

Portál na podporu online výuky cizích jazyků

Web portal to support online teaching foreign languages

Martin Kristofčák



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin Kristofčák, DiS.**

Osobní číslo: **A12467**

Studijní program: **N3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Forma studia: **kombinovaná**

Téma práce: **Portál na podporu online výuky cizích jazyků**

Zásady pro vypracování:

1. Analyzujte problematiku a vytvořte úvodní studii k projektu.
2. Navrhněte strukturu systému, databáze a uživatelského rozhraní.
3. Analyzujte bezpečnostní rizika aplikace a navrhněte vhodné zabezpečení.
4. Vytvořte webovou aplikaci a popište zásadní části řešení.
5. Nasaďte aplikaci a otestujte hotové řešení.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. ACHOUR Mehdi, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Georg Richter, Damien Seguy a Jakub Vrana. PHP Manual. Hypertext Preprocessor. [online]. [cit. 2014-01-19]. Dostupné z: <http://www.php.net/manual/en/>
2. MySQL Reference Manuals. Developer Zone. [online]. [cit. 2014-01-19]. Dostupné z: <http://dev.mysql.com/doc/>
3. Programmers Reference Guide of Zend Framework 2. Zend Framework. [online]. [cit. 2014-01-19]. Dostupné z: <http://framework.zend.com/manual/2.2/en/index.html>
4. jQuery API Documentation. jQuery. [online]. [cit. 2014-01-19]. Dostupné z: <http://api.jquery.com/>
5. WebRTC API. Your Web, documented. [online]. [2014] [cit. 2014-01-19]. Dostupné z: <http://docs.webplatform.org/wiki/apis/webrtc>

Vedoucí diplomové práce:

Ing. Bc. Pavel Vařacha, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce:

21. února 2014

Termín odevzdání diplomové práce:

20. května 2014

Ve Zlíně dne 21. února 2014

prof. Ing. Vladimír Vašek, CSc.
děkan



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

ABSTRAKT

Tato práce popisuje vývoj webového portálu sloužícího jako podpora online výuky cizích jazyků. V teoretické části práce jsou popsány technologie použité pro vývoj aplikace tedy jazyky HTML, PHP, databázový systém MySQL, Zend Framework 2, front-end framework Bootstrap, javascriptový framework jQuery a projekt WebRTC jenž umožňuje realizaci real-time komunikace prostřednictvím webového prohlížeče. Praktická část začíná návrhem aplikace, který je rozdělen na úvodní studii, analýzu funkčních a nefunkčních požadavků, analýzu bezpečnosti aplikace a návrhy USE CASE modelu, databáze, diagramu tříd a designu. Dále je v praktické části popsán postup vývoje aplikace, její testování a nasazení do provozu.

Klíčová slova: webový portál, online výuka cizích jazyků, video hovor, PHP, MySQL, Zend Framework 2, WebRTC

ABSTRACT

This thesis describes the development of the web portal to support online teaching foreign languages. In the theoretical part there are described technologies used in development, HTML and PHP languages, MySQL database system, Zend Framework 2, Bootstrap front-end framework, jQuery framework and project WebRTC that enables web browsers with Real-Time Communications (RTC) capabilities via simple JavaScript APIs. In the practical part there is described the initial study followed by the analysis of requirements, the analysis of application security, UML USE CASE diagram, UML class diagram, the model of database and the design proposal. Next there are described application development, testing and launching in the end of the practical part.

Keywords: web portal, online teaching foreign languages, video call, PHP, MySQL, Zend Framework 2, WebRTC

Zde bych chtěl poděkovat vedoucímu mé práce Ing. Bc. Pavlu Vařachovi, Ph.D. za ochotu, trpělivost a spolupráci při tvorbě této diplomové práce. Dále bych chtěl poděkovat své manželce za její psychickou podporu a trpělivost.

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.....	10
I TEORETICKÁ ČÁST.....	12
1 POUŽITÉ TECHNOLOGIE.....	13
1.1 JAZYK HTML	13
1.2 JAZYK PHP	14
1.3 MYSQL	14
1.4 ZEND FRAMEWORK 2	14
1.4.1 Modulární struktura aplikace	15
1.4.2 MVC.....	15
1.4.3 Překladač	15
1.4.4 Formuláře	15
1.4.5 Router	16
1.5 BOOTSTRAP FRONT-END FRAMEWORK	16
1.6 FRAMEWORK JQUERY	16
1.7 WEBRTC PROJEKT	17
II PRAKTICKÁ ČÁST	18
2 NÁVRH APLIKACE	19
2.1 ÚVODNÍ STUDIE.....	19
2.2 ANALÝZA FUNKČNÍCH POŽADAVKŮ	19
2.2.1 Registrace uživatele	20
2.2.2 Kontrola uživatelského jména a hesla	20
2.2.3 Přihlášení a odhlášení uživatele	21
2.2.4 Vyžádání přihlašovacích údajů	22
2.2.5 Změna osobních údajů	22
2.2.6 Zrušení účtu uživatele	22
2.2.7 Správa konta uživatele	23
2.2.8 Zveřejnění volného termínu	24
2.2.9 Výběr lektorů dle zadaných kritérií.....	24
2.2.10 Rezervace termínu uživatelem	26
2.2.11 Zrušení rezervace termínu	26
2.2.12 Realizace výuky	26
2.2.13 Ohodnocení lektora	27
2.2.14 Ohodnocení studenta	27
2.2.15 Změna jazyka portálu	27
2.2.16 Změna měny portálu	27
2.2.17 Sdílení stránky portálu na sociálních sítích.....	28
2.2.18 Zobrazení FAQ stránky	28
2.2.19 Zobrazení stránky O nás.....	28
2.2.20 Editace článku	28
2.2.21 Aktivace a deaktivace administrátora	28
2.2.22 Odstranění hodnocení.....	28
2.2.23 Rozesílání upozornění	28
2.2.24 Připsání prostředků na konto uživatele	29
2.2.25 Aktualizace kurzovního lístku.....	29
2.2.26 Označení požadavku na výběr prostředků jako zpracovaný	29

2.2.27	Záznam statistik návštěvnosti stránek	29
2.3	ANALÝZA NEFUNKČNÍCH POŽADAVKŮ	29
2.3.1	Zabezpečení.....	29
2.3.2	Použití real-time komunikátoru.....	30
2.3.3	Aplikace responsivní rozložení	30
2.4	BEZPEČNOST	30
2.4.1	Ochrana dat	30
2.4.2	Ochrana dat v databázi	32
2.4.3	Zabezpečení heslem	34
2.4.4	Ochrana před brute force attack	35
2.4.5	Využití protokolu HTTPS	36
2.4.6	Různé úrovně oprávnění pro přístup k datům	36
2.5	USE CASE MODEL	37
2.5.1	Aktéři systému	37
2.5.2	Případy užití	39
2.6	NÁVRH DATABÁZE	39
2.7	DIAGRAM TŘÍD	42
2.8	NÁVRH DESIGNU	46
2.8.1	Drátěné modely	47
2.8.2	Grafika stránky.....	54
2.8.3	Použité prvky	55
3	VÝVOJ APLIKACE	56
3.1	STRUKTURA APLIKACE	56
3.1.1	Knihovny aplikace	57
3.1.2	Modul Application	57
3.2	PŘÍPRAVA LAYOUTŮ.....	57
3.3	PŘÍPRAVA MODELOVÝCH TŘÍD	58
3.4	PŘÍPRAVA FORMULÁŘŮ	59
3.4.1	Filtry	59
3.4.2	Validátory.....	59
3.5	CONTROLLERY	60
3.6	POPIS STĚŽEJNÍCH FUNKCÍ PORTÁLU.....	60
3.6.1	Registrace / Přihlášení uživatele	60
3.6.2	Zpracování obrázků.....	61
3.6.3	Dobití / výběr kreditu	61
3.6.4	Aktualizace kurzovního lístku.....	63
3.6.5	Realizace video hovoru	63
3.7	OBSLUHA JAVASCRIPTEM	63
3.7.1	JQuery	64
3.7.2	RequireJS	64
3.7.3	TinyMCE.....	64
3.7.4	Lightbox	64
4	TESTOVÁNÍ APLIKACE	65
4.1	TESTOVACÍ SCÉNÁŘE.....	65
4.1.1	Registrace a editace uživatele	65

4.1.2	Správa konta.....	65
4.1.3	Výuka	65
4.1.4	Články, sociální sítě, statistiky.....	65
5	NASAZENÍ APLIKACE DO PROVOZU	66
5.1	NAHRÁNÍ SOUBORŮ APLIKACE	66
5.2	KONFIGURACE.....	66
5.3	PŘÍPRAVA DATABÁZE	66
5.4	NASTAVENÍ SERVERU	66
5.5	ZMĚNA HESLA HLAVNÍHO ADMINISTRÁTORA	66
6	DALŠÍ VÝVOJ PORTÁLU	67
	ZÁVĚR	68
	ZÁVĚR V ANGLIČTINĚ.....	69
	SEZNAM POUŽITÉ LITERATURY	70
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	72
	SEZNAM OBRÁZKŮ	73
	SEZNAM TABULEK.....	74

ÚVOD

Cílem této diplomové práce je vyvinout webový portál, který bude sloužit jako pomůcka pro komunitu lidí, kteří chtějí online vyučovat cizí jazyky na straně jedné, a kteří se chtějí online učit cizí jazyky na straně druhé. Portál bude tříditelným seznamem lektorů s různou jazykovou vybaveností, nabízejících své služby a konkrétní termíny vyhrazené pro výuku jazyka, bude umožňovat rezervaci těchto termínů, vyžadovat dobítí kreditu, kterým se bude rezervace termínu platit a po realizaci výuky se tento kredit bude přesouvat na konto lektora, který takto nabytý kredit bude moci vybrat formou bankovního převodu, či jiného platebního kanálu. Realizace výuky bude probíhat formou video konzultace s použitím vestavěného online komunikátoru nebo softwaru třetí strany.

Na začátku byla myšlenka snadného, flexibilního a efektivního studia cizího jazyka. Cizí jazyk se dnes lidé běžně učí od prvního stupně základní školy, často i dříve, avšak domluvit se cizí řečí je pro mnohé z nich oříšek. Ve škole se většinou naučí všelikou gramatiku, ale chybí dostatek praxe. Pokud někdo po ukončení školní docházky přestane komunikovat cizí řečí úplně, začne se jeho praktická schopnost použití cizího jazyka strmě zhoršovat. Pro oživení komunikačních dovedností pak často lidé navštěvují různé výukové kurzy. Čas strávený docházením či dojížděním na kurz, rozvrh hodin takového kurzu či absence dostatečně kvalifikovaného personálu mohou činit takovéto kurzy časově náročnými, neflexibilními a neefektivními.

Jako zajímavá alternativa, jak pilovat komunikační dovednosti, by se mohla jevit komunikace s lektorem přes nějaký real-time komunikátor, např. Skype. Velkou výhodou této metody výuky je nepochybně efektivita, výuku si lze naplánovat do časového okna mezi prací, je možné studovat či vyučovat večer po práci, na pracovní cestě, z dovolené, ráno, v poledne, večer i v noci. Netrávíte čas dojížděním na kurzy, cestu na kurz vám nezhatí dopravní zácpa, porucha auta, nepotřebujete učebnu, stačí laptop, případně smartphone.

Pro někoho může být problém absence osobního kontaktu, která se použitím video hovoru téměř maže. Hlavním problémem potenciálního studenta je zajištění svého osobního lektora. Pokud nemá někoho vhodného ve svém okolí, musí se poohlédnout jinde. Může to zkusit v inzerci ať už tištěné, nebo na Internetu, nebo třeba získat kontakt od někoho známého. Pokud takového lektora sežene, musí si s ním sjednat lekce. Může si buď naplánovat pravidelné lekce, což ovšem není příliš flexibilní, nebo si může lekce domlouvat průběžně, ale tento způsob může být vcelku náročný. Proto by bylo vhodné, aby lektor měl někde

veřejně dostupný rozvrh termínů, které jsou volné, a na které je možné se nahlásit. K tomu může dobře posloužit webová prezentace. Aby taková prezentace byla v internetu dobře dostupná, musí se umisťovat na tematicky zaměřená klíčová slova v popředí výsledků internetových vyhledávačů nebo musí využít dostatečné marketingové podpory. To může být ve velké konkurenci náročné nebo příliš drahé na realizaci. Pokud je tento problém překonán, je potřeba zvolit některý z dostupných real-time komunikátorů, za všechny uvedme Skype, Hangouts či Whatsapp. Komplikace může nastat ve chvíli, kdy student a lektor nevyužívají stejný komunikátor a jedna či obě z komunikujících stran jsou méně technicky zdatné. Pak i zdánlivě malý problém může výuku zcela zhatit.

Mezi další nástrahy lze zařadit způsob doručení odměny od studenta lektorovi. Mohou se domluvit na bankovním převodu, převodu peněz pomocí PayPal nebo jiném způsobu úhrady služeb. Bankovní převod je ovšem neflexibilní, neboť jeho realizace trvá zpravidla několik dní. Pokud se navzájem neznají, může to být poměrně riziková transakce, zvláště, kdyby chtěl lektor zaplatit předem.

Hlavní myšlenkou, kterou je inspirována tato práce, je tedy vytvořit webový portál, který bude umožňovat lektorům vyučujícím cizí jazyky, případně překladatelům, nabízet své služby, nabízet svoji volnou kapacitu formou plánovacího kalendáře, dále pak umožňovat studentům rezervovat si výuku a tu umožnit realizovat pomocí vestavěného real-time komunikátoru přímo prostřednictvím tohoto portálu. Dále poskytnout možnost přesouvat finanční prostředky mezi studentem a lektorem prostřednictvím tohoto portálu, provádět vklad prostředků na konto vedené na portálu, stejně tak výběr prostředků z konta portálu na vlastní bankovní konto. Pro lepší orientaci mezi lektory a studenty je nutné umožnit hodnocení výuky a spolehlivosti lektora i studenta, tak aby se další uživatelé mohli snadno mezi ostatními uživateli portálu orientovat.

