Katedra informatiky Přírodovědecká fakulta Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Hlasovací systém



2017

Vedoucí práce: Mgr. Jiří Zacpal, Ph.D.

Josef Beták

Studijní obor: Aplikovaná informatika, prezenční forma

Bibliografické údaje

Autor:	Josef Beták			
Název práce:	Hlasovací systém			
Typ práce:	bakalářská práce			
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci			
Rok obhajoby:	2017			
Studijní obor:	Aplikovaná informatika, prezenční forma			
Vedoucí práce:	Mgr. Jiří Zacpal, Ph.D.			
Počet stran:	46			
Přílohy:	1 CD/DVD			
Jazyk práce:	český			

Bibliograhic info

Author:	Josef Beták
Title:	Voting system
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2017
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Jiří Zacpal, Ph.D.
Page count:	46
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Práce pojednává o webové aplikaci pro tvorbu online testů a hlasování. Představuje technické aspekty, dokumentaci pro správu a používání aplikace. Cílem práce bylo zjednodušit a zefektivnit práci učitelů na poli vzdělávání. Hlavními funkcemi aplikace jsou tvorba hlasování, testů a automatické vyhodnocení vyplněných testů.

Synopsis

The thesis work represents a web application that allows you to make online tests and vote. It consists of technical aspects and documentation for managing and using the application. The aim of thesis project is to make more efficient system and reduce the amount of work for professors. Main function of web application is voting, having online tests and automatically evaluate them.

Klíčová slova: hlasovací systém; webová aplikace; PHP; Framework Phalcon; Javascript; Ratchet; MySQL

Keywords: voting system, web application; PHP, Framework Phalcon; Javascript; Ratchet; MySQL

Děkuji Mgr. Jiřímu Zacpalovi, Ph.D. a Davidu Pokornému za rady a připomínky při vedení této práce. Dále bych rád poděkoval rodině a všem, kteří mě po celou dobu studia podporovali.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvo	od 8
	1.1	Zadání práce, vlastnosti a funkce aplikace
		1.1.1 Studentský účet
		1.1.2 Učitelský účet \ldots 10
2	Ana	lýza vybraných aplikací 11
	2.1	Socrative
	2.2	ClassMaker
3	Pou	žité technologie 13
	3.1	РНР
	3.2	Phalcon
	3.3	$MySQL \dots \dots \dots 14$
	3.4	XAMPP
	3.5	Javascript
	3.6	Ratchet
	3.7	Ostatní
4	Uži	vatelská dokumentace 16
	4.1	Dokumentace pro učitele
		4.1.1 Registrace a přihlášení
		4.1.2 Nastavení účtu
		4.1.3 Nápověda
		4.1.4 Domovská stránka
		4.1.4.1 Tabulka - Šablony testů \ldots \ldots \ldots 17
		4.1.4.2 Tabulka - Nezveřejněné testy
		4.1.5 Test a hlasování \ldots 19
		$4.1.5.1 \text{Test} \dots \dots \dots \dots \dots \dots \dots \dots \dots $
	4.2	Studentská dokumentace
		4.2.1 Neregistrovaný student
		4.2.1.1 Přihlášení a vyplnění testu/hlasování 23
		$4.2.2 \text{Registrovaný student} \dots \dots \dots \dots \dots \dots \dots 25$
		4.2.2.1 Registrace a přihlášení
		4.2.2.2 Domovská stránka $\ldots \ldots \ldots \ldots \ldots \ldots \ldots 25$
		4.2.2.3 Nastavení účtu
	4.3	Technická dokumentace
		4.3.1 Struktura projektu
		4.3.1.1 Nový projekt ve Phalconu
		4.3.1.2 Struktura Simplexu
		4.3.2 Struktura databáze
	4.4	Popis řešení vybraných partií Simplexu
		4.4.1 Ratchet server
		4.4.1.1 Instalace Ratchetu

	4.4.1.2	Struktura a spuštění serveru	36
	4.4.1.3	Serverová část	37
	4.4.1.4	Serverová část	37
	4.4.1.5	Teacher.js	38
	4.4.1.6	Student.js	38
4.4.2	Řešení za	bezpečení přístupu stránek	39
Závěr			40
Conclusions			41
A Nasazení	a testovái	ní aplikace	42
A Nasazení A.1 Požad	a testová avky	ní aplikace	42 42
A Nasazení a A.1 Požad A.2 Konfig	a testovár avky gurace	ní aplikace	42 42 42
A Nasazení a A.1 Požad A.2 Konfig A.2.1	a testovár avky gurace PHP fran	ní aplikace 	42 42 42 42
A Nasazení a A.1 Požad A.2 Konfig A.2.1 A.2.2	a testovár avky gurace PHP fran MySQL	ní aplikace	42 42 42 42 42
A Nasazení a A.1 Požad A.2 Konfig A.2.1 A.2.2 A.2.3	a testovár avky gurace PHP fran MySQL Simplex	ní aplikace 	42 42 42 42 42 42 43
 A Nasazení a A.1 Požad A.2 Konfig A.2.1 A.2.2 A.2.3 B Obsah při 	a testovár avky gurace PHP fran MySQL Simplex loženého	ní aplikace 	 42 42 42 42 42 42 43 44

Seznam obrázků

1	Domovská stránka učitele po přihlášení	11
2	Domovská stránka po přihlášení	12
3	Schéma architektury MVC	14
4	Navigace učitelského účtu	16
5	Tabulka s šablonami testů	17
6	Tabulka s ukončenými testy	18
7	Formulář nové otázky	19
8	Informační okénko pro návrat k poslednímu upravovanému testu .	20
9	Stránka s detaily aktivního testu	21
10	Detail neopraveného testu	21
11	Detail testu vybraného studenta	22
12	Formulář pro přihlášení do testu/hlasování	23
13	Příklad stránky pro vyplnění testu	24
14	Příklad stránky pro vyplnění testu	25
15	Mapa adresářů nového projektu	28
16	Kompletní mapa adresářů Simplexu	32
17	Relační schéma databáze	34
18	Kompletní struktura Nudge serveru	36

Seznam tabulek

Seznam vět

1	Definice (Framework)	26
2	Definice (Webový socket)	26
3	Definice (JSON)	26
4	Definice (Instance)	26
5	Definice (Phalcon plugin a komponenta)	26
6	Definice (jQuery plugins)	26

Seznam zdrojových kódů

1	JSON	
2	РНР	
3	РНР	
4	РНР	
5	JS	
6	JS	
7	JSON	
8	PHP	

1 Úvod

Testování znalostí studenta za pomoci papírových testů je běžnou záležitostí, ale velice málo rozšířený způsob je testovat studenta za pomoci účelové aplikace. Jednou z nepopiratelných výhod takového způsobu testování je fakt, že lze testy automaticky a okamžitě vyhodnotit, což usnadňuje práci a šetří čas nejen učiteli, ale i samotným studentům. Učiteli by měla být také umožněna kontrola nad testem. To znamená například ukončit probíhající test studentovi, který jeví známky podvádění nebo sledovat, kdo aktuálně daný test vyplňuje. Taková míra kontroly by měla být doplněna o možnost přístupu k testu pouze přes daný účet či speciální kód (klíč) a nebo nejlépe pomocí obojího dohromady. Posledním důležitým faktorem je možnost archivovat výsledky testů.

1.1 Zadání práce, vlastnosti a funkce aplikace

Bakalářská práce byla vypsána ve spolupráci s firmou OLC Systems, jejímž zástupcem byl v tomto případě David Pokorný. Cílem bylo vytvořit webovou aplikaci založenou na PHP frameworku Phalcon (význam viz Definice 1). Aplikace měla učitelům umožnit vytvářet jednoduché testy a hlasování. Systém měl být svižný a triviální na používání. Dále pak celá aplikace neměla být závislá na internetovém připojení.

