.cz> a není tedy oficiálně spuštěna. Verze, která je dostupná na testovacím serveru, byla otestována a je k dispozici i na CD-ROM, které je přiloženo k této práci.

CONCLUSION

The main aim of this thesis was creating of a web application which can help lecturers in tuition processes and evaluation. System for presentation of curriculum and system for collecting of homework was implemented into the application. This part of the application was extended by a message communication module and a module which is able to show reminders.

Very useful is tool for creation of survey from student's works. Students can rate their own works and after that teacher can present the best results. It's kind of tools for motivation of students.

One of the main parts of the application is about survey system. Modules for creating, publishing and evaluation of surveys are available for teachers.

In the theoretical part, expressions like evaluation and e-learning were explained. In this part, technologies which were used in this application are described too.

The practical part is about requirement analysis and introduction to particular parts of the application. The application's structure from the perspective of the directory structure and database design is described further in the text. Description of the solution and examples of questionnaires are involved in this part.

Next part is about web security. The most common types of security threats are listed and described there. A method of protection is stated for each of the security threats.

At the moment the application is published only on the test server <http://dp.ancinec.cz>. The version on the test server has been tested tested and is available on the CD-ROM which is attached to this thesis.

SEZNAM POUŽITÉ LITERATURY

- [1] PRŮCHA, Jan, 1996. *Pedagogická evaluace: hodnocení vzdělávacích programů, procesů a výsledků*. Vyd. 1. V Brně: Masarykova univerzita, 166 s. ISBN 80-210-1333-8.
- [2] VAŠŤATKOVÁ, Jana, 2010. *Úvod do pedagogické evaluace a jejích metod: studijní texty pro distanční vzdělávání : projekt Školský management*. 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 75 s. ISBN 978-80-244-2681-5.
- [3] SLAVÍK, Jan, 1999. *Hodnocení v současné škole: východiska a nové metody pro praxi*. Vyd. 1. Praha, 190 s. Pedagogická praxe. ISBN 80-717-8262-9.
- [4] NEZVALOVÁ, Danuše. *Pedagogická evaluace. Metodický portál RVP: inspirace a zkušenosti učitelů*[online]. 25. 10. 2006 [cit. 2013-05-14]. Dostupné z: <http://clanky.rvp.cz/clanek/o/z/963/PEDAGOGICKA-EVALUACE.html>
- [5] STARÝ, Karel, 2005. *Autoevaluace školy*. In: *Národní ústav odborného vzdělávání* [online]. Praha [cit. 2013-05-14]. Dostupné z: http://www.nuov.cz/uploads/Publikace/Zajistovani_kvality_OV/Autoevaluace_skoly.pdf
- [6] SEBEROVÁ, Alena a Martin MALČÍK, 2009. *Autoevaluace školy - od teorie k praxi a výzkumu*. Vyd. 1. V Ostravě: Ostravská univerzita v Ostravě, Pedagogická fakulta, 133 s. ISBN 978-807-3687-595.
- [7] PRŮCHA, Jan, Eliška WALTEROVÁ a Jiří MAREŠ, 2003. *Pedagogický slovník*. 4., aktualiz. vyd. Praha: Portál, 322 s. ISBN 80-717-8772-8.
- [8] ZOUNEK, Jiří, 2009. *E-learning - jedna z podob učení v moderní společnosti*. Vyd. 1. Brno: Masarykova univerzita, 161 s. ISBN 978-80-210-5123-2.
- [9] STRÍTESKÁ, Hana, 2003. *Historie e-learningu v České republice*. [online]. [cit. 2013-05-15]. Dostupné z: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xstrites.htm>
- [10] KVĚTOŇ, Karel, 2007. *Základy e-learningu. Centrum informačních technologií OU* [online]. [cit. 2013-05-15]. Dostupné z: http://cit.osu.cz/dokumenty/elearning_kkveton.pdf
- [11] KOPECKÝ, Kamil, 2006. NET-UNIVERSITY S.R.O. *Co je e-learning?: Úvod do problematiky*. Dostupné z: www.net-university.cz/download/seminar/1.ppt

- [12] KOPECKÝ, Kamil, 2007. Blended learning jako skutečně efektivní přístup ke vzdělávání. *Net univesity: e-learning - education - online* [online]. [cit. 2013-05-15]. Dostupné z: <http://www.net-university.cz/elearning/27-blended-learning-jako-skuten-efektivni-pistup-ke-vzdlavani>
- [13] PEJŠA, Jan, 2004. KONTIS S.R.O. *LCMS a LMS, vývoj kurzů*. 6 s. Dostupné z: http://www.e-learn.cz/soubory/LMS_LCMS.pdf
- [14] PHP Manual. THE PHP GROUP. *PHP Manual* [online]. [1997] 17 May 2013 [cit. 2013-05-18]. Dostupné z: <http://www.php.net/manual/en/index.php>
- [15] CASTRO, Elizabeth, 2012. *HTML5 a CSS3*. Brno: Computer press. ISBN 978-802-5137-338.
- [16] KOSEK, Jiří, 1998. *HTML: tvorba dokonalých www stránek : podrobný průvodce*. Vyd. 1. Praha: Grada, 291 s. Průvodce (Grada). ISBN 80-716-9608-0.
- [17] Facts about W3C. WORLD WIDE WEB CONSORTIUM. *W3C* [online]. [2012] [cit. 2013-05-18]. Dostupné z: <http://www.w3.org/Consortium/facts>
- [18] ASLESON, Ryan, 2006. *AJAX: vytváříme vysoce interaktivní webové aplikace*. Vyd. 1. Překlad Jakub Zemánek. Brno: Computer Press, 269 s. ISBN 80-251-1285-3.
- [19] RESIG, John, 2007. *JavaScript a Ajax: moderní programování webových aplikací*. Vyd. 1. Překlad Ondřej Baše, Ondřej Žížka. Brno: Computer Press, 360 s. ISBN 978-80-251-1824-5.
- [20] Ajax: A New Approach to Web Applications. In: GARRETT, Jesse James. *Adaptive path* [online]. February 18, 2005 [cit. 2013-05-19]. Dostupné z: <http://adaptivepath.com/ideas/ajax-new-approach-web-applications>
- [21] VRÁNA, Jakub. PDO a další novinky v PHP 5.1. *PHP triky* [online]. 28.11.2005 [cit. 2013-05-19]. Dostupné z: <http://php.vrana.cz/pdo-a-dalsi-novinky-v-php-5-1.php>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