I. TEORETICKÁ ČÁST

1 POUŽITÉ TECHNOLOGIE

Volba technologií a vývojových nástrojů je jednou z klíčových fází vývoje software. Výběr musí být podřízen různým požadavkům, např. požadavkům na prostředí, ve kterém bude výsledný produkt provozován, bezpečnostním kritériím, rozpočtu projektu, znalostem vývojového týmu, zkušenostem a dalším kritériím.

V případě tohoto projektu je kladen důraz na využití volně dostupných technologií, které se stále vyvíjí a používá je dostatečná komunita vývojářů. Struktura stránek bude definována pomocí značkovacího jazyka HTML, styl stránek bude popsán jazykem CSS, generování obsahu stránek bude obstarávat skriptovací jazyk PHP, data portálu bude ukládat a spravovat SŘBD MySQL.

Projekt nebude stavěn na „zelené louce“, pro jeho vývoj bude použit Zend Framework 2, potřebná javascriptová podpora bude realizována pomocí frameworku jQuery. K dosažení dnes potřebné responzibility rozložení stránek bude využit Bootstrap Front-End Framework. Pro realizaci real-time video hovorů bude použit javascriptový projekt WebRTC.

1.1 Jazyk HTML

Jedná se o hypertextový značkovací jazyk, jehož hlavní využití spočívá ve vytváření dokumentů publikovaných na Internetu. Vývoj tohoto jazyka jde ruku v ruce s vývojem webových prohlížečů, jejichž vývoj je tímto jazykem ovlivněn, stejně tak, jako je zpětně ovlivněn vývoj tohoto jazyka právě prohlížeči.

Historie jazyka HTML sahá do roku 1990, přes verze 1, 2 a klíčové verze 3 a 4 se tento jazyk dočkal až aktuální verze 5, která je zatím pouze ve stavu „Candidate Recommendation“ a svého schválení by se měla dočkat koncem roku 2014.

Pro sestavení HTML kódu portálu bude použito verze 5 tohoto jazyka, což by mělo být základem jeho dlouhodobější snadné aktualizace. Velkou výhodou bude využívání sémantických značek, které umožňují dobrou orientaci v kódu nejen vývojáři, ale především botům internetových vyhledávačů. Dále mj. bude možné s úspěchem využít mnoho nových hodnot dostupných pro atribut type formulářového pole input, popřípadě bude možné vkládat do stránek různá multimédia, vše ovšem bude záviset na rychlosti rozšíření podpory klíčových vymožeností jazyka HTML verze 5 jednotlivými internetovými prohlížeči. Více se o HTML 5 lze dočíst z [1].

1.2 Jazyk PHP

Jedná se o velmi rozšířený Open Source skriptovací jazyk, jež je vhodný pro vývoj webových stránek, především pro dynamické generování obsahu stránek, zpracování formulářů atp.

Historie jazyka PHP sahá do roku 1995, postupným vývojem dospěl až k verzi 5, která je zatím poslední v řadě a na rozdíl od předchozí verze nabízí především vylepšenou a plnohodnotnou podporu objektově orientovaného programování, popřípadě PDO rozšíření, které definuje rozhraní pro připojení k databázím. Aktuálně je jazyk k dispozici ve verzi 5.4, jež bude použita i při vývoji tohoto projektu. Více informací o jazyce PHP je možné získat v literatuře [2-4].

1.3 MySQL

Tento celosvětově velmi rozšířený databázový systém je dostupný jak pod bezplatnou licenci, tak pod placenou (komerční) licenci. Pro účely provozování portálu lze používat MySQL bezplatně, čímž je opět splněna podmínka volně šiřitelných technologií.

MySQL podporuje několik datových úložišť, např. ARCHIVE, CSV, MEMORY, MyISAM, InnoDB a další. Z důvodu potřeby použití relací mezi tabulkami databáze a transakcí bude použito úložiště InnoDB, které jejich podporu má.

Historie MySQL se začíná psát v roce 1995, původně systém fungující pouze na OS unix a linux se v roce 1998 dočkal verze pro MS Windows. Postupem času se vyvinul přes verze 3.23, 4.0, 4.1 a 5.0 až po aktuální verzi 5.1, která nabízí oproti první verzi subqueries, kurzory, uložené procedury, trigger, pohledy a další užitečné nástroje. Další informace o MySQL lze čerpat z publikací [4-6].

1.4 Zend Framework 2

Zend Framework 2 je open source framework vhodný pro vývoj webových aplikací běžících na PHP 5.3 a novějším. Je na 100% psán objektově, užívá modulární architektury, umožňuje vývojáři používat pouze potřebné komponenty, je ale potřeba dávat pozor na často se vyskytující závislosti mezi komponentami. Nabízí robustní a výkonnou MVC implementaci, rozhraní pro přístup k databázi, formulářovou komponentu pracující s HTML 5, validátory, filtry, cache, autorizační komponentu a mnoho dalšího.

Vývoj Zend Frameworku (verze 1) začal v roce 2005, získal si poměrně hodně velkou popularitu v podobě několika milionů stažení, a tak logicky došlo s postupem času ke kompletnímu přepsání frameworku a v roce 2012 spatřil světlo světa Zend Framework 2, který je nyní k dispozici ve verzi 2.25, která bude použita pro vývoj tohoto projektu. Další informace o tomto frameworku je možné nalézt v [7-8].

1.4.1 Modulární struktura aplikace

Aplikace postavená na tomto frameworku se skládá z modulů. Moduly jsou části aplikace, které dokáží pracovat nezávisle na ostatních modulech aplikace. Každá aplikace vytvořená v Zend Framework musí obsahovat alespoň jeden modul.

1.4.2 MVC

Moduly aplikace vytvořené v Zend Frameworku využívají jím poskytované prostředky architektury MVC. Každý modul odděluje datový model, uživatelské rozhraní a řídicí logiku. Tato architektura je dnes velmi populární a zároveň výhodná, neboť jednotlivé tři části jsou na sobě téměř nezávislé, a je tedy možná náhrada pouze jedné z nich novější verzí bez nutnosti zásahu do kterékoliv další. Řídicí logika se nazývá controller, uživatelské rozhraní tvoří tzv. view skripty a model je tvořen třídami zpřístupňující data v databázi.

1.4.3 Překladač

Překladač je jedním z modulů tohoto frameworku. Umožňuje snadnou správu pevného obsahu webu, jako jsou různé popisky, nadpisy atp. Texty jsou udržovány v souborech s příponami po a mo a pro jejich editaci a generování se využívá specializované aplikace, např. Poedit. Postup je takový, že se nejprve použije tohoto modulu všude tam v kódu, kde se vkládá nějaký text podléhající překladu. Následně se pomocí specializované aplikace z kódu vyextrahují všechny takto vložené texty do souboru s příponou po, přeloží se do příslušných jazyků a z těch se pak vygenerují soubory s příponou mo. Tyto soubory je pak možné v aplikaci napojit na modul překladače a ten bude na určená místa dodávat překlady ze zvoleného souboru.

1.4.4 Formuláře

Formulář je základem každého dynamického webu. Při práci s formuláři musí vývojář řešit neustále stejné úlohy např. vykreslení formulářových elementů, předvyplnění dat elementů, kontrola vstupů atp. Tomu se však vyhne při použití formulářových tříd poskytovaných Zend

Frameworkem. Jsou zde obsaženy třídy automatizující generování formulářů a formulářových prvků, zastřešující kontrolu vstupů formuláře, vyhodnocení formuláře včetně výpisu upozornění na chybně vložené vstupy.

1.4.5 Router

Tento klíčový plugin frameworku umožňuje nakonfigurovat strukturu validních požadavků na adresu a pak napříč portálem snadno generovat potřebné odkazy a také automatizovaně vyhodnocovat validitu požadované adresy v rámci webu.

1.5 Bootstrap Front-End Framework

Tento volně dostupný a jeden z nejpopulárnějších front-end frameworků nabízí předdefinované styly a javascriptové knihovny pro snadnou tvorbu responzivních webových stránek optimalizovaných pro zařízení s různým rozlišením zobrazovacích zařízení. Obsahuje předdefinované HTML šablony, formulářové prvky, navigační prvky a další komponenty.

Jedná se o poměrně mladý projekt, který svým vznikem v roce 2012 reagoval na masové rozšíření mobilních zařízení, především tabletů a smartphonů, které nedisponují tak vysokým rozlišením jako běžné osobní počítače, a proto potřebují speciální pozornost vývojáře, aby udělal web snadno čitelný a obsluhovatelný i z těchto zařízení. Aktuální verze z února 2014 nese označení 3.1.1 a bude použita při tvorbě tohoto portálu. Vše potřebné o projektu Bootstrap včetně zdrojových kódů a uživatelské příručky lze dohledat v [9].

1.6 Framework jQuery

jQuery je malá, rychlá, volně dostupná javascriptová knihovna, která díky svému jednoduchému API umožňuje snadnou manipulaci s elementy dokumentu a CSS vlastnostmi, obsluhu událostí, tvorbu efektů a animací, použití AJAXu a mnoho dalšího napříč nejpoužívanějšími webovými prohlížeči.

Framework vznikl v roce 2006, dnes je k dispozici souběžně ve verzích 1.11.0 a 2.1.0. Druhou zmíněnou verzi však nepodporuje MS Internet Explorer až do verze 8, proto pro tento portál bude prozatím využito verze 1.11.0. Více informací o JavaScript, jQuery a AJAX je k dispozici v [10-15].

1.7 WebRTC projekt

WebRTC je volně dostupný projekt, který umožňuje uskutečnit real-time komunikaci prostřednictvím jednoduchého javascriptového API. Tento projekt je zatím podporován prohlížeči Chrome, Firefox a Opera.

Historie projektu se píše od roku 2011, souběžně s projektem se vyvíjí konkurenční projekt od Microsoftu CU-RTC-WEB, což komplikuje situaci s jednotnou podporou projektu WebRTC napříč prohlížeči, neboť právě Explorer od Microsoftu

jeho podporu dosud nezavedl. Aktuální informace o projektu WebRTC je možné čerpat z [16].

II. PRAKTICKÁ ČÁST

2 NÁVRH APLIKACE

Návrh aplikace je stěžejní částí vývoje spolehlivého software. Před samotným programováním aplikace je vhodné seznámit se s požadavky na její funkcionalitu, stanovit okruh budoucích uživatelů, analyzovat rizika a v dostatečné míře všechny důležité části vývoje popsat. Tato fáze bude rozdělena do těchto částí: úvodní studie, analýza požadavků, bezpečnost, USE CASE model, návrh databáze, diagram tříd a návrh designu.

2.1 Úvodní studie

Cílem projektu je vytvořit portál, sloužící komunitě lidí, kteří chtějí online vyučovat cizí jazyky formou konverzace na straně jedné, a kteří se chtějí online učit cizí jazyky na straně druhé.

Projekt je reakcí na absenci podobného produktu mezi do češtiny lokalizovanými webovými stránkami, tuto mezeru na trhu se pokusí zacelit a zároveň konkurovat zahraničním projektům.

Portál bude plnit funkci tříditelného seznamu lektorů, kteří budou formou plánovacího kalendáře nabízet svoje vyučovací hodiny návštěvníkům portálu. Návštěvníci si mohou konkrétní hodinu prostřednictvím portálu rezervovat. Výuku bude možné realizovat prostřednictvím vestavěného nástroje pro online video konverzaci. Návštěvníci budou mít možnost dobýt si kredit prostřednictvím PayPal, tento kredit bude čerpán v rámci výukových lekcí, lektoři si takto nabytý kredit budou moci vybrat formou bankovního převodu. Portál bude automatizovaně zasílat potvrzovací a notifikační emaily upozorňující na důležité akce.

Skupinu uživatelů přistupujících k portálu vedle studentů a lektorů doplní ještě nepřihlášení uživatelé a administrátoři. Možnosti jednotlivých typů uživatelů budou definovat uživatelská oprávnění.

K vytvoření aplikace bude použito výhradně volně šiřitelných vývojových nástrojů a technologií.

2.2 Analýza funkčních požadavků

Tato fáze návrhu aplikace se zaměřuje na analýzu a popis požadavků na vyvíjený produkt. Požadavky na produkt lze rozdělit do dvou základních skupin a to na funkční a nefunkční. Tato kapitola je zaměřena na funkční požadavky, tedy funkce, které musí portál uživatelům

poskytovat. Jednotlivé požadavky budou podrobně popsány a bude se z nich vycházet v dalších krocích návrhu a realizace aplikace.

2.2.1 Registrace uživatele

Nepřihlášený uživatel portálu musí mít k dispozici registrační formulář, prostřednictvím kterého se bude moci zařadit mezi registrované uživatele portálu. Registrační formulář bude požadovat zadání emailové adresy uživatele, přístupového hesla a typu uživatele.

Při zadávání emailové adresy se bude kontrolovat validita struktury emailové adresy.

Heslo bude muset uživatel zadat dvakrát z důvodu ochrany před překlepy. Při zadávání bude probíhat kontrola shody těchto dvou řetězců a zároveň bude kontrolováno, zda heslo plní bezpečnostní požadavky.

Pokud zadané údaje budou vyhovovat všem kontrolám, uživateli bude umožněno odeslání registračního formuláře. Po odeslání se provede znova kontrola na straně serveru. V případě selhání se znova zobrazí registrační formulář s příslušnou chybovou hláškou, v opačném případě se provede registrace uživatele a odeslání potvrzovacího emailu na zadanou emailovou adresu. Emailová zpráva bude obsahovat odkaz pro potvrzení vlastnictví emailové adresy. Uživateli se zobrazí potvrzovací hláška o úspěšné registraci a o nutnosti kliku na odkaz v potvrzující emailové zprávě.