Výstupem dříve zmíněného hlasování je procentuální statistika odpovědí studentů v reálném čase (učitel v prohlížeči sleduje průběh, bez nutnosti obnovy stránky). Dalším hlavním požadavkem byla použitelnost aplikace na mobilních platformách. Místo programování separované mobilní aplikace bylo po konzultaci se zástupcem firmy, panem Davidem Pokorným, zvoleno responsivní provedení webové aplikace. Výsledná aplikace byla pojmenována Simplex a dále v textu bude místo termínu **webová aplikace**, používán přímo její název. Simplex neměl být cílený pouze pro školství, ale také například pro školení zaměstnanců firmy apod.

V následujících odstavcích bude stručně nastíněny funkce Simplexu. Podrobněji budou rozebrány v kapitole Uživatelská dokumentace. Simplex rozlišuje 2 základní typy uživatelských účtů – učitel, student. Nejprve budou rozebrány funkce studentského účtu.

1.1.1 Studentský účet

Simplex dělí účty na *registrovaný* (student má založený účet) a *neregistrovaný* (student nemá založený účet). Oba typy studentů se do aktivního testu/hlasování přihlašují pomocí speciálního klíče, který je unikátní pro každý test/hlasování. Klíč není možné získat jinak než od učitele.

- Neregistrovaný student je po zadání správného klíče testu/hlasování vyzván k zadání jména (v případě že test/hlasování nebylo nastaveno jako anonymní)
- *Registrovaný student* logicky žádné jméno zadávat nemusí a oproti *neregistrovanému účtu* má funkce navíc, například možnost zobrazit si úspěšnost vyplněných testů nebo přidat na neopravený test hlídání. To znamená, že mu budou na e-mail zaslány informace o tom, jestli byl test již opraven

Logika dvou studentských účtu, *registrovaného* a *neregistrovaného*. Byla implementována pro možnost rychlého a jednorázového přístupu. Pro představu je zde uvedena ukázková situace. Student je na kurzu - *Základy počítačové gramotnosti*. Jeho školitel využívá aplikaci Simplex pro ověření znalostí studentů na konci kurzu. Student, který ovládá pouze základy informatiky, je zdržován nutnostmi, jako je založení účtu pouze pro tento jeden kurz. V případě, že by měl kurzů více, by si nemusel pamatovat heslo a musel by zažádat o jeho obnovení (což není vůbec praktické). Dále by byly zapotřebí další kroky navíc, které by student vícekrát nevyužil.

1.1.2 Učitelský účet

Nejdůležitějšími funkcemi pro učitelského účtu jsou vytváření, spouštění a správa jednotlivých testů/hlasování. Možnosti tvorby otázek a odpovědí jsou v Simplexu jednotné. To znamená jak v testu, tak v hlasování lze vytvořit otázky s jednou nebo více odpovědmi (v dalších verzích je plánováno přidání textové odpovědi). Jak už bylo v textu zmíněno, Simplex je aplikace pro tvorbu testů a hlasování. Proto by bylo vhodné představit jejich rozdíly a vlastnosti.

- Test slouží k bodovému ohodnocení znalostí studenta. Výstupem je počet bodů udávající úspěšnost podle stanovené stupnice. Učitel tvoří takzvané šablony testů, z kterých je možné spustit libovolné množství testů, každé s vlastním klíčem. Šablona obsahuje otázky, odpovědi, body pro dané otázky, bodovou stupnici známek, název, popis apod. Pro spuštění testu dle šablony je nutné nastavit nebo vygenerovat unikátní klíč k přihlášení, datovou platnost, časové omezení a označení, zda se jedná o anonymní test či nikoli. Výsledky testu se archivují a po ukončení nabývají statusu neopravený, nezveřejněný (test je opravený, ale registrovaný student si nemůže zkontrolovat výsledky) a zveřejněný(test je opravený a registrovaný student může přistoupit k výsledkům daného testu, ne však ke správným odpovědím). Učitel má možnost také prohlížet jednotlivé otázky ukončených testů a jejich případné odpovědi u vybraných studentů. Učitel může také exportovat výsledky do formátu .xls.
- Hlasování slouží k průzkumu názorů v daném čase, výstupem pro každou otázku je informace, kolik studentů hlasovalo pro danou odpověď. Informace je vyjádřena v procentech. Výsledky hlasování se neukládají, není nutná ani tvorba šablony.

2 Analýza vybraných aplikací

V úvodní podkapitole Zadání práce, vlastnosti a funkce aplikace, byly vysvětleny základní funkce Simplexu. Nyní budou zanalyzovány vybrané aplikace se stejným účelem. Podobných aplikací na trhu je mnoho, proto byly vybrány pouze dvě. První z nich je *Socrative*, ze kterého byla čerpána valná část inspirace, proto bude rozebrán podrobněji. Druhá bude komerční aplikace *ClassMarker*, kterou využívá i firma *Google* (dle referencí na jejich webových stránkách).

2.1 Socrative

Jde o webovou aplikaci pro tvorbu online testů. Od roku 2016 ji lze zdarma využívat jen omezeně a přibyla možnost zpoplatněné verze s názvem Pro, která přidává některé rozšiřující funkce. V následujícím textu budou popsány pouze funkce, které nabízí verze zdarma. Vzhled aplikace po přihlášení lze vidět na obrázku číslo 1.



Obrázek 1: Domovská stránka učitele po přihlášení

Učitel zde tvoří kvízy (analogie testu), do nichž mohou být přidány otázky typu více odpovědí, pravda/lež (pouze možnost vybrat ano/ne) a textová odpověď. Vytvořené kvízy lze pak spustit s individuálním nastavením. Je možné vybrat z 3 typů nastavení okamžitá odezva, otevřená navigace a řízeno učitelem. Každý z vyjmenovaných typů pouze udává výchozí nastavení pro daný kvíz (některé typy mají zakázána určitá nastavení). Ve zkratce může být nastaveno požadování jména, náhodné pořadí otázek nebo odpovědí, zobrazení správné odpovědi po zodpovězení otázky a poslední možností je zobrazení výsledné úspěšnosti na konci kvízu. K takto spuštěnému kvízu se lze připojit pomocí unikátního klíče (symbolizující třídu ve škole). Průběh testu může učitel sledovat v záložce **Výsledky**, kde se kromě jmen studentů zobrazují i vybrané odpovědi. Pro účel hlasování má učitel možnost vytvořit kvíz pouze s jednou odpovědí, bez možnosti přidat text k otázce či odpovědi. Ukončené kvízy se archivují a v případě úpravy se nezmění výsledky kvízů, které již proběhly.

Specialitou v Socrative je možnost spustit kvíz v režimu *Space Race*, v překladu vesmírný závod. Učitel nastaví počet týmů a student si po přihlášení do kvízu, vybere jeden z nich. Učitel pak může promítat grafický průběh textů symbolizovaný ikonami a progres barem odrážející úspěšnost týmu.

Webové stránky: https://www.socrative.com/

2.2 ClassMaker

Profesionální webová aplikace pro tvorbu kvízů určená pro vzdělání i podnikání. Pyšní se řadou významných klientů jako *Google, Facebook, Microsoft, Amazon* a jiné. Aplikaci je možné vidět na obrázku 2.

Deabhraid	11-1-1-			~
Dashboard V	Help 🗸		My Account 🔻	q
Dashboard			Tests Gr	oups Links
Recent Tests Taken	Recent Group Results Recent	Link Results		
You have no Recer	nt Test results to display here at p	resent.	My Data [©]	
You can also access	Test results under the Tests section.	0	Tests	2
			Questions	2
	How ClassMarker	Works	Categories	2
			Files	0
			Certificates	0
	οO		Broups	1
	\rightarrow $$	\rightarrow	Results	0
	P		e Links	0

Obrázek 2: Domovská stránka po přihlášení

Aplikaci je možné používat zdarma s omezeními, ale pro potřeby pravidelného používání je nutné přejít na placenou licenci. Pomocí aplikace lze vytvářen nejen kvízy, ale také certifikáty či vybrané kvízy vytisknout. Studenti se do spuštěného testu přihlašují pomocí účtu, který jim byl vytvořen učitelem anebo pomocí speciálního kódu. Student je pak vyzván k zadání přihlašovacých údajů. Zpřístupnit kvíz širší veřejnosti lze za pomocí internetového odkazu, který lze snadno rozeslat například e-mailem. Ve kvízu může učitel použít různá nastavení od rozmezí platnosti přihlášení, přístupového hesla, časového limitu až po možnost platby přes PayPal za přístup ke kvízu.