AJAX	Asynchronous Javascript and XML.
API	Application Programming Interface
CBT	Computer Based Training
CSRF	Cross-Site Request Forgery
CSS	Cascade Style Sheets
DOM	Document Object Model
ECMA	European Computer Manufacturers Association
HTML	Hypertext Markup Language
JSON	JavaScript Object Notation
JSONP	JavaScript Object Notation with Padding
LCMS	Learning Content Management System
LMS	Learning Management System
MVC	Model View Controller
ORM	Object Relational Mapping
PDO	PHP Data Objects
PHP	Hypertext Preprocesor
RVP	Rámcový vzdělávací program
SQL	Structured Query Language
WBT	Web-Based Training
WWW	World Wide Web
W3C	World Wide Web Consorcium
XML	Extensible Markup Language
XSS	Cross-Site Scripting

SEZNAM OBRÁZKŮ

Obr. 1: Klasický (synchronní) model webové aplikace. (Garrett, 2005).....	22
Obr. 2: Model ajaxové webové aplikace (Garrett, 2005)	23
Obr. 3: Životní cyklus presenteru (Nette Framework: Dokumentace)	26
Obr. 4: Adresářová struktura.....	34
Obr. 5: Struktura tabulek pro kurzy	38
Obr. 6: Struktura tabulek pro úkoly	39
Obr. 7: Struktura tabulek pro zprávy	40
Obr. 8: Struktura tabulek pro systém upozornění	41
Obr. 9: Struktura tabulek pro dotazníky	42
Obr. 10: Rozvrh hodin na úvodní stránce	43
Obr. 11: Formulář pro odevzdání úkolu z pohledu učitele	44
Obr. 12: Formulář pro odevzdání úkolu z pohledu studenta.	45
Obr. 13: Upozornění na stránkách skupiny	46
Obr. 14: Upozornění na úvodní stránce	46
Obr. 15: Ikona znázorňující novou zprávu	46

SEZNAM TABULEK

Tab. 1: Rozdíly mezi hodnocením a evaluací podle Nezvalové (2006)	13
---	----

SEZNAM PŘÍLOH

PŘÍLOHA P I: Ukázkové dotazníky

PŘÍLOHA P II: Dokumentace ke správě a používání aplikace

PŘÍLOHA P III: CD-ROM

PŘÍLOHA P I: UKÁZKOVÉ DOTAZNÍKY

V této příloze se nachází snímky obrazovky s ukázkovými dotazníky. U obou dotazníků uvedu nejdříve návrh, poté snímek obrazovky s formulářem pro vyplnění a následně náhled na graf, který je k danému dotazníku k dispozici.

Hodnocení učitele:

*Na všechny otázky se odpovídá výběrem z možností
souhlasím - spíše souhlasím – nevím - spíše nesouhlasím - nesouhlasím.*

1. Didaktické schopnosti

Učitel vysvětluje učivo jasně a srozumitelně
Když něco nevím, nebojím se učitele na to zeptat
Učitel se v hodině zajímá, zda učivu rozumím
Učitel ve mě probudil zájem o daný předmět
Učitel má v hodině dobrou kázeň

2. Chování k žákům

Mám pocit, že učiteli na mně záleží
Učitel mě pochválí, když se mi něco podaří
Chová se ke všem stejně (nemá oblíbené či neoblíbené žáky)
Tento učitel mi pomůže, když budu mít problémy s učením

3. Při výuce

Učitel dokáže přiznat svou chybu
Při hodnocení je spravedlivý
Učitel vytváří v hodině příjemnou atmosféru

Hodnocení učitele

1. Didaktické schopnosti (Tato otázka je povinná)

	Souhlasím	Spíše souhlasím	Nevím	Spíše nesouhlasím	Nesouhlasím
Učitel vysvětluje učivo jasně a srozumitelně	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Když něco nevím, nebojím se učitele na to zeptat	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Učitel se v hodině zajímá, zda učivu rozumím	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Učitel ve mě probudil zájem o daný předmět	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Učitel má v hodině dobrou kázeň	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Chování k žákům (Tato otázka je povinná)

	Souhlasím	Spíše souhlasím	Nevím	Spíše nesouhlasím	Nesouhlasím
Mám pocit, že učiteli na mně záleží	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Učitel mě pochválí, když se mi něco podaří	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Chová se ke všem stejně (nemá oblíbené či neoblíbené žáky)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Učitel mi pomůže, když budu mít problémy s učením	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