Další službou, kterou bude registrační formulář poskytovat, bude registrace administrátora. Přihlášený administrátor bude mít možnost využít registračního formuláře pro registraci nového administrátora. Pro tento účel uvidí kromě základních typů uživatele také typ Administrátor, který bude zatržený a bude signalizovat, že se jedná o registraci nového administrátora. Proces registrace se nijak neliší od výše uvedeného, jen se uživatel označí jako administrátor.

2.2.2 Kontrola uživatelského jména a hesla

Při registraci uživatele a při změně emailové adresy či nastavení nového přístupového hesla bude docházet ke kontrole správnosti zadaných údajů. Tato kontrola bude prováděna jak na straně klienta, tak na straně serveru.

Po vyplnění uživatelského jména bude provedena kontrola, zda tato položka splňuje následující kritéria:

1. splňuje formát emailové adresy,

2. neexistuje v databázi shodná emailová adresa.

Jestliže položka nebude ve formátu emailové adresy, zobrazí se uživateli chybová hláška, informující o nutnosti vložení emailové adresy ve správném formátu. Jestliže bude zjištěno, že daná emailová adresa se již nachází v databázi portálu, bude o tomto uživatel informován informativní hláškou a bude směřován na formulář sloužící k přihlášení.

Heslo bude muset uživatel při registraci zadat dvakrát z důvodu ochrany před překlepem. Heslo bude muset splňovat následující kritéria:

1. obsah obou vstupních polí pro zadání hesla se musí shodovat,
2. heslo musí splňovat požadavky definované v kapitole Bezpečnost.

Jestliže se obě zadaná hesla neshodují nebo heslo nesplňuje stanovené požadavky, bude o této skutečnosti uživatel informován chybovou hláškou.

Jestliže uživatel odešle registrační formulář ke zpracování na server, provede se znovu kontrola správnosti zadaných údajů na straně serveru. V případě existence emailové adresy v databázi bude uživatel přesměrován na stránku s přihlašovacím formulářem, kde se mu předvyplní uživatelské jméno. Ve všech ostatních případech neshody bude uživatel přesměrován na stránku s registračním formulářem, který se předvyplní zadanými údaji a zobrazí se mu příslušné chybové hlášky.

2.2.3 Přihlášení a odhlášení uživatele

Kdekoliv napříč portálem, musí mít nepřihlášený uživatel k dispozici odkaz pro přihlášení se ke svému účtu. Po kliku na odkaz bude uživatel přesměrován na stránku s přihlašovacím formulářem.

Přihlašovací formulář bude vyžadovat zadání uživatelského jména a hesla. Pokud uživatel nic nevyplní a pokusí se odeslat formulář, bude o tom informován chybovou hláškou.

Po odeslání formuláře bude na straně serveru provedena kontrola vyplněnosti údajů a kontrola existence uživatele se shodnými údaji. Pokud některé z polí formuláře nebylo vyplněno, nebo neexistuje v databázi uživatel se shodnými přihlašovacími údaji, bude uživatel přesměrován na přihlašovací formulář a na chyby bude upozorněn chybovou hláškou. V případě, že bude nalezen uživatel se shodnými přihlašovacími údaji, portál uživatele přihlásí a přesměruje jej na stránku jeho profilu, kde si může editovat údaje o sobě, případně na jinou stránku, ze které byl na přihlašovací formulář přesměrován.

2.2.4 Vyžádání přihlašovacích údajů

Přihlašovací formulář bude obsahovat odkaz vedoucí na formulář umožňující zaslání nového hesla. Formulář pro zaslání nového přístupového hesla bude vyžadovat pouze zadání uživatelského jména. Po odeslání formuláře dojde ke kontrole existence takového uživatelského jména v databázi. Pokud takové uživatelské jméno v databázi neexistuje, bude na tuto skutečnost uživatel upozorněn chybovou hláškou. V opačném případě se uživateli vygeneruje a nastaví nové náhodné heslo, které splňuje podmínky definované v kapitole Bezpečnost a toto se společně s uživatelským jménem zašle na zadanou emailovou adresu. O odeslání emailu bude uživatel informován potvrzující hláškou.

2.2.5 Změna osobních údajů

Formulář umožňující změnu osobních údajů přihlášeného uživatele bude dostupný z menu stránky. Na tento formulář bude uživatel zároveň automaticky přesměrován ihned po jeho přihlášení.

Formulář pro změnu osobních údajů bude umožňovat editaci jména, země, časového pásma, emailové adresy, hesla, pohlaví, rodného jazyka, ovládaných jazyků, profilového obrázku, motto, podrobného popisu profilu a certifikací. Obsah pole emailová adresa musí splňovat formát emailové adresy, heslo musí splňovat požadavky definované v kapitole Bezpečnost, motto slouží k vystižení uživatele jednou větou, podrobný popis slouží pro detailnější charakteristiku uživatele, jeho životopis atp.

Povinné položky jsou emailová adresa, pohlaví a rodný jazyk. Kontrola správnosti vyplnění formuláře bude opět probíhat jak na straně klienta, tak na straně serveru a o problémech při kontrole bude uživatel informován chybovou hláškou. O správném vyplnění údajů a uložení do databáze bude uživatel informován potvrzující hláškou.

2.2.6 Zrušení účtu uživatele

Každý uživatel bude mít možnost zrušení svého konta. Tato možnost bude k dispozici ve formuláři sloužícímu k editaci osobních údajů. Po zvolení této možnosti, bude uživatel vyžádán k zadání svého přístupového hesla. Po zadání hesla a odeslání požadavku na server, dojde na serveru k ověření hesla, pokud se neshoduje, zobrazí se uživateli chybová hláška, v opačném případě je uživatel odhlášen, účet deaktivován a uživateli se zobrazí potvrzovací hláška. O zrušení konta bude uživatel informován prostřednictvím emailové zprávy.

2.2.7 Správa konta uživatele

Formulář umožňující editaci konta uživatele bude dostupný z menu portálu každému přihlášenému uživateli. Formulář bude umožňovat editaci výchozí měny uživatele, dále pak vklad a výběr prostředků. Na této stránce bude rovněž k vidění hodnota zůstatku konta uživatele a seznam transakcí, které od registrace do portálu na uživatelském kontě proběhly. O transakci bude potřeba evidovat datum a čas uskutečnění transakce, popis transakce, hodnotu transakce, stav konta po transakci a měnu, ve které byla transakce zaznamenána.

Vklad prostředků bude v první verzi portálu umožněn pouze prostřednictvím PayPal, po nasazení portálu do provozu budou podle potřeby doimplementovány další platební kanály.

Po zadání požadavku na vklad prostředků dojde ke kontrole částky zadané do pole určené k tomuto účelu. Pokud toto pole neobsahuje celé číslo v povoleném rozsahu, zobrazí se uživateli chybová hláška, v opačném případě bude uživatel přesměrován na stránky PayPal. Po přijetí potvrzení platby ze systému PayPal dojde k připsání prostředků na konto uživatele a odeslání potvrzovací emailové zprávy na emailovou adresu uživatele.

Výběr prostředků z konta uživatele bude umožněn formou bankovního převodu. Požadavek k převodu prostředků bude mít možnost přihlášený uživatel zadat také na stránce konta uživatele. Uživatel bude muset uvést celočíselnou částku určenou k převodu, číslo účtu, na který chce prostředky převést a heslo pro přístup ke svému kontu.

Kontrola správnosti vyplněných údajů bude probíhat jak na straně klienta, tak na straně serveru. Na straně klienta se zkontroluje vyplnění všech údajů, dále pak se provede kontrola částky, jestli je její hodnota celočíselná a v povoleném rozsahu a kontrola formátu čísla bankovního konta, které musí být zadáno v mezinárodním formátu. V případě neúspěšné kontroly bude uživateli zobrazena chybová hláška, v opačném případě bude formulář odeslán ke zpracování na server.

Na serveru se opět zkontroluje hodnota částky a formát čísla účtu, navíc se zkontroluje správnost zadaného hesla. V případě nějaké neshody se uživateli zobrazí chybová hláška a předvyplní se mu zadané údaje do formuláře pro požadavek na výběr prostředků. V opačném případě se zaznamená požadavek do databáze a prostředky se uživateli odečtou z jeho konta.

Fyzické zadávání příkazů k úhradě se bude prozatím dělat ručně, v budoucnu bude tato akce částečně automatizovaná generováním dat pro hromadné zadávání příkazů k úhradě nebo plně automatizovaná propojením portálu s bankovním účtem přes API bankovního ústavu.

O uskutečněném požadavku na výběr prostředků bude uživatel informován prostřednictvím emailové zprávy.

2.2.8 Zveřejnění volného termínu

Lektor bude mít k dispozici plánovací kalendář, ve kterém bude mít možnost založit jednorázové i opakované termíny výuky. Přihlášený lektor bude mít v menu k dispozici odkaz na editaci volných termínů, který povede na stránku, kde bude k dispozici jednak kalendář, ve kterém bude přehled vytvořených termínů lektora, které může lektor editovat, dále pak odkaz na založení nového termínu. Formulář pro založení a editaci termínu výuky bude vypadat stejně.

Formulář pro založení/editaci termínu výuky bude obsahovat následující pole:

1. datum od kdy je termín platný,
2. datum do kdy je termín platný,
3. čas od kdy je lektor k dispozici,
4. čas do kdy je lektor k dispozici,
5. poznámka k termínu,
6. čas pozdní rezervace (kolik hodin před začátkem lekce si může student rezervovat termín),
7. čas pozdního zrušení rezervace (kolik hodin před začátkem lekce může student či lektor rezervaci zrušit),
8. dny v týdnu, pro které je termín platný.

Pokud některou z položek, kromě poznámky, uživatel nevyplní, bude na to upozorněn chybovou hláškou. Kontrola bude probíhat standardně jak na straně klienta, tak na straně serveru. Pokud je formulář vyplněn správně, termín se založí a uživateli se zobrazí potvrzovací hláška.

2.2.9 Výběr lektorů dle zadaných kritérií

Na úvodní stránce portálu bude seznam registrovaných lektorů. Výchozí seznam bude obsahovat maximálně 20 lektorů, případní další lektoři se budou načítat Ajaxem v případě kliknutí na tlačítko na konci seznamu. Nefiltrovaný seznam bude řazen podle následujících kritérií a v tomto pořadí:

1. Existence rezervace termínu u lektora (v současnosti i v minulosti),
2. Lektor hovořící v jazyce dané jazykové verze portálu, případně v jazyce přihlášeného uživatele,
3. Průměrné hodnocení lektora (vyšší = lepší),
4. Počet uskutečněných lekcí celkem,
5. Datum registrace na portálu (dříve = lépe).

Uživatel bude mít k dispozici filtr, kterým může zúžit výběr lektorů. Filtrování bude umožněno podle následujících faktorů:

1. Jazyk, který musí lektor ovládat,
2. Země, ze které lektor pochází,
3. Hodinová sazba lektora,
4. Počet odučených lekcí.

Řazení vybraných lektorů bude možné podle následujících kritérií:

1. Výchozí,
2. Podle hodnocení,
3. Cena lekce,
4. Počet odučených lekcí.

Po potvrzení filtru tlačítkem „Filtrovat“ dojde k načtení seznamu lektorů splňujících zvolená kritéria. Každý lektor v seznamu bude zobrazen formou karty, na které budou následující informace:

1. Jméno a fotografie lektora jako aktivní odkaz na profilovou stránku lektora,
2. Průměrné hodnocení lektora, pokud nemá žádné, tak zašedlé s textem „dosud nehodnocen“,
3. Hodinová sazba lektora v měně jazykové verze portálu, případně měně přihlášeného uživatele,
4. Počet odučených lekcí,
5. Počet hodnocení s odkazem na seznam hodnocení na profilové stránce lektora,

6. Odkaz na vytvoření rezervace termínu, vedoucí na kalendář na profilové stránce lektora,
7. Odkaz s textem „zobrazit profil“ vedoucí na profilovou stránku doktora.

2.2.10 Rezervace termínu uživatelem

Na profilu lektora bude zobrazen kalendář, ve kterém budou zveřejněny volné i obsazené termíny výuky daného lektora. Přihlášený uživatel bude mít možnost si volný termín rezervovat. Klikem na vybraný termín výuky se uživateli zobrazí rezervační formulář termínu výuky. Pokud student rezervaci potvrdí, dojde ke kontrole zůstatku na účtu studenta. Jestliže pokryje náklady na výuku vypočtené jako hodinová sazba lektora krát počet rezervovaných hodin, dojde k zablokování prostředků ve prospěch této výuky, studentovi se zobrazí potvrzovací hláška a lektorovi se odešle emailová zpráva s informací o nové rezervaci termínu výuky. V případě nedostatku prostředků se studentovi zobrazí chybová hláška.

2.2.11 Zrušení rezervace termínu

Student i lektor budou mít možnost zrušit rezervaci termínu výuky, pokud tak učiní nejpozději tolik hodin před začátkem výuky, kolik je nastaveno v detailu termínu výuky. Na seznam rezervovaných termínů se uživatel dostane z menu. Každá rezervace, kterou je možné zrušit, bude mít v seznamu rezervací tlačítko pro zrušení rezervace. Po kliku na toto tlačítko se provede zrušení rezervace termínu a odešle se zúčastněným uživatelům emailová zpráva potvrzující zrušení rezervace výuky. Zároveň se provede odblokování prostředků na účtu studenta, které se zablokovaly právě pro účely výuky v tomto termínu. O správnosti provedení této akce bude uživatel informován potvrzující hláškou.