Webové stránky: https://www.classmarker.com/

3 Použité technologie

V následujících kapitolách budou popsány použité technologie, které byly vybrány v souladu se zadáním.

3.1 PHP

Název PHP je rekurzivní zkratkou pro anglické spojení *Hypertext Preprocessor*, které v češtině znamená Hypertextový preprocesor. Jazyk PHP patří do kategorie skriptovacích jazyků a je primárně určen pro programování dynamických internetových stránek, kde je jeho využití nejrozšířenější (v roce 2014 činil odhad 82%). Kód je vykonáván na straně serveru a k uživateli je přenášen až výsledek, nejčastěji ve formátu HTML. Jazyk PHP je nezávislý na platformě a lze ho tedy volně přenášet mezi jednotlivými systémy. V současné době je v jazyku PHP naprogramováno nespočet knihoven pro účely od zpracování textu, práci se soubory, grafiky až po přístup k různorodým databázovým systémům. PHP podporuje celou řadu internetových protokolů – HTTP, SMTP, SNMP, FTP, IMAP, POP3 atd.[1]

V projektu je použita stabilní verze 5.6.8. a není plánován přestup na novější verzi, kvůli problémům s kompatibilitou.

3.2 Phalcon

Jedná se o open-source¹ framework pro jazyk PHP. Kvůli optimalizaci pro výkon je napsán jako rozšíření v jazyku C, kód samotné aplikace se píše v jazyce PHP. Framework je postaven na softwarové architektuře MVC[15] (Model View Controller) viz obrázek 3. MVC se skládá ze tří částí. [11]

- Model (model) slouží pro abstrakci dat, s nimiž aplikace pracuje (například databázové tabulky)
- View (pohled) pro prezentaci dat uživateli (například HTML stránka)
- Controller (řadič) reaguje na události (typicky od uživatele) a na jejich základě manipuluje s částmi Model a View



Obrázek 3: Schéma architektury MVC

3.3 MySQL

MySQL[9] je relační databázový systém, populární především v oblasti webových technologií a tím pádem i v jazyce PHP. Velmi rozšířeným se stal zásluhou lehké nastavitelnosti, spolehlivosti a rychlosti. Systém je multiplatformní a pro práci s relačními daty používá strukturovaný dotazovací jazyk SQL[16] (Structured Query Language).

Výběr databázového systému nebyl nijak svazující, protože Phalcon díky technologii MVC je schopen pracovat i s dalšími populárními databázovými systémy jako například *PosgreSQL*. Technologie byla vybrána z důvodu dřívějších zkušeností. Pro správu databáze z hlediska dat byla využit *PHPMyAdmin*[17], pro návrh a úpravy databázových struktur (tabulek, sloupců, relací) byl použit nástroj *MySQL Workbranch*[12].

 $^{^1\}mathrm{software}$ licence dovolující upravovat a distribu
ovat kód

3.4 XAMPP

Jedná se o multiplatformní open-source nástroj poskytující webový server *Apache* a v základu i technologie pro podporu *MySQL*, *PHP* a *Perl*. Kromě vyjmeno-vaných technologií lze přidat i jiná rozšíření jako například PHP Framework Phalcon.

XAMPP[10] byl vybrán kvůli jeho jednoduchosti a menší náročnosti oproti konkurenčnímu WAMP serveru a také předchozím zkušenostem. [10]

3.5 Javascript

Multiparadigmatický, dynamicky typovaný, skriptovací jazyk jehož autem je Brendan Eich. Jak by mohl název mást, Javascript nemá z technického hlediska nic společného s jazykem Java, šlo pouze o marketingový tah v době, kdy byla Java populární.

Kód Javascriptu může být stažen v samostatném souboru nebo být vložen přímo do HTML kódu stránky. Pomocí interpretu je pak kód zpracováván na straně klienta. Standardně podporován většinou prohlížečů bez nutnosti externí instalace. V současnosti se jedná o nejrozšířenější jazyk pro dynamickou manipulaci s prvky webových stránek.

Javascript umí mnohé, ale nemůže například přistupovat k souborům klienta či nějakým způsobem sabotovat klientský systém. [2]

3.6 Ratchet

Nezávislá PHP knihovna (technicky se jedná o samostatnou aplikaci) pro obousměrnou komunikaci klient-server v reálném čase skrze Webové Sockety (viz Definice 2). Pomocí knihovny Ratchet se pak dají vytvořit například internetové chaty, online hry apod.

Ratchet byl původně vybrán kvůli představě, že by bylo možné propojit ho přímo s aplikací. Bohužel představa se ukázala býti špatná. Díky faktu, že se jedná o nezávislou aplikaci, bude v budoucnu nahrazen konkurenční technologií NodeJS². [3]

3.7 Ostatní

- JQuery multifukční Javascriptová knihovna obsahující optimalizované funkce pro práci s webovými prvky a snadnější práci s technologií AJAX³
 [4]
- AJAX technologie pro tvorbu asynchronních webových aplikací[5]
- Bootstrap responzivní CSS knihovna [6]

²vývojové prostředí umožnící psát webové servery ³technologie pro psaní asynchronních webových aplikací

4 Uživatelská dokumentace

Tato kapitola slouží jako manuál pro uživatele. Rozdělena je na dvě základní podkapitoly, v první bude představena základní práce se Simplexem z pohledu učitele, krátce poté z pohledu studenta.

Na úvod dokumentace je nutné podotknout, že pokud prohlížeč uživatele nebude podporovat minimální technologické požadavky pro používání Simplexu. Pak jej aplikace přesměruje na chybovou stránku, obsahující odkazy na podporované prohlížeče.

4.1 Dokumentace pro učitele

4.1.1 Registrace a přihlášení

Pro používání Simplexu jako učitel je v první řadě nutné provést registraci. Jeden ze standardních způsobů jak se k registraci dostat je přes odkaz **Učitel** na hlavní stránce aplikace. Uživatel je potom přesměrován na formulář s přihlášením, kde je mimo jiné odkaz na registraci. Po řádném vyplnění údajů e-mail, heslo, jméno a jazykovém nastavení aplikace se může učitel přihlásit s vybraným heslem a emailem. Po úspěšném přihlášení je pak přesměrován na hlavní stránku svého účtu. Na obrázku číslo 4 je vyobrazeno záhlaví aplikace přihlášeného uživatele.



Obrázek 4: Navigace učitelského účtu

Uživatelský panel dává informaci o celém jménu učitele, které se bude zobrazovat i studentům po přihlášení do testu/hlasování. Také skrývá odkazy na stánky s nápovědou, nastavením účtu a tlačítko pro odhlášení. Navigační panel pak obsahuje hlavní odkazy aplikace pro přechod mezi stránkami.

4.1.2 Nastavení účtu

Na stránce nastavení účtu si lze kromě hesla, e-mailu a celého jméno změnit i jazykovou variantu aplikace (dostupná je čeština a angličtina). Aktualizovaný email samozřejmě nesmí být existovat a heslo nelze zadat prázdné. Při změně hesla si lze text napsaný do kolonky zobrazit podržením bočního tlačítka **Zobrazit**.

4.1.3 Nápověda

Obsahuje všechny potřebné informace od přihlášení po tvorbu a úpravu samostatných testů/hlasování. Stránka s nápovědou obsahuje stručnou uživatelskou dokumentaci.