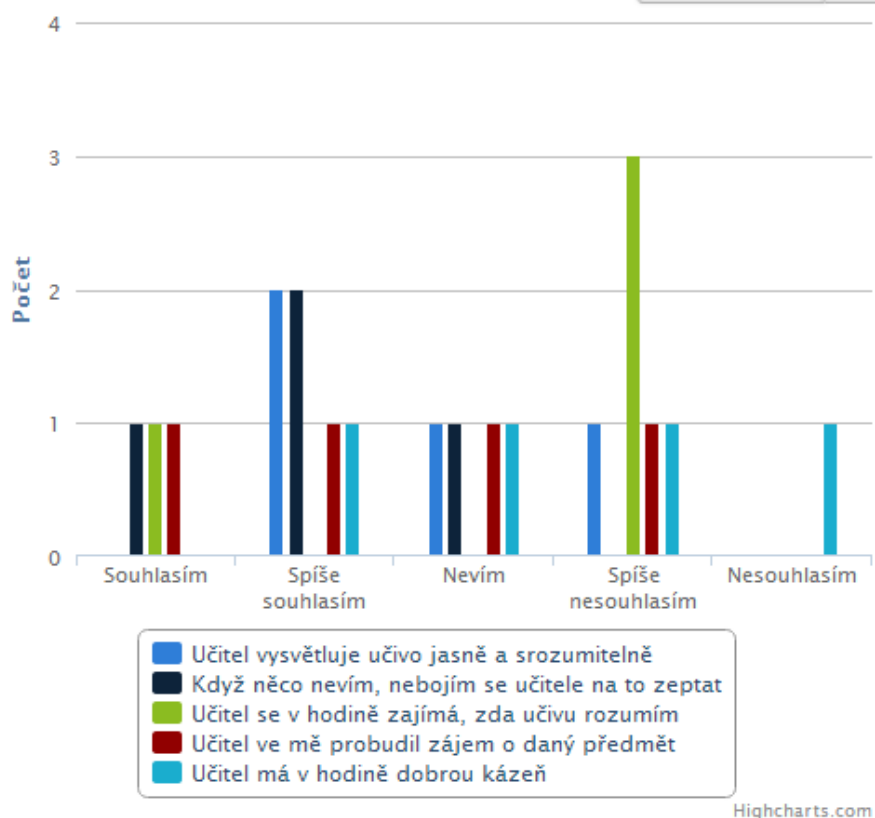
3. Při výuce (Tato otázka je povinná)

	Souhlasím	Spíše souhlasím	Nevím	Spíše nesouhlasím	Nesouhlasím
Učitel dokáže přiznat svou chybu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Při hodnocení je spravedlivý	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Učitel vytváří v hodině příjemnou atmosféru	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Aplikace pro podporu a evaluaci výuky

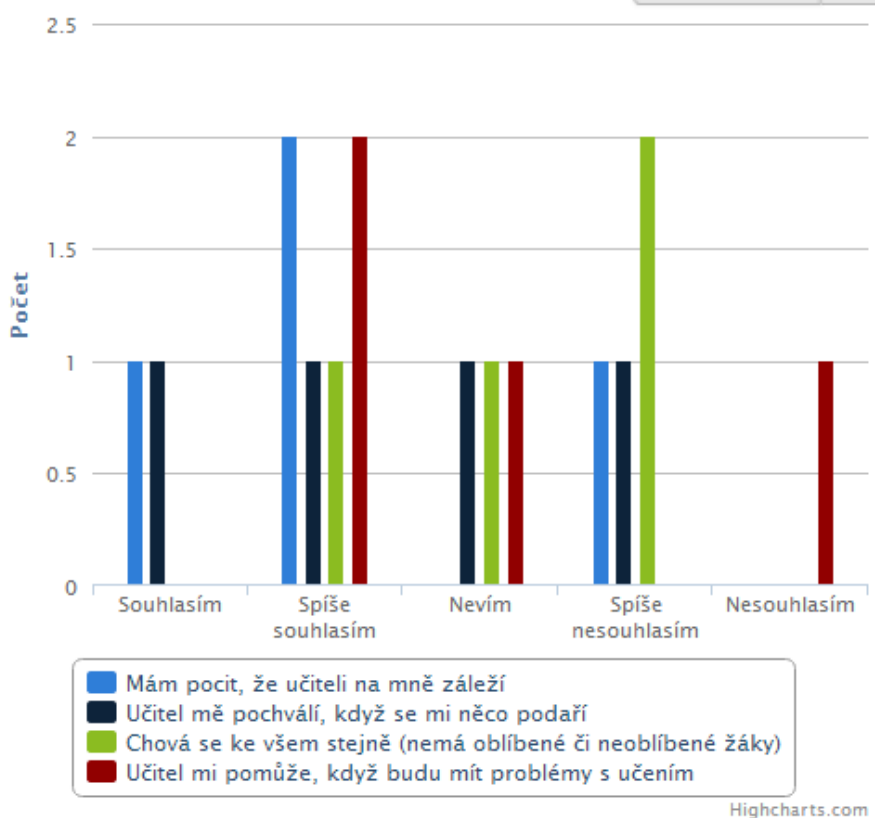
1. Didaktické schopnosti

Sloupcový graf



2. Chování k žákům

Sloupcový graf



Názor žáků na dnešní hodinu:

- 1. Co nového jste se v dnešní hodině naučil/a?**
(*textová odpověď*)
- 2. Bylo probírané učivo jasně a srozumitelně prezentováno?**
(*Uzavřená odpověď: Zcela, Z větší části, Z poloviny, Jen v některých případech, Vůbec*)
- 3. Vysvětlil vyučující všechny nejasnosti?**
(*ano x ne*)
- 4. Je něco, čemu jste nerozuměl/a a nezeptal/a jste se na to? Co to bylo?**
(*textová odpověď*)
- 5. Jak byste celkově ohodnotil/a úroveň hodiny?**
(*uzavřená odpověď' výborně až nedostatečně*)
- 6. Stručně vysvětlete, proč jste dal/a v páté otázce právě takové hodnocení.**
(*textová odpověď*)
- 7. Ke kterému tvrzení se přikláníte?**
(*sémantický diferenciál: výběr v rozmení 1 – 7*)
 - a. Méně úkolů v hodině x Více úkolů v hodině
 - b. Méně domácích úkolů x Více domácích úkolů
 - c. Méně vysvětlujících textů x Více vysvětlujících textů
 - d. Méně času na úkoly v hodině x Více času na úkoly v hodině
- 8. Máte připomínky k dnešní hodině? Co byste chtěl/a zlepšit?**
(*textová odpověď*)

Názor žáků na dnešní hodinu

1. Co nového jsi se v dnešní hodině naučil?

2. Bylo probírané učivo jasně a srozumitelně prezentováno?

- Zcela
 - Z větší části
 - Z poloviny
 - Jen v některých ohledech
 - Vůbec
-

3. Vysvětlil vyučující všechny nejasnosti?

-- vyberte -- ▼

4. Je něco, čemu jste nerozuměl/a a nezeptal/a jste se na to? Co to bylo?

5. Jak byste ohodnotil dnešní hodinu?

Výborný **Chvalitebný** **Dobry** **Dostatečný** **Nedostatečný**

Komentář

Nápověda: Ohodnoťte jako ve škole. Do textového pole uveďte, proč jste zvolili právě takové hodnocení.

6. Ke kterému tvrzení se přikláníte?

1 **2** **3** **4** **5** **6** **7**

Méně úkolů v hodině Více úkolů v hodině

Méně domácích úkolů Více domácích úkolů

Méně vysvětlujících textů Více vysvětlujících textů

Méně času na úkoly v hodině Více času na úkoly v hodině

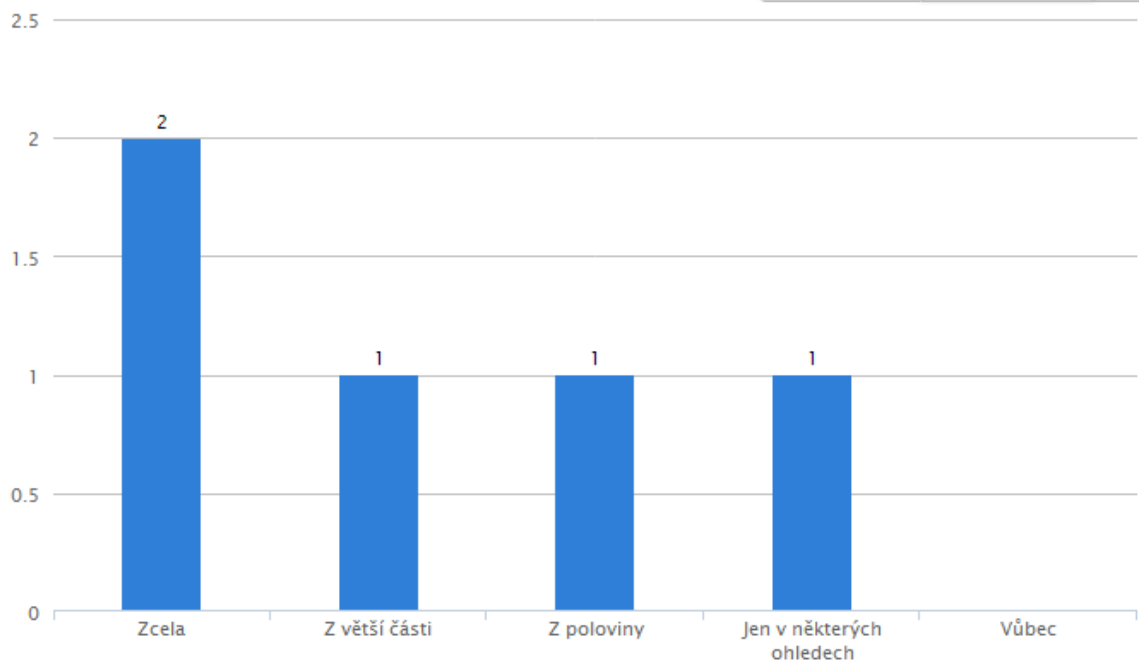
7. Máte připomínky k dnešní hodině? Co byste chtěli zlepšit?

Ukázka vygenerovaných grafů:

2. Bylo probírané učivo jasně a srozumitelně prezentováno?

Koláčový graf

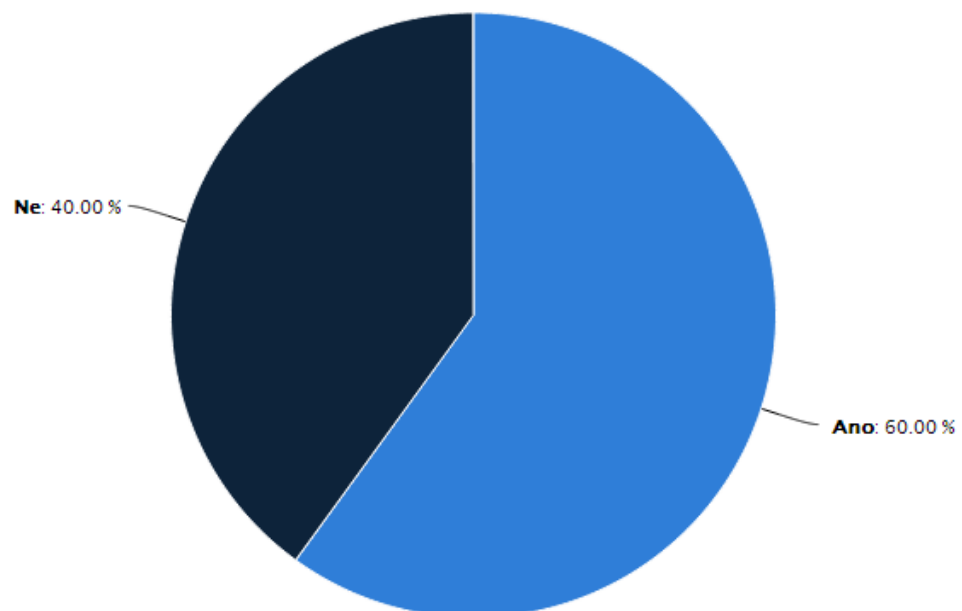
Sloupcový graf



3. Vysvětlil vyučující všechny nejasnosti?

Koláčový graf

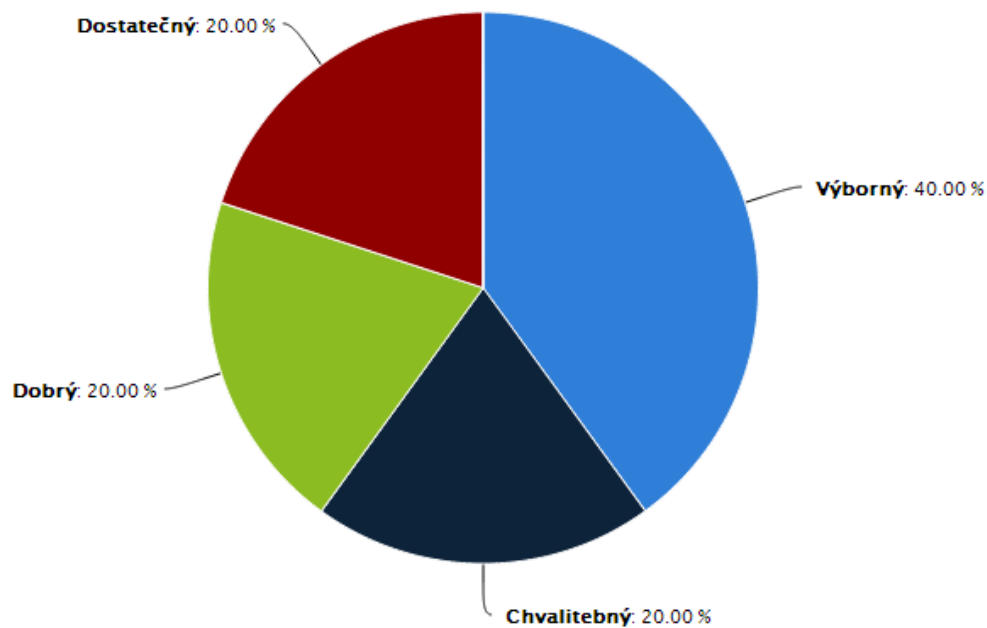
Sloupcový graf



5. Jak byste ohodnotil dnešní hodinu?

Koláčový graf

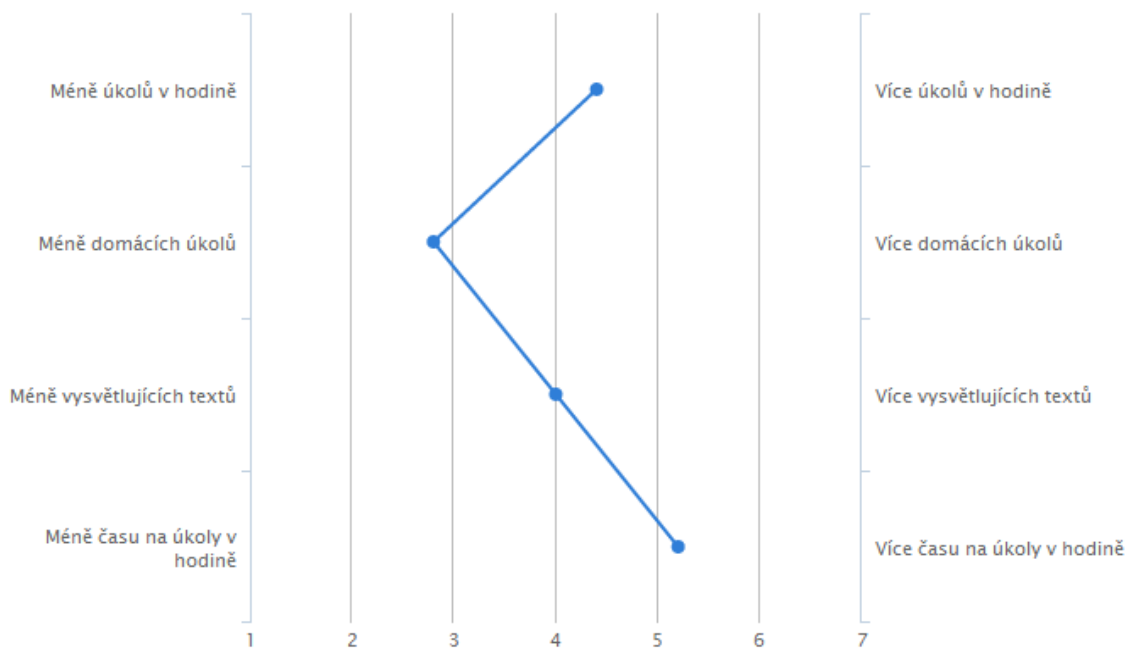
Sloupcový graf



Highcharts.com

6. Ke kterému tvrzení se přikláníte?

Graf



Highcharts.com

PŘÍLOHA P II: DOKUMENTACE KE SPRÁVĚ A POUŽÍVÁNÍ APLIKACE

Registrace

Formulář pro registraci naleznete pod odkazem, který se nachází v pravé horní části stránky.