2.2.12 Realizace výuky

Výuka bude uskutečňována prostřednictvím real-time komunikátoru integrovaného do portálu. Přihlášený uživatel uvidí v záhlaví stránky odkaz na realizaci nejbližšího termínu výuky spolu s časem, za který bude tato výuka započata. Po kliku na určený odkaz se uživateli zobrazí stránka s detailem lekce a dvěma rámečky, jeden sloužící pro výstup z vlastní videokamery a druhý sloužící pro video protistrany. Ve chvíli, kdy se připojí protistrana, dojde ke spojení hovoru. Pro případ, že nebude možné uskutečnit hovor pomocí vestavěného komunikátoru, bude zde informace o možnosti použití nějakého softwaru třetí strany.

2.2.13 Ohodnocení lektora

Přihlášený uživatel bude mít možnost ohodnotit lektora, se kterým uskutečnil v minulosti prostřednictvím portálu výukovou lekci. Odkaz na hodnotící formulář uživatel uvidí v seznamu uskutečněných lekcí na řádku vedle každé lekce, která již proběhla, ale nebyla dosud uživatelem ohodnocena. Tento seznam bude dostupný z hlavního menu.

Hodnotící formulář bude vyžadovat zadání slovního hodnocení a následně ohodnocení body 1-5 v přednastavených hodnotících kritériích. Při pokusu o odeslání se nejprve provede kontrola úplnosti vyplněných údajů javascriptem na straně klienta, pokud některá část hodnocení chybí, bude na toto uživatel upozorněn chybovou hláškou. V opačném případě dojde k odeslání formuláře. Kontrola úplnosti vyplnění jednotlivých položek formuláře proběhne ještě jednou na straně serveru. Pokud některé pole formuláře nebylo správně vyplněno, uživatel bude přesměrován na předvyplněný hodnotící formulář a zobrazí se mu chybová hláška. V opačném případě bude uživatel přesměrován zpět na seznam lekcí a zobrazí se mu potvrzovací hláška. Ve stejné chvíli bude hodnocenému lektorovi odeslána emailová zpráva s upozorněním na nové hodnocení od studenta. Email bude odeslán ve skryté kopii na k tomuto účelu zřízenou emailovou adresu z důvodu monitoringu nevhodných hodnocení.

2.2.14 Ohodnocení studenta

Lektor bude mít možnost v návaznosti na uskutečněnou výuku ohodnotit studenta. Odkaz na seznam dosud neohodnocených studentů bude mít lektor k dispozici v menu. Samotné hodnocení bude probíhat analogicky jako hodnocení lektora.

2.2.15 Změna jazyka portálu

V záhlaví stránky bude k dispozici volba jazyka portálu. Pokud uživatel zvolí jazyk, bude přesměrován na takto lokalizovanou verzi portálu.

2.2.16 Změna měny portálu

Částky se budou přihlášenému uživateli zobrazovat v jeho měně, nepřihlášenému uživateli se budou zobrazovat v měně určené jazykovou verzí portálu. Měnu si může uživatel změnit v nastavení svého konta, jenž bude dostupné z menu portálu.

2.2.17 Sdílení stránky portálu na sociálních sítích

Uživatel bude mít možnost sdílet kteroukoliv veřejně přístupnou stránku portálu na sociálních sítích Google+, Twitter a Facebook.

2.2.18 Zobrazení FAQ stránky

Uživatel si bude moci zobrazit stránku s často kladenými dotazy a odpověďmi na ně.

2.2.19 Zobrazení stránky O nás

Uživatel bude mít možnost zobrazení stránky O nás.

2.2.20 Editace článku

Přijde-li na stránku nějakého článku přihlášený administrátor, bude mít možnost editovat nadpis a text článku.

2.2.21 Aktivace a deaktivace administrátora

Přihlášený uživatel typu Administrátor bude mít přístup na stránku Administrace portálu, kde bude k dispozici seznam všech administrátorů. Každá položka v seznamu, kromě administrátora s příznakem ROOT, bude mít vedle sebe tlačítko pro aktivaci či deaktivaci administrátora, v závislosti na stavu účtu daného administrátora. V případě potvrzení se provede aktivace či deaktivace administrátora.

2.2.22 Odstranění hodnocení

Přihlášený uživatel typu Administrátor, bude mít možnost smazat hodnocení lektora. K této akci bude sloužit tlačítko, které bude zobrazeno u jednotlivých hodnocení na prezentaci lektora. Po stisku tohoto tlačítka si stránka vyžádá vyplnění důvodu odstranění a potvrzení akce. Po potvrzení se provede odstranění hodnocení a odeslání emailové zprávy informující o odstranění a to jak na emailovou adresu lektora, tak na emailovou adresu studenta.

2.2.23 Rozesílání upozornění

System kromě již uvedených upozornění na různé akce bude uživateli zasílat v různých intervalech formou emailových zpráv další upozornění a informace. Bude se jednat o následující:

1. Upozornění na nepotvrzenou registraci - Každé pondělí bude všem uživatelům, kteří nepotvrdili svoji registraci na portále, odcházet emailová zpráva upozorňující na tuto

skutečnost. Tato zpráva kromě informace bude obsahovat odkaz sloužící k potvrzení své registrace.

2. Upozornění na blížící se termín výuky - Každý den se bude všem uživatelům, kteří mají na následující den plánovanou nějakou výuku zasílat emailová zpráva upozorňující na blížící se termín výuky.

2.2.24 Připsání prostředků na konto uživatele

Systém bude kontrolovat zprávy od platebního portálu PayPal a na jejich základě bude připisovat prostředky na konto uživatele.

2.2.25 Aktualizace kurzovního lístku

Systém bude uchovávat aktualizovaný kurzovní lístek ve své databázi. Denně bude provádět jeho aktualizaci z dat ČNB.

2.2.26 Označení požadavku na výběr prostředků jako zpracovaný

Přihlášený uživatel typu Administrátor bude mít přístup na stránku Administrace portálu, kde bude k dispozici seznam všech nezpracovaných požadavků na výběr prostředků. Každá položka v seznamu bude mít vedle sebe tlačítko pro označení požadavku za zpracovaný. Po kliku na tlačítko se provede označení požadavku jako zpracovaný a odešle se potvrzení na email uživatele, který o převod požádal.

2.2.27 Záznam statistik návštěvnosti stránek

Návštěvy jednotlivých stránek bude potřeba monitorovat, bude potřeba mít přehled o četnosti návštěv podle data, podle typu prohlížeče, zařízení a lokality.

2.3 Analýza nefunkčních požadavků

Kromě funkčních požadavků na portál je potřeba vydefinovat v této části návrhu aplikace nefunkční požadavky na aplikaci.

2.3.1 Zabezpečení

Každá aplikace a zvláště aplikace přístupná v síti, musí být kvalitně zabezpečena. Pokud tato aplikace navíc pracuje s finančními prostředky, je nutnost kvalitního zabezpečení ještě znásobena. Způsob zabezpečení aplikace bude podrobně vyspecifikována v kapitole Bezpečnost.

2.3.2 Použití real-time komunikátoru

K uskutečnění online komunikace bude použit projekt WebRTC resp. jeho implementace knihovnou SimpleWebRTC. Bližší informace o tomto produktu jsou k dispozici v kapitole Použité technologie.

2.3.3 Aplikace responsivní rozložení

Rozložení stránek portálu se musí měnit podle rozlišení zařízení, na kterém je stránka zobrazena. Responsivní rozložení bude řešit framework Bootstrap. Bližší informace o tomto produktu jsou k dispozici v kapitole Použité technologie.

2.4 Bezpečnost

Klíčovou úlohou této kapitoly je analyzovat rizika a navrhnout řešení jak omezit dopad těchto rizik.

2.4.1 Ochrana dat

Data portálu lze obecně rozdělit na dvě skupiny, na data uložená v databázi a zdrojové kódy a soubory tvořící samotnou aplikaci. Základem bezpečného webu je dobře navržená adresářová struktura a citlivé oddělení veřejné a neveřejné části adresářové struktury.

Adresářová struktura bude vypadat takto:

```
config
├─ autoload
data
├─ contentimgs
module
public
├─ index.php
├─ css
├─ img
├─ js
scripts
vendor
```

Adresář `config` obsahuje soubory s konfigurací webu, např. přístupové údaje k databázi, ftp serveru a další více či méně citlivá data, adresář `data` obsahuje různé datové soubory související s portálem, např. obrázky uživatelů, soubory sdílené uživateli atp., adresář `module` obsahuje jednotlivé moduly, které tvoří jednotlivé sekce aplikace, adresář `vendor`

obsahuje php knihovny a to jak vlastní tak i knihovny třetích stran a konečně adresář `public` obsahuje soubory veřejné části webu.

Aby nemohla být zneužita některá citlivá data, případně aby nedošlo k odcizení zdrojových kódů, je nezbytné vymežit přístup k jednotlivým adresářům a jejich částem. Způsob tohoto nastavení se liší podle použitého webového serveru. V tomto případě se jedná o server Apache, konkrétně verzi 2.4.

Nastavení webového serveru Apache se provádí v souboru `httpd.conf`, případně jeho rozšířeních. V první řadě je dobré nastavit direktivou `DocumentRoot` cestu do adresáře webu, v našem případě `public`, tedy nějak takto:

```
DocumentRoot path/public
```

Tím je zajištěno, že se nikdo nedostane přes adresu webového portálu např. do adresáře `config` s citlivými údaji a také do adresáře `modules`, který obsahuje know-how v podobě vyvinutých modulů.

Pak je tu ovšem adresář `data`, který takto není přístupný z vnějšku, ale některé jeho části zpřístupnit bude potřeba. Například adresář `/data/contentimgs` bude sloužit pro nahrávání profilových obrázků uživatelů. Proto je potřeba k tomuto adresáři explicitně povolit přístup návštěvníkům portálu a dále k němu nastavit nějaký `alias`, pod kterým bude jeho obsah dostupný.

```
Alias /contentimgs/ "path/data/contentimgs "  
<Directory "path/data/contentimgs ">  
Options MultiViews  
AllowOverride None  
Order allow,deny  
Allow from all  
</Directory>
```

Dle zvyklostí se v aplikaci vytvořené na základě Zend Framework 2 data závislá na prostředí, kde je aplikace provozována, mezi něž patří i konfigurace připojení k databázi, ukládají do adresáře `/config/autoload`, konkrétně do souboru `local.php`. Výhodou toho řešení je jednak fakt, že tento soubor není z vnějšku přístupný, data v něm jsou tedy uložena na bezpečném místě, dále pak v případě použití verzovacího systému, např. SVN nebo GIT, tento soubor lze označit jako ignorovaný verzovacím systémem a tudíž se nebude při aktualizaci zdrojových souborů přepisovat, čímž nám na serveru zůstane serverová verze souboru a na lokálním stroji vývojová verze souboru.

Kromě uložení přístupových údajů k databázi i zde platí nutnost volby bezpečného hesla, jelikož se jedná o databázi, která obsahuje citlivá a důležitá data, tak spíše nutnost volby precizního hesla.

2.4.2 Ochrana dat v databázi

Databáze portálu bude spravována systémem řízení báze dat MySQL. O bezpečnosti právě MySQL serveru by se daly vést dlouhé debaty, ale to není úkolem této práce. Tato kapitola se zaměřuje na bezpečnost dat z hlediska vývojáře portálu.

Kromě toho, že je potřeba chránit data zabezpečením přístupu k databázi, je důležité také chránit data správnou konstrukcí databáze, vhodnou volbou datových typů polí jednotlivých tabulek, správnou volbou klíčů a správným nastavením relací mezi tabulkami.

Je vhodné vytvářet databáze pokud možno ve třetí normální formě. Relační tabulky splňují třetí normální formu (3. NF), jestliže splňují 2. NF a žádný atribut, který není primárním klíčem, není tranzitivně závislý na žádném klíči.

Definice 2. NF:

Tabulka splňuje 2. NF, právě když splňuje 1. NF a navíc každý atribut, který není primárním klíčem je na primárním klíči úplně závislý.

Definice 1. NF:

1. NF říká, že všechny atributy jsou atomické, tj. dále již nedělitelné (jinými slovy, hodnotou nesmí být relace).

Každá tabulka by měla mít svůj primární klíč, který slouží k jednoznačné identifikaci každého řádku tabulky, potažmo k vytváření relací mezi tabulkami. Existují následující relace mezi tabulkami:

1. 1:1 se používá, pokud záznamu odpovídá právě jeden záznam v jiné databázové tabulce a naopak. Takovýto vztah je používán pouze ojediněle, protože většinou není pádny důvod, proč takovéto záznamy neumístit do jedné databázové tabulky. Jedno z mála využití je zprehlednění rozsáhlých tabulek. Jako ilustraci je možné použít vztah řidič - automobil. V jednu chvíli (diskrétní časový okamžik) řídí jedno auto právě jeden řidič a zároveň jedno auto je řízeno právě jedním řidičem.
2. 1:N přiřazuje jednomu záznamu více záznamů z jiné tabulky. Jedná se o nejpoužívanější typ relace, jelikož odpovídá mnoha situacím v reálném životě.

Jako reálný příklad může posloužit vztah autobus - cestující. V jednu chvíli cestující jede právě jedním autobusem a v jednom autobuse může zároveň cestovat více cestujících.

3. M:N je méně častým. Umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé. V databázové praxi bývá tento vztah z praktických důvodů nejčastěji realizován kombinací dvou vztahů 1:N a 1:M, které ukazují do pomocné tabulky složené z kombinace obou použitých klíčů (třetí resp. tzv. vazební tabulka). Příkladem z reálného života by mohl být vztah výrobek - vlastnost. Výrobek může mít více vlastností a jednu vlastnost může mít více výrobků. V reálném životě nicméně existuje velké množství vztahů M:N, mimo jiné také proto, že často existuje praktická potřeba zachovávat i údaje o historii těchto vztahů z časového hlediska (jeden řidič v delším časovém období řídí více rozličných aut a jedno auto v delším časovém období může mít více různých řidičů).