4.1.4 Domovská stránka

Stránka obsahuje kromě navigace také dvě důležité tabulky a to s šablonami testů a nezveřejněnými testy. Účel obou tabulek budou vysvětleny později.

4.1.4.1 Tabulka - Šablony testů

Tabulka obsahuje informace o existujících šablonách testů. Tabulka disponuje filtrem, tlačítko se nachází v pravém horním rohu tabulky. Kliknutím na tlačítko je možné zobrazit/skrýt možnosti filtrování. Šablony lze pak filtrovat pomocí podnázvů nebo štítků. Štítky podávají informaci o kontextu daného testu, například pro přírodopisný test o rostlinách bychom mohli zvolit štítky *Přírodopis, rostliny, IT2B*. Vymazat nastavení filtru lze pak pomocí tlačítka Vyčistit. Z obrázku číslo 5 lze vidět příklad použití filtru a jeho výsledeku.

Řetězcová filtrace	Tlačítko filtru 🔶 🕇
=iltry	Šítková filtrac
Podřetězec názvu	Přírodověda X
Vyhledat	Menu akcí šablony
Název	Таду
Přírodověda - rostliny a živočichové	Přírodověda 28

Obrázek 5: Tabulka s šablonami testů

Každý záznam o šabloně v tabulce má navigaci s tlačítky.

- Spustit uživatel je přesměrován na stránku pro spuštění testu, kde je možné nastavit různorodé vlastnosti nového testu a také přístupový klíč.
- Editovat přesměruje uživatele na stánku s editací otázek a detailů testu (editovat je možné pouze šablony, dle nichž nebyl spuštěn test).
- Odstranit odstraní záznam o šabloně.

Pokud tabulka obsahuje víc jak 3 záznamy, zobrazí se stránkování tabulky.

4.1.4.2 Tabulka - Nezveřejněné testy

Tabulka s neveřejnými testy obsahuje záznamy (pokud tabulka obsahuje více jak 3 záznamy, zobrazí se stránkování tabulky) o testech, které byly ukončeny automaticky nebo manuálně učitelem. Test je pak ve stavu *neopraven*, a přes tlačítko **Detail testu** se uživatel dostane na stránku, kde je možné test opravit. Opravený test přejde do stavu *nezveřejněn*. Stav nezveřejněn znamená, že výsledky testů nejsou zveřejněny studentům, kteří se do testu přihlásili skrze svůj účet. Důvod je prostý, někdy potřebuje učitel zveřejnit výsledky až ve chvíli, kdy všechny skupiny studentů napíší test. Zveřejnit test lze pak na stránce s **Detailem testu**. Na obrázku 6 pak lze vidět příklad testu v nezveřejněném stavu.

	Hacitko "Deta	ili testu
Vázev	Stav	X
Přírodověda - rostliny a živočicho	VÉ Nezveřejněn	0

Obrázek 6: Tabulka s ukončenými testy

4.1.5 Test a hlasování

V téhle podkapitole bude vysvětleno, jakým způsobem lze vytvořit šablonu testu, spravovat samotný test a jak vytvořit hlasování.

4.1.5.1 Test

Vytvoření nové šablony

Novou šablonu založíme pomocí tlačítka **Vytvořit** umístěné v navigačním panelu, kde po rozkliknutí zvolíme možnost **Test**. Uživatel je přesměrován na stránku pro tvorbu testu, která se skládá ze dvou kroků.

Tvorba otázek

Pomocí tlačítek pod navigací lze do šablony přidávat otázky. Implementovány jsou dva typy otázek *výběr jedné odpovědi* a *výběr více odpovědí*. Po přidání vybraného typu otázky se pak formulář s otázkou otevře automaticky pro úpravu. Obrázek s číslem 7 vyobrazuje formulář otázky – *výběr jedné odpovědi*, ve stavu editace.



Obrázek 7: Formulář nové otázky

V editačním stavu lze pak upravovat texty otázek a odpovědí (formátovat vybraný text), přidat novou odpověď nebo vybranou odpověď odstranit. Správná odpověď se pak vybírá dle příslušných prvků po levé straně otázky. Spodního iterátor slouží pro přidělení bodů otázky (počet bodů může nabývat pouze hodnot přirozených čísel). Uložení otázky lze provést stiskem žlutě podbarvené tlačítko s ikonou diskety, v pravém horním rohu formuláře. Bez řádného uložení není možné přejít na další otázku.

Informace o testu

Posledním krokem k úspěšnému vytvoření šablony je doplnění informace o šabloně testu jako název, popis a případně přidání relevantní štítky. Kromě zmíněných informací se nastavuje známkování a hranice neúspěšnosti. Hranice určuje, od které známky jsou ty následující hodnoceny jako nedostatečné. Každý záznam by měl mít vyplněny kolonky s minimálním a maximálním počtem bodů a následně pak název známky.

Nová šablona se ukládá do dočasné paměti prohlížeče, v případě přerušení například při zavření prohlížeče se k úpravě šablony lze vrátit z domovské stánky viz 8. obrázek.

Posledně jste nedokončili tvorbu testu, chcete se k ní vrátít?			👗 Mgr. M	 Topoláne
	Posledně iste nedokr	ončili tvorbu testu, chcete	se k ní vrátít?	
	osicune jste neuolit			

Obrázek 8: Informační okénko pro návrat k poslednímu upravovanému testu

Editace šablony

Editaci je možné provést u šablon, které nebyly vzorem pro žádný test. K editaci je možné přistoupit pomocí rozkliknutí tlačítka akcí v tabulce s šablonami testů. Samotná editace pak funguje podobně jako tvorba nové šablony. Editovat šablony dle niž byl spuštěn test nelze kvůli možné změně dat již hotových testů.

Spuštění testu podle šablony

Vygenerovat test podle vybrané šablony lze pomocí tlačítka spustit v tabulce Šablony testů. Po kliknutí je učitel přesměrován na formulář možných nastavení pro nový test. Každý nový test má v systému svůj unikátní klíč, učitel si může vymyslet svůj (musí mít alespoň 3 znaky) anebo si pomocí tlačítka vedle pole pro klíč vygenerovat náhodný. Testu lze pak nastavit platnost (od kdy a do kdy se k němu bude možné přihlásit), časový limit (počet minut, které má student k dispozici k vyplnění testu) a jestli se jedná o test anonymní (student nebude vyzván k napsání jména).

Sledování průběhu testu

K sledování aktivního testu se učitel dostane přes navigační menu, po rozkliknutí možnosti **Aktivní**. V jednu chvíli může být aktivních více testů a každý lze sledovat nezávisle.

Jak je z obrázku číslo 9 patrné, lze ukončit celý test manuálně nebo pouze odpojit vybrané studenty v případě, že by učitel zjistil náznak podvodu.

	Tabulka studentů	Počet piřhlášených studnetů 🔶 🔺 6
Test - Přírodověda - rostliny a živočichové		Ukončit instanci
	# Přezdivka	Stav
Ne © 60 min.	1 Hana Poláková	Odpojen
di 29-12-2016 18:45 / 29-12-2016 19:55	2 Petr Novák	Odpojit
₹.	3 Josef Beták	Odpojen Odpojit
Nastavení testu	4 Tomáš Skřivan	Odpojen Odpojit
	5 Apolenka Veronová	Odpojen Odpojit
	6 Petr Zaoral	Připojen Odpojit
		~

Obrázek 9: Stránka s detaily aktivního testu

Detail ukončeného testu

Opravit a následně zveřejnit test lze na stránce detailu testu (jedná se o identickou stránku jako v případě sledování průběhu testu), na kterou uživatel může přejít přes tabulku nezveřejněných testů. Automatickou opravu zde provádí tlačítko **Automatická oprava**, viz obrázek 10.

nformace				Automaticky opravit
Test - Přírodověda - rostliny a	#	ldentifikační jméno studenta		▲0 ▲6 ▲6
živočichové	1	Hana Poláková	Neopraven	0
SqGoAS ₩ Ne # Ne	2	Petr Novák	Neopraven	0
29.12.2016 / 29.12.2016	3	Josef Beťák	Neopraven	0
Zveřejnit test	4	Tomáš Skřivan	Neopraven	۲
Po zveřejnění se výsledné známky testu zveřejní pro dané studenty	5	Apolenka Veronová	Neopraven	0
	6	Petr Zaoral	Neopraven	0
Stupnice známek				

Obrázek 10: Detail neopraveného testu

Pod tlačítkem **Automaticky opravit**, se nachází informace sdělující kolik testů je opravených, neopravených a celkový počet studentů. Tlačítko **Automaticky opravit** by se mohlo zdáti zbytečné, ale je tu z důvodu plánované implementace textové otázky. Po opravě přejde test ze stavu *Neopravený* do stavu *Nezveřejněný*, až po zveřejnění student (v případě že se přihlásil přes účet v aplikaci) uvidí své výsledky.