Ve formuláři jsou pro studenty povinné všechny položky. Pouze učitelé nemusejí vyplňovat třídu.

Přihlášení

Formulář pro přihlášení se nachází pod odkazem vedle tlačítka registrovat, případně na úvodní stránce v pravé části.

Po přihlášení se uživateli změní horní lišta:



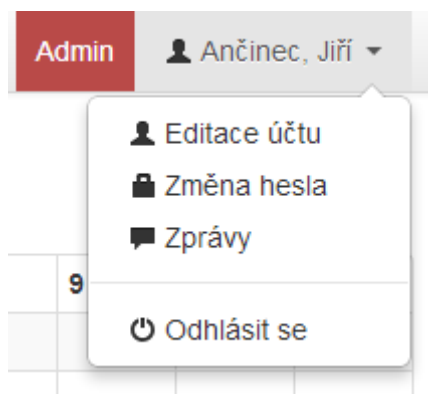
Vedle jména a příjmení se nachází barevně označená role uživatele.

Zapomenuté heslo

V případě, že uživatel zapomněl své heslo, může si jej obnovit. Odkaz pro obnovu hesla se nachází pod přihlašovacím formulářem. Po kliknutí na tento odkaz je potřeba do formuláře uvést emailovou adresu, kterou použil při registraci. Na tuto adresu bude následně zaslán odkaz, který zpřístupní formulář pro vytvoření nového hesla.

Editace účtu

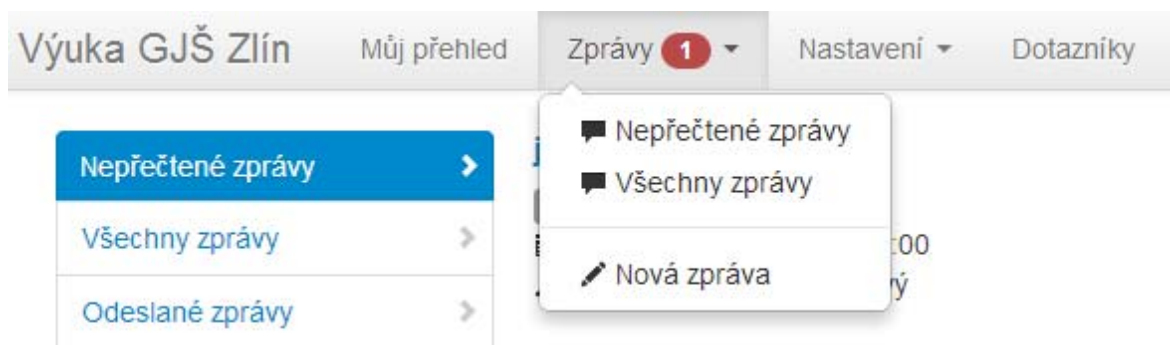
Po přihlášení do aplikace má uživatel po kliknutí na své jméno v pravé horní části aplikace k dispozici nabídku pro práci s účtem.



Ke změně hesla je potřeba zadat nejprve původní heslo a poté nové heslo a nové heslo znovu (pro kontrolu).

Zprávy

V případě, že obdržíte novou zprávu, se ihned objeví v horní liště u tlačítka „Zprávy“ ikonka s počtem nepřečtených zpráv.



Po kliknutí na tlačítko „Zprávy“ vyvoláte nabídku, pomocí které můžete zobrazit nepřečtené zprávy, všechny zprávy případně můžete začít psát novou zprávu.

Nová zpráva

Při psaní nové zprávy je potřeba vybrat příjemce. Učitelé mají na výběr mezi odesláním zprávy konkrétním uživatelům, nebo skupinám.

V případě, že zvolí „Pro studenty“, zpřístupní se mu v položce „Příjemce“ vyhledávací pole, pomocí kterého může studenty vyhledat.

Příjemce *

Předmět *

Please enter 1 more character

V případě, že zvolí možnost „Pro skupiny“, pole „Příjemce“ mu nabídne výběr skupin,

Příjemce *

Předmět *

G1

G2

Studenti mají k dispozici pouze vyhledávací pole, pomocí kterého mohou vyhledat své učitele. Vyhledávací pole vypadá stejně, jako vyhledávání studentů v prvním příkladu.

Detail zprávy

Detail zprávy lze otevřít kliknutím na předmět ve výpisu zpráv.

V detailu zprávy můžete začít psát odpověď kliknutím na tlačítko „Odpovědět“

Připomínky / Upozornění

← Květen 2013 →

Po	Út	St	Čt	Pá	So	Ne
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23			
27	28	29	30			

Opakování
Zkoušení 30.5.2013
 Adobe Photoshop CS5 (G1)
 Budu zkoušet z...

Rozvrh

	1
Po	
Út	
St	
Čt	
Pá	

30. 5. 2013
 Opakování Zkoušení

Na úvodní stránce se nachází kalendář s připomínkami. V případě, že je v kalendáři modré políčko, nachází se v tomto dni jedna či více připomínek. Tyto připomínky si zobrazíte kliknutím na dotyčný den. Zobrazí se Vám bublina s přehledem připomínek, které jsou barevně rozlišeny. Po kliknutí na název se pod kalendářem objeví detail.