Dále účast ve vztahu může být povinná, nebo nepovinná. Pokud je účast ve vztahu povinná, záznam nemůže bez nastaveného cizího klíče existovat. V obou případech je potřeba stanovit, jak nakládat se záznamem tabulky v případě pokusu o úpravu či smazání cizího klíče. K tomu slouží tzv. integritní omezení a referenční integrita. Definují, jak se záznamem tabulky naložit v případě, že se pokusíme integritu narušit. Definujeme jak se závislým záznamem naložit v případě úpravy a v případě smazání cizího klíče. Máme možnost na akci nereagovat (IGNORE), dále můžeme akci nechat probublat, tzn. úpravu promítnout do závislé tabulky (CASCADE), nebo můžeme akci nad cizím klíčem zamítnout (RESTRICT).

Pokud by relace mezi tabulkami byly nastaveny nevhodně, mohou v databázi zůstat nadbytečná data, která již nejsou k ničemu potřebná, nebo může dojít k nežádoucímu smazání či modifikaci dat. Proto je potřeba řešení relací v databázi věnovat zvýšenou pozornost.

Pokud je databáze vhodně navržena a přístup k ní ochráněn, je potřeba ještě správně sestavit samotnou aplikaci pracující nad touto databází tak, aby tato aplikace nevhodnou manipulací data nepoškodila.

Dále je nutné zabezpečit aplikaci proti útokům zvenčí, které by mohly vést k poškození dat. Známým druhem útoku je tzv. SQL injection technika, jíž se útočník snaží poškodit, či získat data z SQL databáze skrz neošetřený formulář, upravením adresy requestu, modifikací cookie, případně dalšími technikami. Úkolem je odeslat na server ke zpracování kousek SQL

kódu, který se vlivem neošetření aplikace odstane až do SQL dotazu a pokud splňuje syntaxi i sémantiku v rámci dané databáze, tak se také vykoná, čímž může buď nenávratně modifikovat či smazat data, nebo útočníkovi naopak zobrazit data, ke kterým nemá mít vůbec přístup.

Řešením je ošetření uživatelských dat před jejich použitím v SQL dotazu. Možností je několik, buď je možné explicitně escapovat nebezpečné znaky z dat, případně lze použít předpřipravených metod a funkcí jazyka PHP, které se o vše potřebné postarají.

2.4.3 Zabezpečení heslem

V řešené aplikaci bude existovat několik akcí vyžadujících zadání hesla. Heslo je vhodné vyžadovat nejen k autentizaci uživatele při přihlášení se ke svému účtu, ale také při zásadních akcích, vyžadujících vyšší stupeň bezpečnosti.

Může se stát, že se uživatel přihlásí ke svému účtu, ale zachová se nevhodně a od svého počítače odejde. V té chvíli si může nenechavec pod jeho identitou dělat na portále, co chce. Je to následek nezodpovědného chování majitele konta, přesto je vhodné některé akce zabezpečit opětovným vyžádáním hesla, aby se v takovýchto případech zabránilo větším škodám.

Následující akce budou vyžadovat zadání hesla:

1. Registrace uživatele,
2. Přihlášení uživatele k účtu,
3. Požadavek na výběr kreditu,
4. Požadavek na změnu hesla,
5. Požadavek na zrušení účtu uživatele.

Z hlediska návrhu aplikace je potřeba stanovit, jaké nároky na heslo budou kladeny. Z pohledu zapamatovatelnosti hesla je potřeba klást na heslo co nejnižší nároky, z hlediska bezpečnosti naopak nároky co nejvyšší. Vždy je potřeba s ohledem na citlivost dat spravovaných aplikací stanovit relevantní nároky na heslo.

Jelikož aplikace bude mj. pracovat s finančními prostředky v podobě kreditu, je vhodné na heslo klást větší požadavky, proto bude aplikace vyžadovat minimální délku hesla 8 znaků do budoucna je vhodná zvážit nutnost využít velká i malá písmena v kombinaci s čísly.

Hesla uživatelů se budou ukládat do databáze. Heslo se do databáze nesmí ukládat v nešifrované podobě, hrozilo by, že jej zneužije někdo z vývojářů případně dalších lidí,

kteří mají přístup k databázi, případně by nějakou bezpečnostní dírou aplikace mohl hacker získat seznam všech uživatelů včetně jejich hesel, což je černý scénář, ale i nad touto eventualitou je dobré se zamyslet.

Řetězec představující heslo uživatele se před uložením do databáze přetvoří metodou tzv. jednosměrného šifrování. Jednosměrným šifrováním řetězce dochází k jeho přetvoření na shluk znaků na základě hashovacího algoritmu. Výhodou tohoto principu je neexistence dešifrovacího algoritmu, který by šlo reálně použít k dešifrování výsledného řetězce a také fakt, že k jednomu šifrovanému řetězci by měl existovat pouze jeden šifrovaný hash. Takto zašifrované heslo je již možné bezpečně uložit do databáze.

Dříve se pro kryptování hesla používaly algoritmy MD5 nebo SHA-1. Jejich hashovací funkce vytvářejí otisk o délce 128 bitů resp. 160 bitů. Jejich prolomení usnadňují tzv. slovníky obsahující seznamy řetězců a jejich hashů a také fakt, že vygenerování hashe je velmi rychlé a efektivní, proto lze k prolomení s úspěchem použít metodu hrubé síly. Pro snížení rizika prolomení takto kryptovaného hesla slovníkovými útoky nebo formou brute force attacks je využíván tzv. SALT, neboli sůl, která je vkládána do řetězce s heslem a teprve takto doplněný řetězec se kryptuje zvoleným hashovacím algoritmem. Tato metoda ovšem nemění fakt, že vygenerování hashe, tedy kandidáta na prolomení šifry, je velmi rychlé. Tento nedostatek řeší algoritmus BCrypt, jež jednak používá SALT proměnné délky, navíc provádí kryptování v několika iteracích, čímž se operace vygenerování šifry podstatně zpomaluje a tím se zajišťuje vyšší bezpečnost šifry.

2.4.4 Ochrana před brute force attack

Brute force attackem může útočník na portále tohoto typu například generovat obrovské množství nových registrací, pokoušet se o prolomení hesla uživatele, obecně opakovaně odesílat různé formuláře, případně vyvíjet jinou nežádoucí činnost.

Ochrana před útoky tohoto typu se provádí jak na nižších síťových vrstvách, tak i na vyšších, např. aplikační vrstvě. Zde je vhodné se zaměřit především na řešení ochrany formulářů.

Tato aplikace bude rozlišovat mezi klientem a botnetem několika způsoby:

1. Použitím tzv. honney pots,
2. Kontrolou timeoutu, dobou mezi requestem pro zobrazení formuláře a requestem odesílajícím data formuláře.

Každý formulář bude obsahovat honney pots, tedy formulářová pole, která běžný uživatel neuvidí, jednoduchý botnet se je však před odesláním formuláře pokusí naplnit daty, čímž se prozradí a request s takto vyplněným formulářem se může zahodit jako SPAM.

Podobně se bude nakládat s requesty, které budou následovat v příliš krátkém intervalu po vygenerování formuláře. Jednoduchý botnet se totiž snaží generovat co nejvíce requestů za jednotku času, zatímco běžnému uživateli vyplnění formuláře chvíli trvá. Každý formulář tedy bude mít pole nesoucí informaci o čase jeho sestavení a tento čas se bude porovnávat s časem odeslání formuláře. Pokud tento čas nebude v definovaném intervalu, bude request zahozen jako SPAM.

2.4.5 Využití protokolu HTTPS

HTTPS (Hypertext Transfer Protocol Secure) je nadstavba síťového protokolu HTTP, která má za úkol zajistit zabezpečení spojení mezi webovým prohlížečem uživatele a webovým serverem a tím znemožnit odposlouchávání komunikace, potvrzení dat a umožňuje ověřit identitu protistrany.

Portál bude servírovat stránky prostřednictvím standardního protokolu HTTP, ovšem do budoucna je vhodné zvážit použití protokolu HTTPS využívající šifrování SSL pro stránky, které vyžadují zadání hesla a stránky, na kterých dochází k modifikaci údajů uživatelů. Před použitím tohoto způsobu šifrování komunikace je potřeba zakoupit webový certifikát potvrzující vlastnictví domény, na které portál poběží z důvodu věrohodnosti této domény.

2.4.6 Různé úrovně oprávnění pro přístup k datům

Portál bude rozlišovat 4 typy uživatelů:

1. Administrátor,
2. Lektor,
3. Student,
4. Nepřihlášený uživatel.

Tyto čtyři typy uživatelů budou mít v konfiguračním souboru aplikace definovány úrovně přístupu k jednotlivým sekcím, stránkám či akcím na stránkách. Při generování stránky vždy dojde ke kontrole, jestli má daná stránka definována omezení přístupu, pokud ano, tak zkontroluje oprávnění aktuálního uživatele k dané akci a vyvození následků v různých podobách:

1. Access denied stránky,

2. Nezobrazení určité části stránky,
3. Omezení operací na dané stránce.

2.5 USE CASE model

Jedná se o model vytvářený za použití grafického jazyka UML, který je používán v softwarovém inženýrství jako nástroj pro návrh a dokumentaci vyvíjeného software. USE CASE model má za úkol zachytit pohled na systém zvenčí, jednak definuje uživatele systému, dále pak jejich vztahy se službami, které systém poskytuje. Takto definovaný model by měl být čitelný jak pro vývojáře, tak pro uživatele systému. Více o UML a USE CASE je možné se dočíst v [17].

2.5.1 Aktéři systému

Při tvorbě use case modelu je v první řadě potřeba vydefinovat aktéry systému, tedy uživatele, kteří budou nějakým způsobem ovlivňovat či používat systém.

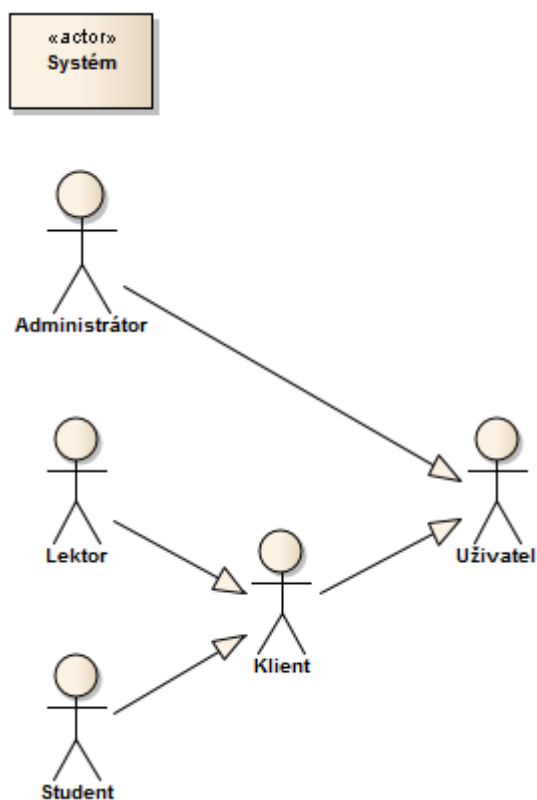
Tento portál bude rozlišovat pět aktérů a to uživatele, administrátora, lektora, studenta a systém. Navíc zde vystupuje abstraktní uživatel klient, který spojuje společné vlastnosti lektorů a studentů.

Uživatel je obecný návštěvník portálu. Má možnost listovat seznamem lektorů a filtrovat je dle různých kritérií, může si prohlížet veřejné profily lektorů, smí se přihlásit či zaregistrovat. Po přihlášení smí spravovat svůj profil a své konto, dále může rezervovat lekce, rušit rezervace lekcí a účastnit se výuky.

Klient je pouze abstraktní aktér, rozšiřuje vlastnosti aktéra Uživatel a definuje společné vlastnosti aktérů Lektor a Student.

Lektor rozšiřuje vlastnosti aktéra Klient, navíc má možnost zveřejňovat a rušit termíny výuky a hodnotit studenty.

Student rozšiřuje vlastnosti aktéra Klient, navíc může provádět rezervace termínu výuky, rušit rezervace termínu výuky a hodnotit lektory.