Kliknutím na tlačítko **Zobrazit výsledky** (ikonka oka na libovolném řádku se záznamem studenta) je učitel přesměrován na podrobné výsledky testu vybraného studenta.

Z obrázku s číslem 11 je patrné, že levá strana je určena pro statistické výsledky. Pravá pak obsahuje kompletní informace o odpovědích studenta. Modrý pruh v levé části znamená, že daná odpověď byla vybrána studentem. Pravá část pak odhaluje, jestli se jednalo o správnou/špatnou odpověď. Správné výsledky studentovi nejsou poskytnuty, student na svém účtu vidí pouze vlastní skóre.

L Hana Poláková	1 2 3 4 5 6	
Úspěšnost		Body:
	Je velryba savec?	
	Ano	
67% 20/30.b	O Ne	

Obrázek 11: Detail testu vybraného studenta

4.2 Studentská dokumentace

Dokumentace určená pro studenty se dělí na dvě základní podkapitoly a to pro neregistrované a registrované studenty. Neregistrovaný student potřebuje k přihlášení do testu/hlasování klíč a v případně, že není test/hlasování anonymní, musí zadat své jméno. Registrovaný student potřebuje (po přihlášení ke svému účtu) znát pouze klíč. Výsledky testů (nikoli hlasování) pak registrovaný student zjistí na svém účtu.

4.2.1 Neregistrovaný student

4.2.1.1 Přihlášení a vyplnění testu/hlasování

Student bez účtu se může k testu/hlasování přihlásit přes odkaz na domovské stránce po rozkliknutí možnosti **Student** v navigačním menu stránky a vybráním první možnosti **Neregistrovaný**. Student je pak přesměrování na stánku s formulářem, kde může zadat klíč testu/hlasování. V okamžiku správného zadání klíče se objeví formulář s detailem testu/hlasování viz obrázek 12.

	Přihlášení
Přírodověda ▲ Mgr. M. Topolánek @ 60 Min. @ 2016-12-30 15:00:00 Vložte své jméno	 rostliny a živočichové
Například Hanka Polová	
	Zpět
	Připojit

Obrázek 12: Formulář pro přihlášení do testu/hlasování

Pokud se formulář neobjevil, ale místo toho se zobrazilo chybové hlášení, pak mohla nastat jedna z následujících možností.

- Klíč <klíč> neexistuje
- Test/hlasování už není aktivní

Po zadání jména je student přesměrován přímo na stránku s otázkami testu/hlasování, příklad formuláře testu vidíte na obrázku 13. V případě časově omezeného testu je v záhlaví vidět časovač (pokud je platnost testu/hlasování také omezena pak se časový limit může zkrátit o rozdíl přesahující čas platnosti), záhlaví také obsahuje tlačítko pro ukončení testu. V zápatí stránky se pak nachází stránkování.

Ukončit test				Ę	9 Min. 49 Sek.
	#1 Je velryba s	savec ?		5 bodů	
	۲	Ano			
	0	Ne			
	#2			5 bodů	
	Ropucha p	atří mezi			
	0	Savce			
	0	Ryby			
	۲	Obojživelníky			
	#3			5 bodů	
	Vyberte vše	chny savce			
←Zpět			1/2		Další →

Obrázek 13: Příklad stránky pro vyplnění testu

4.2.2 Registrovaný student

4.2.2.1 Registrace a přihlášení

Student se může registrovat po vyplnění registračního formuláře, na který se dostane z domovské stránky Simplexu. Po rozkliknutí možnosti **Student** v navigačním panelu a výběru možnosti **Registrovaný** bude uživatel přesměrován na stránku s formulářem pro přihlášení, kde je mimo jiné i odkaz na formulář s registrací. Po registraci se student může pomocí zadaných údajů (e-mail a heslo) přihlásit k nově vytvořenému účtu.

4.2.2.2 Domovská stránka

Úspěšně přihlášení student je přesměrován na domovskou stránku svého účtu, která je vyobrazena na obrázku 14.



Obrázek 14: Příklad stránky pro vyplnění testu

Z obrázku je patrné, že k novému testu se může registrovaný student přihlásit pomocí formuláře na pravé straně. Po úspěšném zadání klíče se objeví formulář s detailem testu/hlasování podobný jako na obrázku s číslem 12. S tím rozdílem, že zde nebude kolonka pro zadání jména.

Test se po vyplnění přidá do tabulky *Výsledky testů*. V tabulce je informace o názvu, datu přihlášení a také výsledném skóre. Pomocí tlačítka **Zobrazit výsledk** (ikonka grafu) se na místo tabulky vykreslí výsledná známka a skóre v testu. V případě, že se ve sloupci *prospěch* nachází znak pomlčky, jedná se o neopravený nebo nezveřejněný test. Na takový test lze nastavit **hlídání**, díky kterému bude po zveřejnění výsledků poslán informační e-mail. Data v tabulce jsou seřazena sestupně podle data přihlášení. Pokud je v tabulce více jak 5 záznamů, pak se objeví panel pro stránkování.

Uživatelský panel nabízí odkazy pro odhlášení, nápovědu a nastavení účtu.

4.2.2.3 Nastavení účtu

Kromě změny základních údajů (e-mail, celé jméno nebo heslo studenta) lze také upravit jazykovou variaci účtu (dostupná je česká a anglická).

4.3 Technická dokumentace

V této kapitole bude proveden podrobnější rozbor použitého frameworku Phalcon a základní struktury Simplexu. Čtenář bude obeznámen se strukturou projektu, databází, Ratchet serverem (technologie zajišťující real-time komunikaci) a základními implementačními rysy Simplexu. Na závěr kapitoly budou popsány postupy řešení vybraných problémů při implementaci Simplexu.

Definice 1 (Framework)

Bálíček podpůrných knihoven, programových řešení, návrhových vzorů apod. pro vývoj a organizaci jiných projektů.

Definice 2 (Webový socket)

Webový socket[7] přichází jako novinka s HTML 5[14], jedná se o technologii umožňující navázat obousměrné spojení se serverem a po úspěšném spojení si vyměňovat informace v reálném čase. Což znamená, že není nutné používat techniku *hearbeater*, kdy se klient v pravidelných intervalech dotazuje na server.

Definice 3 (JSON)

Datový formát určený pro přenos dat nezávisle na platformě. Data jsou agregována v polích nebo objektech, organizována pak v pořadí klíč a hodnota. Více hodnot je pak oddělených čárkou. Vstupem může být libovolná datová struktura (číslo, pole, objekt, řetězec apod.) výstupem je pak vždy řetězec. [13]

```
1 {
2 "jmeno": "Petr Brambora",
3 "pocetDeti": 2,
4 "svobony": false,
5 "jmenaDeti": ["Apolena", "Pepa"]
6 }
```

Zdrojový kód 1: JSON

Definice 4 (Instance)

Nechť slovo instance bude v textu vyjadřovat test nebo hlasování anebo obojí zároveň.

Definice 5 (Phalcon plugin a komponenta)

Plugin ve Phalconu by se dal označit jako *listener*⁴, který implementuje metody a události některých komponent frameworku. Komponenta je pak třída nebo skupina tříd implementující určitou funkcionalitu.