Kurzy

Kurzy si můžete zobrazit ze seznamu kurzů, který se nachází na úvodní straně. Studenti zde mají zobrazeny kurzy, do kterých jsou zapsáni, a kurzy, které jsou veřejně dostupné.

Učitelé mají na úvodní straně vypsány kurzy, které vyučují.

Po kliknutí na vybraný kurz se zobrazí jeho informační stránka. V levém sloupci se nachází seznam lekcí daného kurzu a přehled upozornění, která se k danému kurzu vztahují.

Učitel má nad seznamem upozornění k dispozici tlačítko pro vytvoření nového upozornění.



V pravé části jsou zobrazeny informace o daném kurzu. Učitelé zde mají k dispozici také tlačítka pro editaci. Levé tlačítko je pro editaci skupiny, pravé je pro přidání nové lekce.



Tato navigace provází učitele celou dobu při práci ve skupině, takže ji není těžké najít.

Editace skupiny

V rámci editace skupiny má učitel možnost měnit její název (zkratku) a případně vyučujícího učitele. POZOR – v případě, že změníte učitele, zmizí původnímu učiteli tato skupina z přehledu.

Další položkou v editaci je rozvrh dané skupiny. Pro přidání či změnu stačí kliknout do konkrétního políčka a vyvolat tak nabídku. Změnu uložíte kliknutím na modré tlačítko.

Rozvrh skupiny

	1	2	3	4	5
Po	zvolit	zvolit	zvolit	zvolit	zvolit
Út	zvolit	Vyberte učebnu			zvolit
St	zvolit	-- vyberte --			zvolit
Čt	zvolit	zvolit	zvolit	zvolit	zvolit
Pá	zvolit	zvolit	zvolit	zvolit	zvolit

Poslední položkou editace skupiny je správa studentů. Tato je rozdělena na dvě části. Těmi jsou vyhledávací pole (vlevo) a seznam zapsaných studentů vpravo.

Studenti

Pomocí rozbalovacího seznamu si zobrazíte všechny registrované studenty dané třídy. Následně je můžete přidat do své skupiny.

Nalezení

Zapsaní studenti

Pod formulářem se nachází tlačítko pro import studentů ze systému Bakaláři. Po kliknutí na toto tlačítko se zobrazí pole, do kterého stačí zkopírovat exportovaný seznam studentů dané skupiny. Tento export je potřeba provést ve tvaru „Třída Příjmení Jméno Třídní učitel“ kde oddělovačem je tabulátor. Po odeslání dat jsou v systému vyhledáni odpovídající studenti. Jelikož vyhledání probíhá na základě dat uvedených studentem při registraci, je možné, že někteří nemusí být nalezeni. Proto doporučuji seznam importovaných studentů zkontrolovat a chybějící studenty doplnit ručně.

Lekce

Po výběru lekce ze seznamu, jenž se na stránkách skupin nachází vlevo, je tato zobrazena v pravé části stránky.

Učitel má k dispozici tlačítko pro editaci lekce, které se nachází napravo od jejího nadpisu.

Editace lekce

Zde je třeba zdůraznit, že lekce není vázaná na skupinu, nýbrž na kurz (předmět). Z toho vyplývá, že jakákoli změna v lekci se projeví ve všech skupinách. S případnými změnami je tedy potřeba na toto myslet.

Editace lekce je rozdělena pomocí záložek na dvě části.

V rámci první záložky má učitel možnost upravit název lekce a její texty. Pod druhou záložkou se mu zpřístupní správa úkolů, přiřazených k dané lekci.

Správa úkolů

V této verzi aplikace jsou k dispozici tři typy úkolů:

- Textová odpověď
- Odevzdání souboru
- Odevzdání obrázku

Tyto typy mají společné parametry, kterými jsou název, zadání úkolu, kterým skupinám se má úkol zobrazovat a nastavení termínu odevzdání (deadline).

V souvislosti s odevzdáváním souborů či obrázků, může učitel specifikovat příponu souboru a jeho maximální velikost. Také si učitel může nastavit vlastní prefix souboru, který mu při exportu souborů pomůže se v množství prací orientovat.


V seznamu úkolů, který se nachází pod textem lekce, může učitel měnit individuální nastavení pro danou skupinu. Měnit lze pouze termín odevzdání. Tato změna se projeví pouze u skupiny, ve které tuto změnu provádíte.

Nastavit deadline pouze pro tuto skupinu:

<input type="text" value="dd.mm.rrrr"/>	<input type="text" value="hh:mm"/>	<input type="button" value="Uložit"/>
---	------------------------------------	---------------------------------------

Odevzdané práce

Jakmile začnou studenti odevzdávat úkoly, může je učitel ihned prohlížet. K tomuto slouží stránka, na kterou se dá dostat kliknutím na tlačítko „Odevzdané práce“. V závorce se nachází informace, kolik studentů ze skupiny již svou práci odevzdalo.


 Odevzdané práce (0)


Na této stránce má učitel k dispozici přehled všech studentů skupiny a barevně označeno, zda odevzdali včas, pozdě či vůbec. Nachází se zde také informace o celkovém počtu odevzdaných prací (užitečné v případě, že někteří studenti chybí) a také tlačítko pro export všech prací do zip souboru.

V případě úkolu s odevzdáváním obrázku je navíc k dispozici tlačítko pro vytvoření hodnotícího dotazníku .