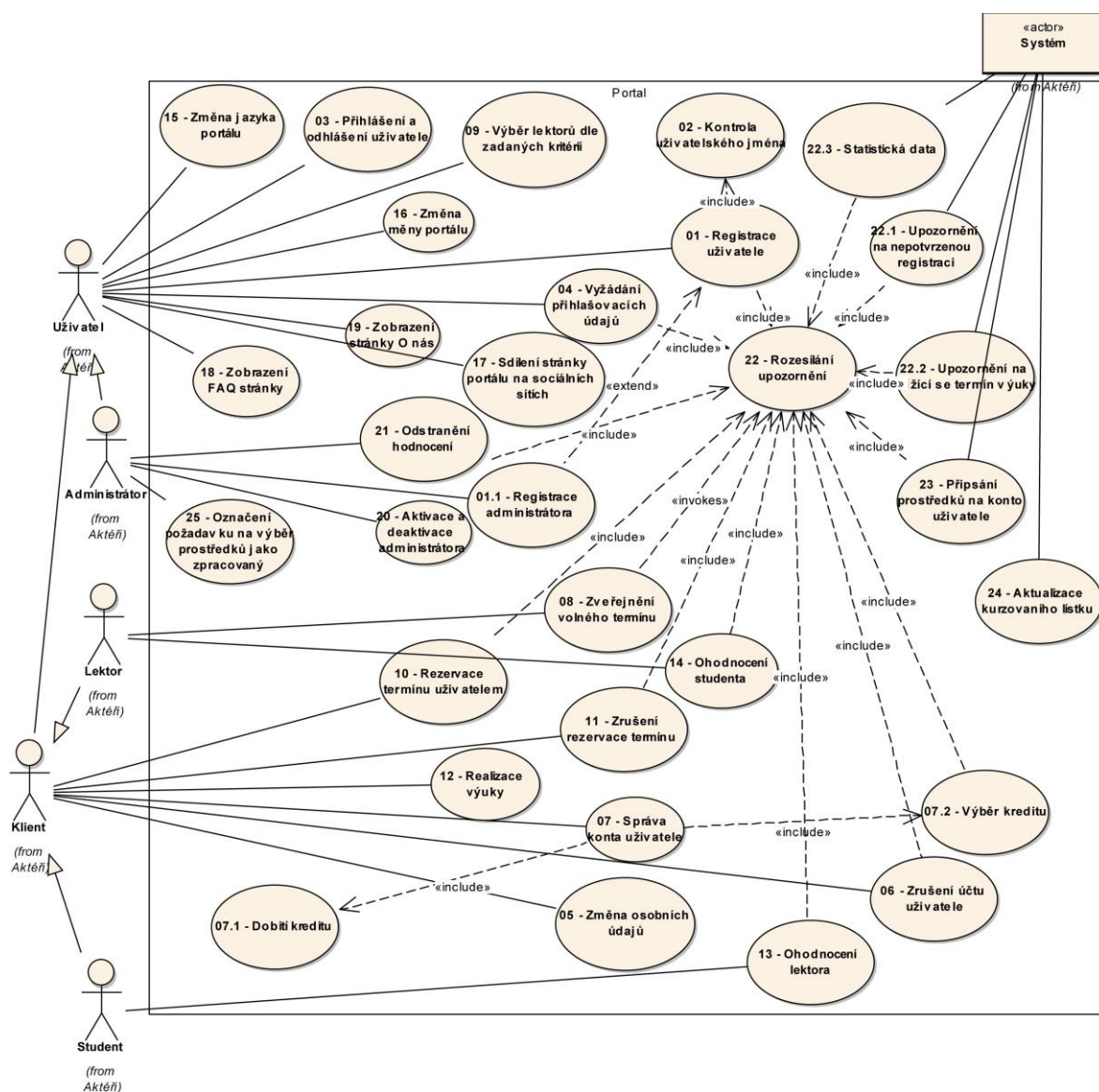
Obr. 1 - Aktéři systému

Administrátor rozšiřuje vlastnosti aktéra Uživatel o možnost vytváření a rušení dalších administrátorů, dále pak může mazat hodnocení lektorů a editovat články.

Aktér Systém spouští automatizované rozesílání emailových zpráv, jako jsou upozornění na výuku, statistiky atp., dále pak spouští proces aktualizace kurzovního lístku a připisuje vložené prostředky na konto uživatele.

2.5.2 Případy užití

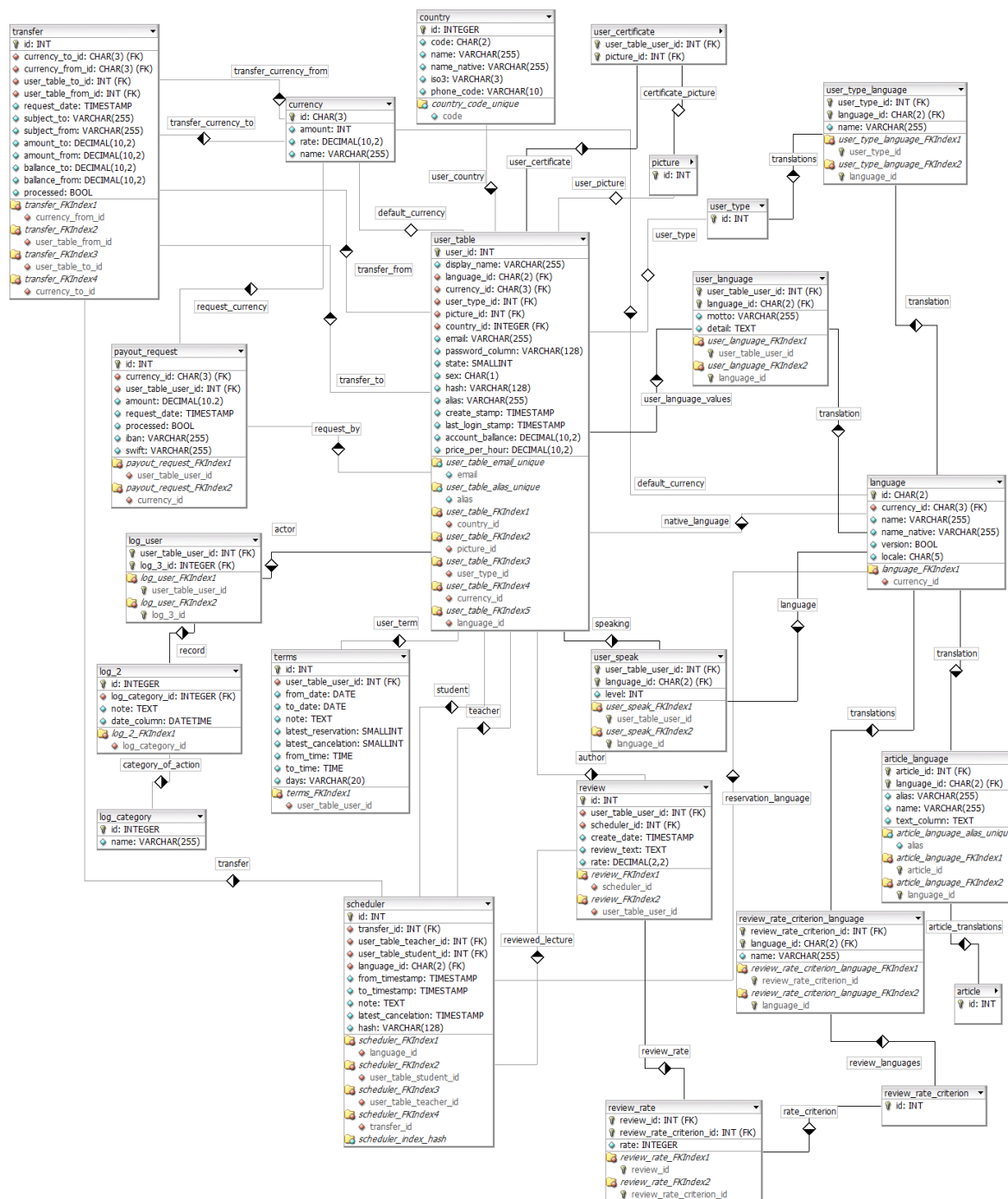
Use case model neboli model případů užití vychází z funkčních požadavků, které jsou definovány v kapitole Analýza funkčních požadavků. Názorně graficky znázorňuje vztah aktérů a služeb poskytovaných systémem. Popis těchto služeb je k dispozici ve výše zmíněné kapitole.



Obr. 2 – USE CASE diagram

2.6 Návrh databáze

Návrh databáze spočívá v definování entit, které bude potřeba rozlišovat a evidovat v databázi. Tvorbu databázového modelu je nutné podřídit zvoleným normám, kterým bude databáze podléhat, dále je potřeba zohlednit druh aplikace a upravit databázový model tak, aby danému využití co nejvíce vyhovoval.



Obr. 3 – Datový model

Páteří databázového modelu této aplikace se stanou následující tabulky:

user_table – eviduje záznamy o jednotlivých uživateli, např. identifikátor (**user_id**), nativní jazyk (**language_id**), výchozí měnu (**currency_id**), typ uživatele (**user_type_id**), uživatelské jméno uživatele/email (**email**), heslo (**password_column**), stav uživatele (**state**) a další.

language – eviduje záznamy o jednotlivých jazycích. Mezi důležité sloupce tabulky patří id, version, currency_id a locale. Id uchovává dvoupísmenný kód jazyka dle normy ISO 639-1. Sloupec version typu BOOL určuje, zda tento jazyk bude mít svoji jazykovou verzi portálu, pokud bude mít hodnotu true, bude existovat verze portálu v tomto jazyce. Sloupec currency_id má význam pouze u záznamů s version nastaveným na true. Bude určovat výchozí měnu pro danou jazykovou verzi. Podobně jako sloupec currency_id, tak i sloupec locale má význam pouze u záznamů s version nastaveným na true a označuje dialekt jazyka, který se v dané verzi bude používat.

Databáze dále bude obsahovat několik dvojic tabulek, které budou sloužit pro ukládání dat o objektech, které budou mít své překlady. Takovouto párovou tabulkou k tabulce user_table bude tabulka user_language, která bude sloužit k ukládání překladů motta a detailních informací uživatele. Dále je zde možné nalézt tabulky article a article_language, které budou ukládat texty článků, kterých příliš v plánu zatím není, ale v budoucnu se mohou začít využívat častěji. Dalším párem bude review_rate_criterion a review_rate_criterion_language pro evidenci hodnotících kritérií, které půjde zohlednit při hodnocení lektorů a studentů.

Hodnocení uživatelů se bude ukládat do skupiny tabulek review, review_rate a již zmíněné review_rate_criterion a review_rate_criterion_language. Samotné slovní hodnocení, vazba na konkrétní lekci a autora hodnocení bude v tabulce review. Kromě slovního hodnocení bude možné ohodnotit uživatele podle různých hodnotících kritérií. Hodnocení jednotlivých kritérií bude uloženo v tabulce review_rate.

Aplikace bude pracovat se dvěma typy obrázků a to profilovými obrázky uživatelů a obrázky certifikací uživatelů. Identifikátory obrázků budou uloženy v tabulce picture, certifikáty uživatele budou pak v tabulce user_certificate.

Uživatelé portálu si budou mít možnost definovat výchozí měnu a zemi původu. Dostupné měny a země budou uloženy v tabulkách currency a country. Tabulka pro evidenci měn bude kromě sloupce pro třípísmenný identifikátor obsahovat sloupce pro množství a kurz měny a název měny. Tabulka currency bude evidovat číselný, dvou i třípísmenný kód státu, název státu a telefonní předvolbu státu.

Převody peněz bude evidovat tabulka transfer. U každého převodu se budou evidovat informace jak z pohledu odesílatele, tak pohledu příjemce. Bude se tedy evidovat dvakrát měna, částka, stav konta pro transakci, text požadavku, pak samozřejmě i identifikátory obou

zúčastněných. Kromě uvedených informací bude sloupec processed evidovat boolovskou hodnotu o tom, jestli byl požadavek již proveden.

Požadavky na výběr prostředků se budou ukládat v tabulce payout_request.

Některé akce prováděné napříč portálem bude možné logovat do databáze. K tomuto účelu budou sloužit tabulky log_2, log_category a log_user.

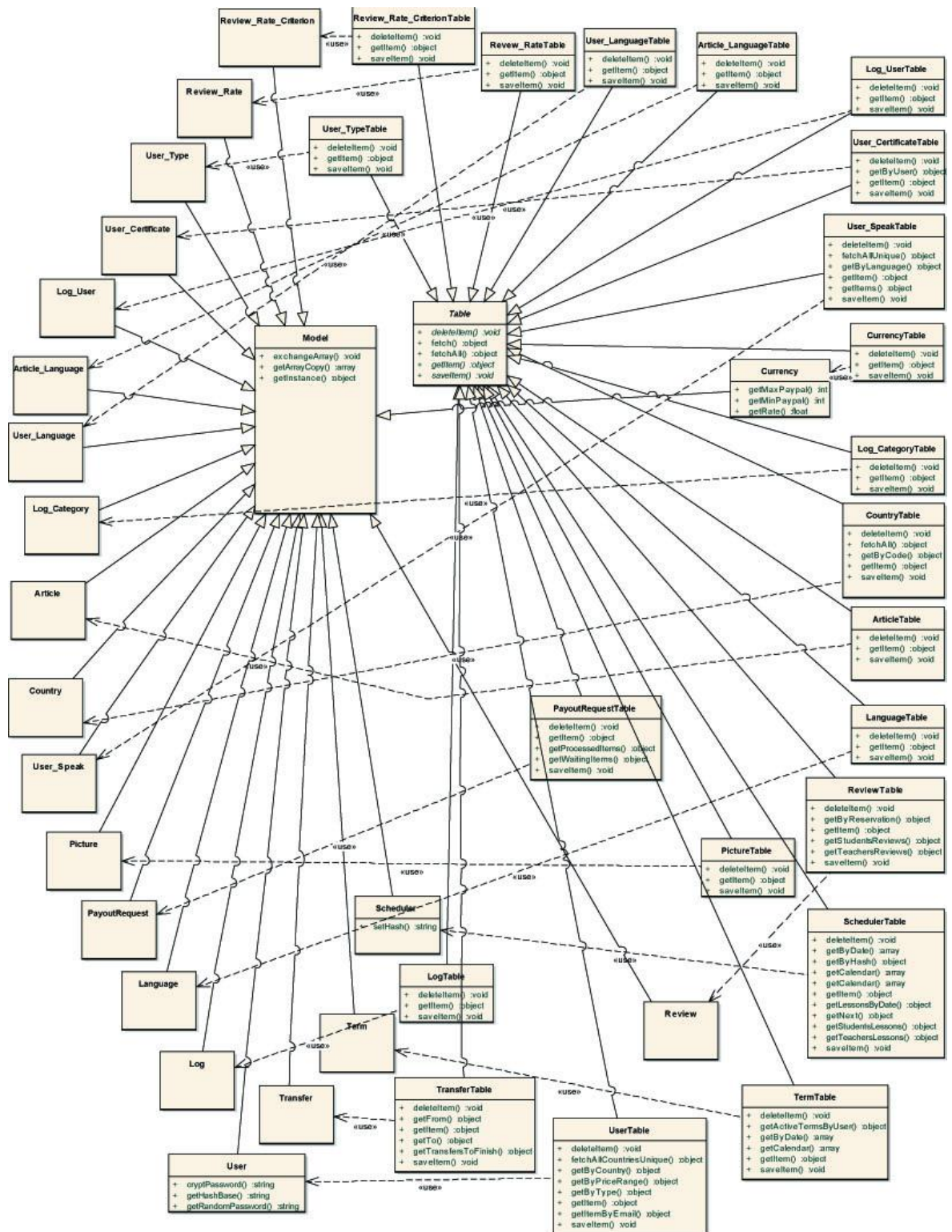
Mezi velmi důležitou část datového modelu budou patřit tabulky terms a scheduler. První uvedená tabulka bude sloužit pro evidenci termínů vypsanych lektory, druhá bude ukládat rezervace termínů studenty.