 $^{^4\}mathrm{procudura}$ čekající na nějakou událost, například kliknutí na tlačíko

Definice 6 (jQuery plugins)

jQuery pluginy jsou menší knihovny psané za pomocí Javascriptové knihovny jQuery, které přidávají do webové aplikace novou funkcionalitu na straně klienta. Například knihovna pro vykreslení kalendáře, ve kterém je možné vybrat datum a čas.

4.3.1 Struktura projektu

Pro lepší pochopení je vhodné v prvé řadě popsat strukturu nového projektu a poté (protože Simplex obsahuje jisté úpravy) bude představena struktura samotného Simplexu, bez nutnosti rozebírat základních principů, které budou obsaženy v následující podkapitole.

4.3.1.1 Nový projekt ve Phalconu

Základní struktura nového projektu ve Phalconu se skládá ze dvou základních adresářů, adresáře *public* a *app*. Detailní struktura každého z adresářů je vidět na obrázku 15.



Obrázek 15: Mapa adresářů nového projektu

Složka *public* obsahuje zdrojové soubory externích technologií v našem případě kaskádové styly (složka *css/*) a Javascriptové skripty (sloužka *js/*). Do složky se také ukládají například vygenerované soubory z aplikace nebo obrázky. Adresář *public* je přístupný z vnějšku aplikace, zatímco adresář *app* nikoli. Složka *public* také obsahuje nejdůležitější soubor celé aplikace a to **index.php**, který na počátku inicializuje služby a potřebné závislosti aplikace.

Složka *app* není zvenčí přístupná a obsahuje zdrojové kódy aplikace. Na obrázku 15 jsou vidět 3 základní podadresáře **Controllers**, **Models** a **Views**, které odráží rozdělení podle architektury **MVC**. Vazby mezi jednotlivými částmi lze vidět na obrázku 3.

Controllers

Složka obsahuje třídy s metodami pro zpracování a vygenerování požadavku klientovy takzvané *Controllers*, česky "řadiče". V textu budou použity originální názvy.

Každá stránka aplikace má svůj vlastní Controller, který musí být potomkem základního controlleru frameworku - $Phalcon \setminus Mvc \setminus Controller$. Název vlastního Controlleru (stejně jako souboru) se pak zpravidla skládá z názvu stránky a postfixu Controller.

Takže například Controller pro stránku *www.simplex.cz/student/prihlaseni* by se jmenoval *StudentController* (soubor by pak měl příponu *.php*). Takhle vy-tvořený Controller pak musí, obsahoval veřejnou metodu *prihlaseniAction()*. Soubor Controleru by pak mohl, vypadal následovně.

```
1 <?php
2 use Phalcon\Mvc\Controller;
3 class StudentController extends Controller
4 {
5     public function prihlaseniAction()
6     {
7        echo "<hl>Ahoj studente</hl>";
8     }
9 }
```

Zdrojový kód 2: PHP

Model

Adresář obsahuje jednotlivé modely (reprezentované třídami) pro přístup k datům za pomocí implementovaného ORM ⁵ (object-relational mapping). ORM slouží k namapování databázových entit (v našem případě tabulek) na modely (tabulky reprezentované třídami a sloupce jako jejich proměnné). Následně můžeme s daty zacházet jako s objekty. Vývojář je pak odstíněn od nutnosti psát rutinní SQL⁶ dotazy bohužel za cenu nižší flexibility a menšího snížení výkonu. V Simplexu jsou modely využívány poskrovnu, hlavně z důvodu nízké flexibility. I přesto uvedu demonstrační příklad jednoduché tabulky bez relací. Uvažujme databázi o jedné tabulce *ucitele* se sloupci *id* a *jmeno*. Náš model se pak bude jmenovat *Ucitel*. Bylo by tedy potřeba vytvořit v modelech soubor *Ucitel.php* s následujícím kódem.

```
1
   <?php
2
   use Phalcon\Mvc\Model;
3
   class Ucitel extends Model
4
5
   {
 6
      public $id;
7
      public $jmeno;
8
9
10
      public function getSource()
11
       {
          return "ucitele";
12
13
       }
   }
14
```

Zdrojový kód 3: PHP

 $^{^5 {\}rm programovací}$ technika, která zajišťuje automatickou konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem

 $^{^6 \}mathrm{jazyk}$ pro práci s daty v realačních databázích

Vytvoření nového záznamu v tabulce Ucitele by pak v aplikaci mohlo vypadat.

```
1 $ucitel = new Ucitel();
2 $ucitel->jmeno = "Tomas";
3 If($ucitel->save() === true){
4 echo "Tomas byl pridan";
5 } else {
6 $chyba = $ucitel->getMessages();
7 echo "Ups, nastala nasledujici chyba:" . $chyba;
8 }
```

Zdrojový kód 4: PHP

View

Adresář obsahuje soubory pro prezentaci dat aplikace. K efektivnějšímu generování HTML stránky implementuje Phalcon systém *Volt*. Jedná se o šablonovací systém napsaný v jazyce C, syntaxe je podobná jazyku *Jinja* (šablonovací systémem pro Python). Obecně šablonovací systémy slouží pro generování HTML a PHP kódu rychlejším a efektivnějším způsobem než za pomocí čistého PHP a HTML. Také poskytuje lepší přehlednost kódu.

4.3.1.2 Struktura Simplexu

Podobně jako v předchozí kapitole se struktura Simplexu skládá z adresářů *app*, *public* a nově *cache*. *Cache* sloužící pro ukládání již vygenerovaných *views* (pro zvýšení výkonu). Složka *app* obsahuje kromě složek: *controllers*, *view* a *models* i jiné účelové podsložky viz obrázek 16. Jejichž význam bude vysvětlen níže.



Obrázek 16: Kompletní mapa adresářů Simplexu

Config

Obsahuje konfigurační soubory aplikace a nastavení služeb (například služby provádějící překlad), které vzniknou při inicializaci celé aplikace pomocí *index.php* ve složce *public*.

- Config.php obsahuje informace nutné pro připojení do databáze či cesty k adresářům
- Loader.php je určen k registraci tříd, controllerů a modelů aplikace (soubor obsahuje také definice globálních konstant)
- Router.php obsahuje pravidla pro správné přesměrování požadavků konkrétnímu Controlleru
- Services.php je soubor s nastavením služeb aplikace (jehož kód souboru bývá zpravidla v *index.php*). Zde je například nastavení šablonovacího systému *Volt Engine*, inicializace databázového připojení a apod.
- -Translation.
php je soubor pro inicializaci služby zajištující cizojazyčný překlad dle překladů ve složce
 messages

Forms

Složka zpravidla obsahuje třídy s definicemi formulářů aplikace. Bohužel v aplikaci se tahle metoda generování formulářů nepoužívá, důvodem je slabá podpora a flexibilita ze strany frameworku.

V tomhle ohledu je vyspělejší například Framework Nette, který má dobrou podporu pro generování formulářů pomocí tříd.

Library

Obsahuje třídy pro generování některých komponent Simplexu jako například navigační panel učitelského účtu, nebo knihovnu pro export dat z testu do formátu.

Logs

Složka obsahuje chybové logy, každý název logu je tvořen datem vytvoření (dnem a měsícem, oddělené pomlčkou).

Messages

Složka obsahuje soubory s překlady Simplexu.

Plugins

Adresář obsahuje dva důležité pluginy, první je pro zpracování a přesměrování standardních chybových stavů aplikace. Druhý plugin se stará o zabezpečení přístup k aplikaci.

Vendor

Poslední složka v pořadí, skrývá definice různorodých pomocných tříd jako například třídu pro stránkování, manipulaci s účty apod. Ve složce jsou uloženy také zdrojové soubory Ratchet serveru, pro real-time komunikaci skrz webové sockety.