Pod tlačítky jsou navíc zobrazeny náhledy všech prací včetně jmen studentů. Po kliknutí na obrázek se zobrazí zvětšenina obrázku. Mezi jednotlivými obrázky se lze pohybovat pomocí šipek.

Počet odevzdaných úkolů ve skupině G1: 1

Student	Stav	
Jirka Ančinec	Odevzdáno 17.5.2013 13:08	 pocivity-student-3.jpg

 Stáhnout ZIP

 Vytvořit dotazník



[Ančinec Jirka](#)

V rámci usnadnění práce učitele bylo do práce se soubory zakomponováno automatické pojmenovávání souborů. V případě nastavení prefixu je výsledný název souboru následující: „prefix-prijmeni-jmeno-id_souboru.pripona“. Díky tomuto odpadají neustálé problémy s pojmenováváním souborů, které studenti odevzdávají. Soubory jsou s tímto názvem již ukládány, takže v případě pozdější změny jména studenta se toto na názvu souboru neprojeví (toto je patrné na obrázku výše).

Hodnocení odevzdaných obrázků

Pro vytvoření hodnocení stačí kliknout na příslušné tlačítko. Toto tlačítko je k dispozici pouze u úkolů s nastaveným odevzdáváním obrázků. Po kliknutí na tlačítko se zobrazí formulář s nastavením dotazníku. Zde je možné vybrat či práce budou hodnoceny. Dále zda budou studenti hodnotit pouze pomocí bodů, či zda k obrázkům budou moci připojit i komentáře.

Následuje zadání povinného názvu a nepovinných textů. O jednotlivých polích je více napsáno v části věnované dotazníkům.

Po kliknutí na tlačítko „Vytvořit hodnocení“ je učitel nabídnut formulář pro odeslání pozvánek skupině. Když tohoto využije, bude všem studentům skupiny odeslán odkaz na dotazník (jiným způsobem tento dotazník nebude zpřístupněn).

Dotazníky

System dotazníků je přístupný pomocí tlačítka „Dotazníky“ v horní liště.



Na stránce je k dispozici přehled dotazníků. Vypsány jsou pouze ty, které přihlášený učitel vytvořil. U každého dotazníku jsou k dispozici následující možnosti.

 [Upravit](#)  [Pozvat](#)  [Výsledky](#)  [Smazat](#)

Tlačítko smazat po odsouhlasení kontrolního dotazu smaže z databáze dotazník, otázky, respondenty daného dotazníku i všechny jejich odpovědi. Je tedy potřeba si tento krok pořádně rozmyslet.

Vytvoření a editace dotazníku

Při vytváření dotazníku se zadávají stejné informace, jaké jsou při editaci k dispozici ve druhé záložce (podrobnosti). Patří sem povinný název a volitelné položky:

- Úvod – zobrazí se po spuštění nad dotazníkem.
- Základní informace – tyto se zobrazí na stránce, po kliknutí na odkaz k dotazníku.
- Poděkování – tento text se zobrazí po dokončení dotazníku.

V případě, že chcete omezit přístup k dotazníku, můžete zadat heslo. Dotazník je možné začít vyplňovat pouze v případě, že je aktivován.

Otázky

Otázky se nachází v první záložce editace. Správa otázek je rozdělena na dvě části. Vlevo se nachází seznam otázek, který je možné třídit přetáhnutím otázky na jinou pozici.

K přidání nové otázky použijte tlačítko „Přidat otázku“, které se nachází nad seznamem.

+ Přidat otázku

V pravé části se nachází prostor pro zobrazení formuláře. Formulář vypadá následovně:

Otázka *	<input type="text"/>
Typ *	<input type="text" value="Zaškrťovací pole"/>
Možnosti	<input type="text" value="Vlastní"/>
	Pro zadání "Vlastních" odpovědí vyplňte následující pole. Jednotlivé odpovědi pište do samostatných řádků .
Vlastní odpovědi	<input type="text"/>
Nápověda	<input type="text"/>

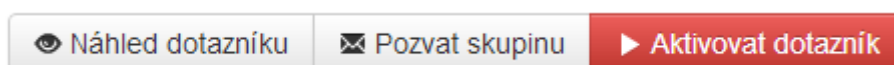
Do pole „Otázka“ patří samotné znění otázky.

Dále následuje Typ otázky, který ovlivňuje, jaké formulářové pole bude použito.

Možnosti (sady odpovědí) se vybírají v případě, že byl vybrán typ otázky s uzavřenými odpověďmi.

Vlastní odpovědi se zadávají v případě, že byly zvoleny „vlastní“ možnosti. Nápověda pomáhá respondentům lépe pochopit otázku.

Po dokončení přípravy otázek můžete využít jednu z možností:



Náhled dotazníku slouží k ověření, zda jste dotazník připravili správně.

Aktivace dotazníku umožní respondentům jej vyplnit. Pokud dotazník není aktivní, není možné jej vyplnit.

Pozvánky k dotazníku

Učitel má možnost rozeslat pozvánky skupinám, které vyučuje. Povinné položky jsou pouze dvě:

- Výběr skupiny, kam poslat pozvánku.
- Předmět zprávy.

Odkaz na dotazník bude do zprávy automaticky připojen.

Výsledky dotazníku

Na stránce s výsledky dotazníků má učitel následující možnosti:

Zobrazit a editovat respondenty pomocí tlačítka „Respondenti“.



Exportovat všechny odpovědi započítaných respondentů do csv souboru.



Zobrazit si grafy otázek s uzavřenými odpověďmi. V případě hodnocení obrázků se zobrazí 3 nejlépe hodnocené. Pomocí tlačítek může učitel měnit počet zobrazených obrázků.