Tabulky jsou provázány pomocí cizích klíčů a mají definována integritní omezení, aby byla zachována konzistence dat. Detailně je databáze popsána modelem na předchozím obrázku.

2.7 Diagram tříd

Jelikož už je znám datový mode a soubor funkčních požadavků a tyto požadavky jsou detailně popsány, je možné se pustit do konstrukce modelu tříd.

Srdcem modelu jsou dvě třídy a to Model a Table. Třída Model bude předkem všech tříd, které budou vytvářet základní objekty kopírující databázové tabulky. Dá se říct, že každá tabulka bude mít svoji modelovou třídu, která bude potomkem abstraktní třídy Model. Tyto modelové třídy nebudou mít přímý přístup k databázi, komunikaci s databází bude zprostředkovávat další sada tříd, které budou dědit abstraktní třídu Table a budou pro své operace využívat svých modelových protějšků.



Obr. 4 – Diagram tříd

Základními metodami, které budou implementovat všechny třídy dědící třídu Table, budou:

- getItem – vrátí objekt modelové třídy naplněný daty z databáze,
- saveItem – uloží hodnoty objektu modelové třídy do databáze,

- deleteItem – smaže záznam z databáze.

Dále budou jednotlivé třídy dědící z třídy Table obsahovat podle potřeby specifické metody:

- CountryTable:
 - getByCode – vrátí záznam podle dvoupísmenného kódu,
 - fetchAll – vrátí všechny záznamy seřazený podle abecedy.
- PayoutRequestTable:
 - getWaitingItems – vrátí seznam dosud nepotvrzených požadavků na výběr prostředků,
 - getProcessedItems – vrátí seznam potvrzených požadavků.
- SchedulerTable:
 - getByDate – vrátí seznam hodin, které má v dané datum obsazené daný uživatel,
 - getByHash – vrátí záznam na základě vloženého hash kódu,
 - getCalendar – vrátí seznam rezervací pro týden dle zadaného data a identifikátoru uživatele,
 - getLessonsByDate – vrátí seznam rezervací ve zvolený den pro zvoleného uživatele,
 - getNext – vrátí záznam aktuální nebo nejbližší následující lekce na základě identifikátoru uživatele,
 - getStudentsLessons – vrátí lekce studenta,
 - getTeachersLessons – vrátí lekce učitele.
- TermTable:
 - getActiveTermsByUser – vrátí seznam termínů výuky, které jsou platné do současnosti,
 - getByDate – vrátí seznam hodin uživatele určených pro výuku v daný den,
 - getCalendar - vrátí seznam termínů pro týden dle zadaného data a identifikátoru uživatele.
- TransferTable:

- getFrom – vrátí uskutečněné transakce směřující od zvoleného uživatele k jinému uživateli,
- getTo – vrátí uskutečněné transakce směřující k zvolenému uživateli od jiného uživatele,
- getTransfersToFinish – vrátí nezpracované transakce, které souvisejí s již probíhající nebo proběhlou výukou.
- UserTable:
 - fetchAllCountriesUnique – vrátí identifikátory zemí, které má přiřazeny alespoň jeden z uživatelů daného typu,
 - getByCountry – vrátí seznam uživatelů dle zadané země,
 - getByPriceRange – vrátí seznam uživatelů, kteří mají cenu za lekci v zadaném rozpětí,
 - getByType – vrátí seznam uživatelů dle zadaného typu,
 - getItemByEmail – vrátí uživatele dle zadaného emailu uživatele.
- User_CertificateTable:
 - getByUser – vrátí seznam záznamů patřících konkrétnímu uživateli.
- User_SpeakTable:
 - fetchAllUnique – vrátí seznam identifikátorů jazyků, které má přiřazeny alespoň jeden uživatel,
 - getByLanguage – vrátí záznamy dle identifikátoru jazyka,
 - getItems – vrátí záznamy patřící jednomu uživateli.

Také abstraktní modelová třída Model bude obsahovat několik metod:

- exchangeArray – naplní datové atributy třídy ze vstupního pole,
- getArrayCopy – vrátí asociativní pole naplněné daty atributů třídy,
- getInstance – vytvoří novou instanci třídy na základě jiné instance této třídy.

Třídy dědicí z třídy Model budou dále obsahovat specifické metody:

- Currency:
 - getMaxPaypal – vrátí maximální částku, kterou lze poslat přes PayPal,

- getMinPaypal – vrátí minimální částku, kterou lze poslat přes PayPal,
- getRate – vrátí převodní koeficient mezi dvěma měnami.
- Scheduler:
 - setHash – nastaví hash řetězec instancí rezervace termínu.
- User:
 - cryptPassword – zakóduje řetězec hesla,
 - getHashBase – vrátí řetězec pro vytvoření hash řetězce sloužícího k identifikaci uživatele při aktivaci odkazem z emailu,
 - getRandomPassword – vygeneruje náhodné heslo.

2.8 Návrh designu

Kromě toho, že musejí webové stránky bezchybně fungovat, musejí zároveň přijatelně vypadat a to nejen na větších rozlišení monitorů používaných na PC, tak i na nižších rozlišeních, které se používají na dnes hojně používaných tabletech a smartphonech.

Základem návrhu designu jsou tzv. drátěné modely, které mají za úkol názorně definovat obsah jednotlivých stránek a rovněž rozmístění jednotlivých prvků na stránce.

Z drátěných modelů se poté vytvářejí samotné grafické návrhy, které již pouze barevně, tvarově a rozmístěním doladí základ položený drátěnými modely.

Při návrhu designu se dnes zpravidla uvažují 4 skupiny rozlišení, na kterých může uživatel stránky prohlížet, přičemž rozhodující je šířka:

- 1) Menší než 768px (telefony, dále už jen XS),
- 2) Větší nebo rovno 768px a menší než 992px (tablety, dále už jen SM),
- 3) Větší nebo rovno 992px a menší než 1200px (desktopy s menším rozlišením, dále už jen MD),
- 4) Větší nebo rovno 1200px (desktopy s větším rozlišením, dále už jen LG).

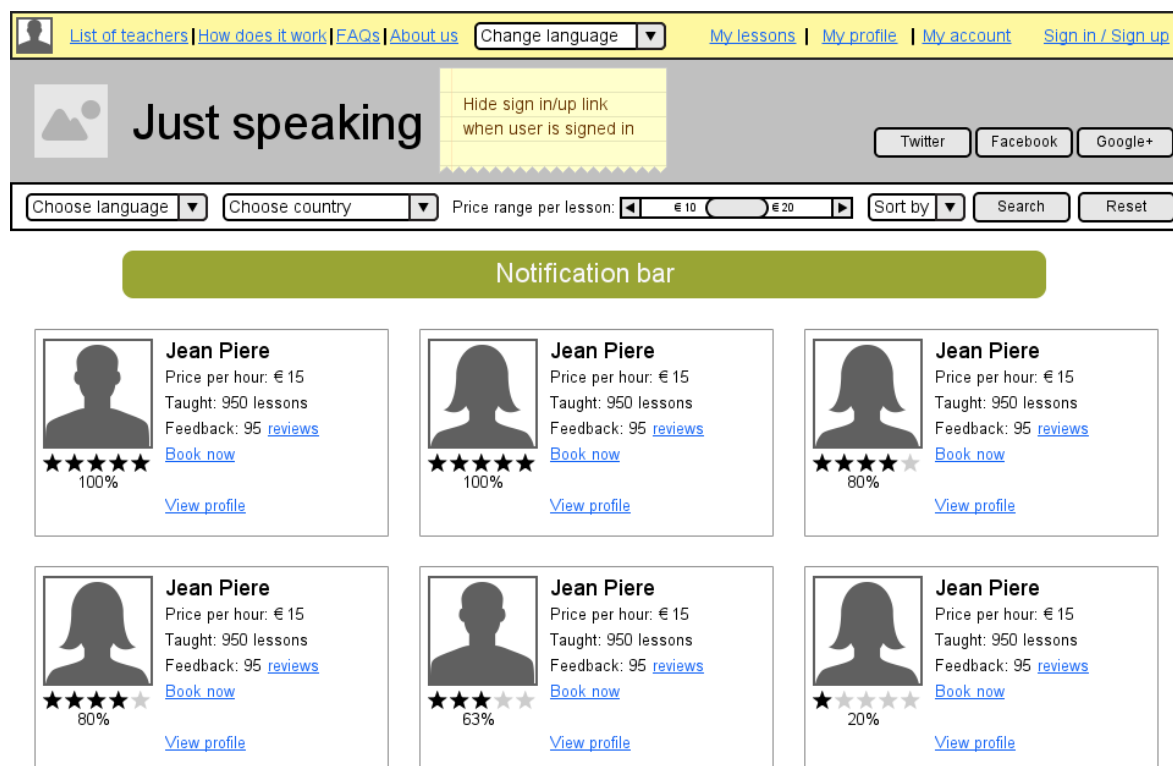
Front-End Framework Bootstrap, jež je využit pro tvorbu tohoto portálu, využívá standardně rozdělení stránky na maximálně 12 sloupců, přičemž jednotlivé prvky mohou na šířku zabírat 1 až 12 sloupců. Pro jednotlivá rozlišení z výše uvedených čtyřech skupin, mohou jednotlivé prvky stránky zabírat na šířku různý počet sloupců, jejichž šířka se automaticky

přizpůsobí šířce okna prohlížeče. Při návrhu rozložení a designu je nutné s použitím těchto dvanácti sloupců počítat a podřizovat mu uspořádání prvků na stránce. Více se lze o návrhu designu webových stránek pro mobilní zařízení dočíst z [18].

2.8.1 Drátěné modely

Portál bude obsahovat několik typových stránek, které lze uspořádat do následující struktury:

- Veřejná část webu,
 - Úvodní stránka,
 - Detail lektora,
 - Články,
 - O nás,
 - Časté dotazy,
 - Jak to funguje,
 - Přihlášení / Registrace,
- Neveřejná část,
 - Můj profil,
 - Můj účet,
 - Moje lekce,
 - Správa administrátorů a požadavků na výběr.



Obr. 5 – Drátěný model úvodní stránky

Úvodní stránka a obecně veřejné stránky portálu budou mít téměř totožnou hlavičku. Bude v ní v horní lince menu stránky rozdělené na dvě části, levou a pravou. V levé části budou položky určené všem návštěvníkům portálu, v pravé části budou položky určené registrovaným návštěvníkům. Pod nabídkou menu bude v levé části logo a název portálu, v pravé části budou tlačítka sociálních sítí umožňujících „lajkování“ jednotlivých stránek.

Na úvodní stránce bude hned pod hlavičkou lišta s filtry, kterými půjde ovlivnit seznam lektorů zobrazených na této stránce. Filtry budou v rozlišeních MD a LG v jedné řádce, v rozlišení SM budou po dvou položkách vedle sebe, v nejmenších rozlišeních bude po jednom filtru na každém řádku. Lektori se budou v MD a LG rozlišení řadit do třech sloupců, na zařízeních s SM rozlišením do dvou a na nejmenších rozlišeních do jednoho sloupce pod sebe.

The image shows a wireframe model of a login form. At the top, the title "Sign in" is centered. Below it, the text "I am" is followed by two radio buttons labeled "student" and "teacher". A yellow callout box points to the "teacher" radio button with the text "In the case of creating new account only". Below the radio buttons, there are three input fields labeled "Email address:", "Password:", and "Password confirmation:". Below the "Password confirmation:" field is a "Login" button. At the bottom, there are two links: "Forgot password" and "Firs time here" (likely a typo for "First time here"). A yellow callout box points to the "Forgot password" link with the text "Sign in form and new account form are similar".

Obr. 6 – Drátěný model přihlašovacího formuláře

Na stránce sloužící pro přihlášení, registraci a zaslání nového přístupového hesla, bude zobrazen jednoduchý formulář, ve výchozím stavu se bude jednat o přihlašovací formulář. Pomocí odkazů ve spodní části formuláře bude možné přejít na formulář registrační, nebo formulář sloužící k zaslání nového hesla pro přístup k účtu.

[List of teachers](#) | [How does it work](#) | [FAQs](#) | [About us](#) | [Change language](#) | [My lessons](#) | [My profile](#) | [My account](#) | [Sign in / Sign up](#)

[Hide sign in/up link when user is signed in](#)

[Twitter](#) | [Facebook](#) | [Google+](#)

Jean Piere

"Teacher's motto"

My characteristics:

Native language: French

Also speaks: English, Spanish

Country: France

Price per hour: € 15

Taught: 950 lessons

Feedback: 95 [reviews](#)

[Previous week](#)

Calendar

[Next week](#)

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
no terms	9am	1pm	9am	no terms	4pm	no terms
	10am	2pm	10am		6pm	
	11am		11am			
	more		more			

Red color will represent booked term

About me

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus aliquam mattis. Ut vulputate eros sed felis sodales nec vulputate justo hendrerit. Vivamus varius pretium ligula, a aliquam odio euismod sit amet. Quisque laoreet sem sit amet orci ullamcorper at ultricies metus viverra. Pellentesque arcu mauris, malesuada quis ornare accumsan, blandit sed diam.