4.3.2 Struktura databáze

Databáze byla navržena, aby splňovala zadání aplikace pro uložení dat. Návrh databázového modelu byl vytvořen za pomocí nástroje *MySQL Workbench*. Nástroj byl vybrán díky funkci, která umožňuje provést tvorbu/aktualizaci entit databáze přímo z modelu. Integrita databáze je zajištěna cizími klíči. Po vykonání dotazu *Delete* je automaticky provedena akce pro *Cascade*, která se postará o smazání odkazujících záznamů. Na obrázku 17 je vidět úplné databázové schéma Simplexu.



Obrázek 17: Relační schéma databáze

Pro lepší pochopení databázové struktury je nutné popsat účely jednotlivých tabulek schématu.

- Teachers záznamy o učitelských účtech
- Languages tabulka s dostupnými jazykovými variacemi aplikace
- Users záznamy o ne-registrovaných studentech
- Students registrovaní studenti, vazbu mají na tabulku users pro neregistrované
- Tests_types tabulka obsahující názvy typů šablon
- Tests popisující informace o šablonách

- Generated_tests obsahuje záznamy o aktivních testech a hlasováních
- Tests_changable_options nastavení pro aktivované testy (limit, platnost apod.), obsahuje také příznak, jestli je test opravený či nikoli
- Tests_marks známky a body pro test
- Questions_types typy otázek
- Questions zadání otázek
- Answers_singlechoices zadání odpovědí typu jedna odpověď
- Answers_multiplechoices zadání odpovědí typu více odpovědí
- Users_rated_questions tabulka opravených otázek, nesoucí informaci o správnosti zodpovězené otázky
- Users_storage záznamy o připojení studentů do testu
- Users_answers_singlechoices odpovědi studentů na otázky typu *Jedna odpověd*
- Users_answers_multiplechoices odpovědi studentů na otázky typu *Více odpovědí*
- Teacher_storage záznamy o uložišti šablon, vazební tabulka mezi šablonou a učitelem
- Tags_storage štítky šablon (neobsahuje duplicitní názvy)
- Teachers_storage_has_tags_storage vazba mezi uložištěm šablony a štítkem

Databáze také obsahuje časovou rutinu (událost) - instance_inactive_by_endDate, která v pravidelných intervalem (každých 5 minut) ukončí testy, kterým skončila platnost.

4.4 Popis řešení vybraných partií Simplexu

V téhle kapitole budou blíže popsány způsoby implementace vybraných částí aplikace.

4.4.1 Ratchet server

Implementace hlasování v aplikaci probíhá v reálném časem za pomocí webových socketů na straně klienta a webového serveru Ratchet pro komunikaci s webovými sockety na straně serveru. Bez fungujícího Ratchet server nemůže být založeno hlasování. Test na druhou stranu má jisté prvky využívající webové sockety, ale bez Ratchet serveru může bez problémů fungovat.

4.4.1.1 Instalace Ratchetu

Nejjednodušší cesta jak stáhnout zdrojové soubory Ratchetu je pomocí *Composer*. Jedná se o správce balíků a závislostí pro PHP. Instalaci *Composer* a Ratchet serveru zde nebudeme popisovat, protože součástí projektu.

4.4.1.2 Struktura a spuštění serveru

Nudge server je název implementace serveru Ratchet pro Simplex, dále v textu budeme používat už pouze Nudge Server. Na obrázku 18 je vidět kompletní struktura.



Obrázek 18: Kompletní struktura Nudge serveru

Definice základních tříd a rozhraní připojení učitele/studenta obsahuje složka *Connection.* Adresář *Repository* pak obsahuje definice repositářů (tříd) pracujících s připojeními učitele a studenta. Samotný Nudge server lze spustit pomocí příkazu z konzole: **php app/vendor/NudgeServer/server.php**.

4.4.1.3 Serverová část

V souboru *server.php* je inicializace třídy *NudgeSystem* obsluhující komunikaci. Program následně běží ve smyčce a čeká na spojení od klienta/ů. V případě, že se student přihlásí k hlasování za pomocí webových socketů, je serveru zaslána zpráva, jenž první řadě zavolá metodu *onOpen* pro navázání spojení.

```
1 public function onOpen(ConnectionInterface $conn)
2 {
3 $this->pubRepository->addClient($conn);
4 echo "[New] User [" . $conn->resourceId . "] connected\n";
5 }
Zdrojový kód 5: JS
```

Klient je pak přidán do uložiště aktuálních připojení. Učitel/student se následně rozeznává dle prvotní zprávy, která je zaslána po úspěšném připojení. V tomhle stavu je spojení se serverem navázáno a může proběhnout poslání prvotní datové zprávy.

4.4.1.4 Serverová část

jQuery pluginy sloužící k připojení učitele a studenta se nachází ve složce s *public/js/connection/*. Klient se k serveru připojí za pomocí Javascriptiového webového API *WebSocket*. Prvotní připojení (například pro *locahost*) může vypadat následovně.

```
1 function getNewConnection(){
2   //Naváže spojení s localhostem na portu 8080
3   return new WebSocket("ws://localhost:8080");
4 }
```

Zdrojový kód 6: JS

Zprávy jsou ve formátu JSON, následující příklad znázorňuje obecnou definic pro zprávu od učitele.

```
1 {
2 who: 'teacher',
3 instanceId: instanceId,
4 command: command,
5 data: data
6 }
```

Zdrojový kód 7: JSON

4.4.1.5 Teacher.js

Slouží k navázání spojení učitele s Nudge serverem. Má mnoho užitečných metod, které obhospodařují dané připojení.

- $\bullet~init$ inicializuje spojení a pošle prvotní zprávu, která má za následek založení záznamu o isntanci
- *close* uzavře spojení
- *quitInstance* funkce pro ukončení instance. Bere jeden argument a to funkci, která je zavolána po odeslání zprávy serveru o ukončení instance (například vymazání záznamu o probíhající instanci)

4.4.1.6 Student.js

Slouží ke komunikaci s Nudge serverem ze stránek studenta. Metody slouží převážně k jedno-účelnému poslání zprávy (navázání spojení a následně poslání patřičné zprávy) a uzavření spojení. Oproti připojení učitele, které se uzavírá až po ukončení celé instance. Tenhle princip zaručuje snížení náporu na serveru a dle něj byl i pojmenován, protože anglické slovíčko *Nudge* se do češtiny překládá jako *šťouchnutí*. Seznam metod obsluhující instanci.

- init inicializuje test, pošle zprávu o připojení do testu příslušnému učitelskému spojení
- initVoting inicializuje hlasování, stáhne obsah daného hlasování
- close uzavře spojení
- end(form) ukončí spojení a provede odeslání formuláře, jenž je brán jako první argument metody
- *vote* provede hlasování

4.4.2 Řešení zabezpečení přístupu stránek

O zabezpečení stránek se stará Phalcon plugin, respektive třída *SecurityPlu-gin.php* (umístěn v *app/plugins/*). Ten má na starost dle typu účtu povolit/zakázat uživateli přístup k požadované stránce. Dále už pouze plugin. Plugin je navázán na metodu, která se volá před zavoláním **Dispatcher**. Ten má za úkol převést požadavek od klienta na frameworkové a extraktorvat jméno modulu, controlleru akce a poté instancovat příslušný controller a zavolat požadovanou akci. Navázání je definováno v konfiguračním souboru *services.php*.

```
1 {
2 $di->set('dispatcher', function() use ($di) {
3
4 $eventsManager = new EventsManager;
5
6 $eventsManager->attach('dispatch:beforeDispatch', new SecurityPlugin);
7 //...
Zdrojový kód 8: PHP
```

Dispatcher pak volá zděděnou metodu pluginu *beforeDispatch* a projde ACL⁷ (Access control list) a rozhodne se, jestli má typ účtu práva přístupu k požadované stránce. V případě, že mu je přístup odmítnut, je uživatel přesměrován na stránku s chybou. Samotná implementace pluginu spočívá v přidělení příslušných jmenných prostorů k typům účtů a určení veřejně jmenných prostorů aplikace. Vygenerované ACL se následně ukládá do session pro rychlejší zpracování dalších požadavků.

 $^{^7\}mathrm{druh}$ seznamu určený pro ukládání přístupových práv uživatelů

Závěr

V rámci bakalářské práce byla implementována webová aplikace za pomocí PHP frameworku Phalcon. Cílem práce bylo vytvořit aplikaci pro tvorbu online testů a hlasování. Vývoj aplikace byl pod dohledem firmy OLC, která se zabývá vývojem sofistikovaných webových systémů. Aplikace splňuje zadané požadavky a bude nadále vyvíjena. Zástupcem firmy Davidem Pokorným, byl také napsán posudek o kvalitě spolupráce během vývoje.

Spolupráci s Josefem Beťákem hodnotím kladně. Je vidět, že má v programování přehled a dokáže se rychle adaptovat na vstupní požadavky. Specifika spočívala ve využití PHP frameworku Phalcon, se kterým neměl dřívější zkušenosti. Projekt dopadl přesně tak, jak jsme chtěl a očekával. Mezi klady řadím schopnost diskuze a ne "slepé"plnění pokynů.

Tento dokument pojednává o aplikacích s podobným účelem, technických aspektech a popisuje řešení vybraných problémů při implementaci aplikace. Součástí dokumentu je také uživatelská dokumentace a příloha s instrukcemi pro instalaci a spuštění aplikace.

Conclusions

The web application is developed as a part of bachelor's thesis and powered by Phalcon PHP framework. The idea of thesis work is to create the application for making tests and voting. Development process was coordinated by OLC company that creates sophisticated information systems. Current development stage was approved by company and project is going to continue. Representative of the company David Pokorný has written the review about bachelor work and cooperation quality.

The cooperation during the development of application with Josef Beták is rated positively. It is obvious that he is perspective and can quickly adapt to input requests. The project requirements consist of knowledge in Phalcon PHP framework that he didn't have any previous experience. He is very responsible and it's very easy to communicate. Also he is professionally solving all the tasks and always sharing his point of view and ideas about the task resolutions

The thesis describes similar applications, technical aspects and selected solution of problems during the development. User documentation and instructions for installation and running the application is included to thesis as well.

A Nasazení a testování aplikace

A.1 Požadavky

Pro nasazení aplikace je nutné mít nainstalovány následující technologie.

- (a) MySQL 5.6.24 (nebo vyšší, ale netestováno)
 Zdroj: https://www.mysql.com/downloads/
- (b) Xampp 5.6.8 (nebo vyšší, ale netestováno) popřípadě Wamp s verzí PHP 5.6.8 (nebylo otestováno). Navíc je nutné stáhnout PHP framework Phalcon s verzí 3.0.3 z oficiálních stránek https://phalconphp.com/cs/ (je nutné dodržet verzi frameworku zvláště v kombinaci s PHP verzí webového serveru, jinak nebude správně fungovat)

Zdroj: https://www.apachefriends.org/index.html

A.2 Konfigurace

A.2.1 PHP frameworku Phalcon v Xampp

Po stažení na oficiálních stránkách Phalconu je nutné přesunout knihovnu (soubor s příponou DLL) do podadresáře *xamp/php/ext*. Následně je nutné knihovnu zaregistrovat v konfiguračním souboru php.ini, kde stačí přidat jeden řádek: extension=php_phalcon.dll

Zkontrolovat jestli byla knihovna korektně naimportována, je možné pomocí stránky Xampp serveru na localhostu nebo za pomocí php funkce – *phpinfo()*. Pokud knihovna není obsažena ani v jednom z výpisů, je nejčastějším problémem nekompatibilita PHP verze na serveru s verzí knihovny Phalcon.

A.2.2 MySQL

Po instalaci je nutné nahrát databázi Simplexu. Docílit toho můžeme různými způsoby. Buď pomocí příkazového řádku nebo vhodného grafického rozhraní například webového správce databáze *phpMyAdmin* (je součástí základní instalace server *Xampp*, popřípadě *Wamp*).

Poté je nutné upravit konfigurační soubor *app/config/config.php*, pro správné fungování je potřeba upravit následující parametry z větve *database*.

- host adresa serveru
- username jméno databázového uživatele
- password heslo databázového uživatele
- name název databáze

A.2.3 Simplex

Pro fungování aplikace je nutné upravit konfigurační soubor app/config/config.php a jeho parametr baseUri z větve application, který určuje cestu ke zdrojovému adresáři aplikace. Následně je pak možné povolit/zakázat používání služeb využívající emaily.

- Kontaktujte nás
- Hlídací pes (registrový student)

Posledním krokem je zapnout Nudge Server viz kapitola Ratchet server.

B Obsah přiloženého CD/DVD

bin/

Datové soubory pro nasazení aplikace a instační soubor XAMP serveru a knihovnu Phalcon

doc/

Text práce ve formátu PDF, vytvořený podle požadavků Katedry Informatiky PřF UP v Olomouci. Obsahem složky jsou také zdrojové soubory pro vygenerování textu ve formátu PDF, včetně příloh a obrázků.

src/

Kompletní zdrojové soubory Simplexu

readme.txt

Instrukce pro spuštění webové aplikace a poznámky k aplikaci

Bibliografie

- [1] PHP.NET, Official PHP documentation [online]. [cit. 2016-09-16]. Dostupné z: http://php.net/docs.php
- [2] W3SCHOOLS, JavaScript Tutorials [online]. [cit. 2016-09-16]. Dostupné z: http://www.w3schools.com/js/
- [3] RATCHET, Ratchet: WebSockets for PHP [online]. [cit. 2016-09-16]. Dostupné z: http://socketo.me/
- [4] JQUERY, jQuery: API documentation [online]. [cit. 2016-09-16]. Dostupné z: http://api.jquery.com/
- [5] AJAX, AJAX: API documentation [online]. [cit. 2016-09-16]. Dostupné z: http://api.jquery.com/jquery.ajax/
- [6] BOOTSTRAP, HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. [online]. [cit. 2016-09-16]. Dostupné z: http://getbootstrap.com/
- [7] WEBSOCKETS, Websockets API documentation [online]. [cit. 2016-09-16]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
- [8] COMPOSER, Dependency Manager for PHP [online]. [cit. 2016-09-16]. Dostupné z: https://getcomposer.org/
- [9] MYSQL, MySQL: Open Source SQL database management system [online]. [cit. 2016-09-16]. Dostupné z: https://dev.mysql.com/doc/refman/5.7/en/whatis-mysql.html
- [10] APACHEFRIENDS, XAMP: Apache distribution containing Ma-PHP. Perl [online]. [cit. 2016-09-16]. Dostupné riaDB. and z: https://www.apachefriends.org/index.html
- [11] PHALCONPHP, PHP C-extension (Phalcon) [online]. [cit. 2016-09-16]. Dostupné z: https://phalconphp.com/en/
- [12] MYSQL Products: MySQL Workbranchs [online]. [cit. 2016-09-16]. Dostupné z: https://www.mysql.com/products/workbench/
- [13] JSON, ECMA-404 The JSON Data Interchange Standard [online]. [cit. 2016-09-16]. Dostupné z: http://www.json.org/index.html
- [14] W3SCHOOLS, HTML5 API documentation [online]. [cit. 2016-09-16]. Dostupné z: https://www.w3schools.com/html/html5_intro.asp
- [15] WIKIPEDIA, The Free Encyclopedia: Model-view-controller [online]. [cit. 2016-09-16]. Dostupné z: https://en.wikipedia.org/wiki/Model-view-controller

- [16] WIKIPEDIA, The Free Encyclopedia: SQL [online]. [cit. 2016-09-16]. Dostupné z: https://en.wikipedia.org/wiki/SQL
- [17] PHPMYADMIN [online]. [cit. 2016-09-16]. Dostupné z: https://www.phpmyadmin.net/