Certifications

Latest reviews

Jean Piere

2013/08/25 9:33pm

Subject: French

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu

★★★★☆ 63%

Jean Piere

2013/08/25 9:33pm

Subject: French

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu

★★★★★ 95%

Jean Piere

2013/08/25 9:33pm


Subject: French

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu

★★★★★ 95%


Obr. 7 – Drátěný model detailu lektora

Hlavní část detailní stránky lektora bude rozdělena v MD a LG rozlišení do dvou sloupců, v ostatních rozlišeních do jednoho. Prvně bude na stránce uvedeno jméno lektora, pod ním jeho motto, fotka a základní charakteristiky lektora. Dále bude uveden detailní popis lektora a seznam jeho certifikátů, který se bude řadit do třech sloupců, po kliku na konkrétní z nich se zobrazí detail certifikátu. V pravé části detailní stránky lektora bude k vidění na větších rozlišeních (na menších ve spodní části) kalendář s možností rezervace lekcí. Rezervační formulář se bude zobrazovat po kliku na konkrétní termín. Pod kalendářem bude seznam doporučení lektora, pokud nějaká má.


[List of teachers](#) | [How does it work](#) | [FAQs](#) | [About us](#)

Change language ▼

[My lessons](#) | [My profile](#) | [My account](#)



Just speaking

Here you can edit your public profile

This is the place where you can edit your name, languages you speak, place where you live, profile picture and many other options.

Basics

Fullname:

Country where you live:

Timezone:


Email address:

Sex: ☒ Male ☐ Female

Languages

Your native language:

Languages you speak:

Profile picture: 

Details

Motto:

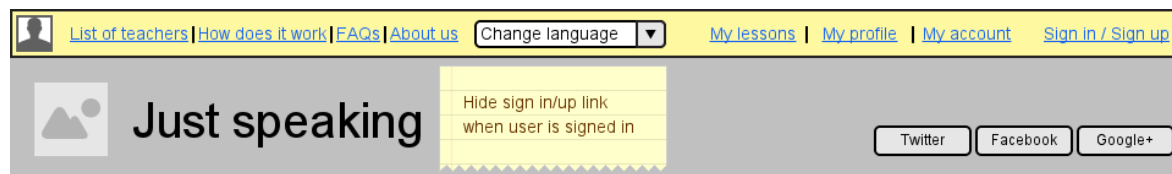
About me:

Multi line textarea

My certifications:

Obr. 8 – Drátěný model editace uživatele

Mezi neveřejné stránky bude patřit například stránka sloužící k editaci profilu uživatele. Hlavička neveřejných stránek se bude od těch veřejných lišit tím, že nebude obsahovat tlačítka sociálních sítí. Svůj obsah bude v MD a LG rozlišeních rozdělovat do dvou sloupců, v levém základní informace, motto, detail a certifikáty, v tom pravém pak nastavení jazyků a profilové fotky. V ostatních rozlišeních budou tyto dva roztaženy přes celou šířku okna prohlížeče a budou tedy pod sebou. Dole bude tlačítko sloužící k uložení informací na této stránce.



Frequency asked questions

Q: Some question?

A: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo lectus, ac blandit elit tincidunt id. Sed rhoncus, tortor sed eleifend tristique, tortor mauris molestie elit, et lacinia ipsum quam nec dui. Quisque nec mauris sit amet elit iaculis pretium sit amet quis magna. Aenean velit odio, elementum in tempus ut, vehicula eu diam. Pellentesque rhoncus

Q: Some other question?

A: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla quam velit, vulputate eu pharetra nec, mattis ac neque. Duis vulputate commodo

Obr. 9 – Drátěný model článku

Další typovou stránkou portálu bude článek, např. seznam často kladených dotazů, ale také další dva, jež budou dostupné odkazem v levé části menu. Stránka článku bude jednoduchá, ve všech rozlišeních využívající jeden sloupec přes celou šířku okna prohlížeče.

Your account

Here you can check your account, view history, withdraw money or deposit.

Account balance

Date: 2013/11/11
Amount: **395.30**
Currency: EUR ▼

Deposit

Amount:

Payout

Amount:
Bank account:
Your password:

History

Outgoing

▼ Date	▼ Time	▼ Subject	▼ Amount	▼ Account ballance	▼ Currency
2013/10/31	1:30pm	Payout (bank account number: IBAN)	400	774.8	EUR

Incoming


▼ Date	▼ Time	▼ Subject	▼ Amount	▼ Account ballance	▼ Currency
2013/11/10	11:22am	Transfer from student 'Fiber'	20.5	395.3	EUR
2013/11/02	2:30pm	Transfer from student 'karol'	18.3	374.8	EUR
2013/10/31	1:30pm	Payout (bank account number: IBAN)	400	774.8	EUR

Waiting

▼ Date	▼ Time	▼ Subject	▼ Amount	▼ Account ballance	▼ Currency
2013/10/31	1:30pm	Payout (bank account number: IBAN)	400	774.8	EUR


Obr. 10 – Drátěný model správy konta

Stránka konta uživatele bude mít horní část v LG a MD rozlišeních rozdělena do dvou sloupců, v levém bude nastavení měny a detail stavu konta, v pravém bude formulář pro zadání požadavku na výběr prostředků a také formulář pro vklad prostředků prostřednictvím PayPal. V dolní části budou tabulky s evidencí provedených operací.


[List of teachers](#) | [How does it work](#) | [FAQs](#) | [About us](#)

Change language ▼

[My lessons](#) | [My profile](#) | [My account](#)



Just speaking

Your lessons

Here you can set up your lessons.

Previous week

Calendar

Next week

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
no terms	9am	1pm	9am	no terms	4pm	no terms
	10am	2pm	10am		6pm	
	11am		11am			
more			more			

Create new term

Create new term

Valid from date:
Valid to date:
Valid from time:
Valid to time:
Days in week:
☒ Mon ☒ Tue ☒ Wed ☒ Thu ☒ Fri ☒ Sat ☒ Sun
Late reservation:
Late storno:

Save term

List of my terms

▼ Date from	▼ Date till	▼ Time from	▼ Time till	▼ Days in week	▼ Late reservation	▼ Late cancellation	▼
3/28/14	7/31/14	8:00 AM	2:00 PM	Saturday	2	2	EDIT / DELETE

List of my lessons

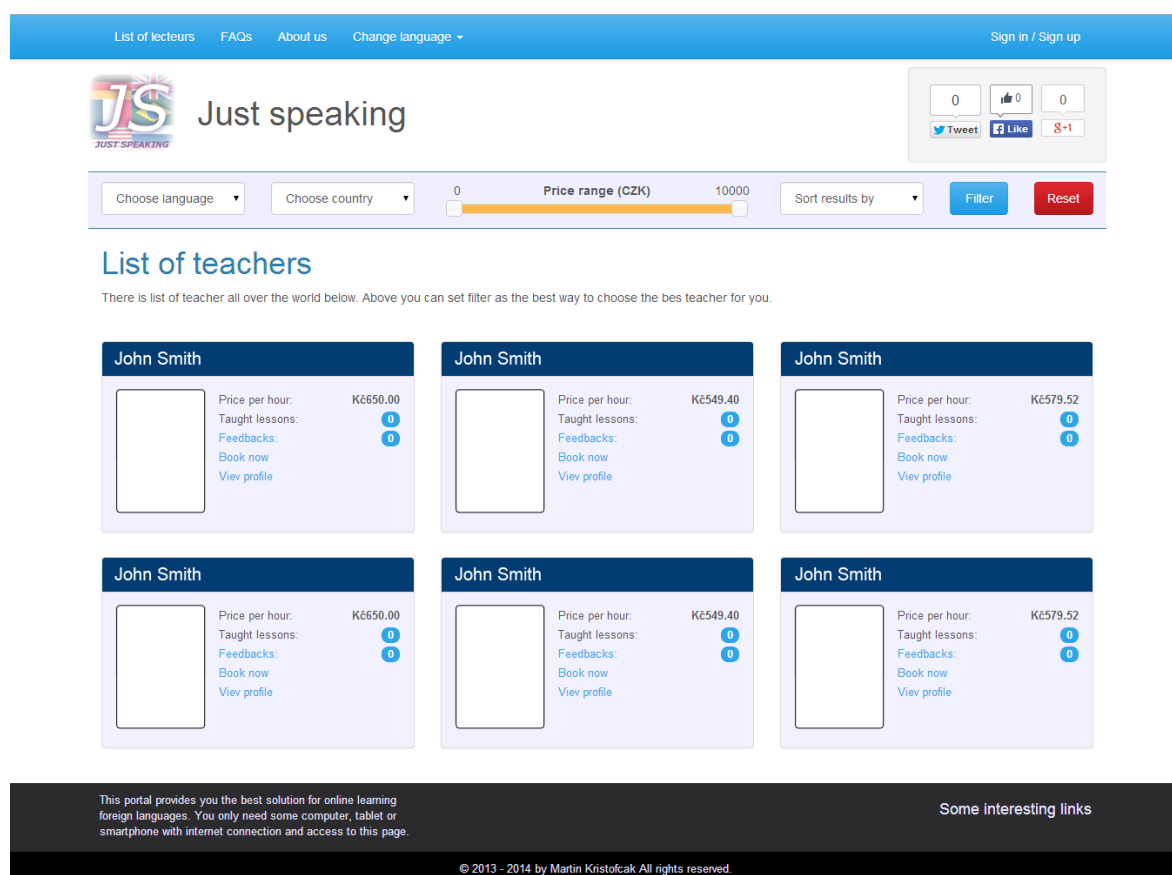
▼ Name of student	▼ Date	▼ Time start	▼ Time end	▼ Language	▼ Message from student	▼
Fantom	3/26/14	10:00 AM	10:59 AM	Czech		CREATE REVIEW
Jill	5/26/14	10:00 AM	10:59 AM	English		CANCEL RESERVATION

Obr. 11 – Drátěný model správy lekcí

Podobně jako stránka konta uživatele, tak i stránka s evidencí lekcí uživatele bude v horní části rozdělena do dvou stejně širokých sloupců v případě vyšších rozlišení a dole bude obsahovat tabulky, tentokrát pro evidenci termínů pro výuku (pouze pro lektory) a pro evidenci lekcí, blížících se a uskutečněných. Lektoři v horní levé části této stránky uvidí kalendář s přehledem svých termínů výuky, v pravé části ve výchozím stavu neuvidí nic, ale po kliku na tlačítko pro vytvoření nového termínu výuky se mu zobrazí formulář pro vytvoření nového termínu výuky. Ve stejném místě se zobrazí formulář pro editaci termínu výuky.

2.8.2 Grafika stránky

Grafická podoba portálu bude velmi jednoduchá a bude vycházet ze základních možností použitého frameworku Bootstrap. Bootstrap nabízí základní sadu CSS pro všechny důležité prvky stránek, tzn. základní sadu typografických stylů, stylů tlačítek, menu, formulářových prvků, dialogových oken a mnoho dalšího. Celou tuto sadu lze navíc stylově přizpůsobit díky možnosti individualizace přímo na stránkách projektu Bootstrap. Navíc je možné omezit sadu komponent, které bude celý balík obsahovat a tím ovlivnit velikost frameworku, jehož soubory si budou muset prohlížeče návštěvníků stránek stahovat. V podstatě si tedy stačí definovat barvy, fonty a velikosti písma, které budou použity, dále stanovit, které komponenty budou v rámci projektu využity, vše nastavit na stránce určené pro individualizaci Bootstrap Front-End Frameworku a nechat si vygenerovat potřebné soubory. Kdykoliv pak bude žádoucí změna barev, fontů či rozměrů, stačí si vygenerovat novou sadu souborů frameworku a tou nahradit sadu stávající.



Obr. 12 – Grafický návrh

2.8.3 Použité prvky

Základní prvky portálu, jejichž vzhled je potřeba zvážit a případně upravit jsou základní typografické prvky, tzn. font, velikost a barva písma, odsazení nadpisů a odstavců, velikost a barva tlačítek, vzhled formulářových prvků, styl tabulek, různých rámečků a boxů, které budou využity.

Tlačítka



Formulář

[Previous](#)
[Next](#)

Legend

Email

Password

☐ Checkbox

Textarea

A longer block of help text that breaks onto a new line and may extend beyond one line.

Radios ☒ Option one is this
☐ Option two can be something else

Selects

1

1
2
3
4

Typografie

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Nullam quis risus eget [urna mollis ornare](#) vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam id dolor id nibh ultricies vehicula.

This line of text is meant to be treated as fine print.

The following snippet of text is **rendered as bold text**.

The following snippet of text is *rendered as italicized text*.

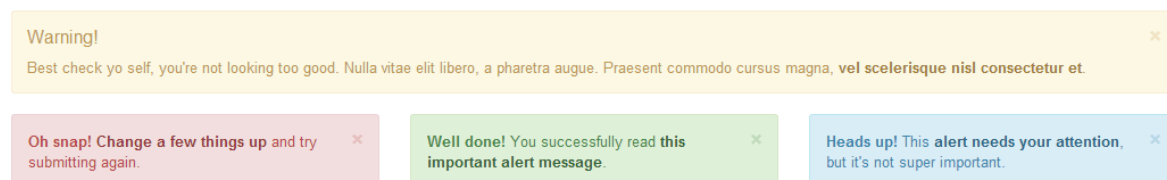
Karty



Tabulky

#	Column heading
1	Column content
2	Column content
3	Column content
4	Column content
5	Column content
6	Column content
7	Column content

Upozornění



Obr. 13 – Návrh grafických prvků

3 VÝVOJ APLIKACE

Základ pro samotný vývoj aplikace byl položen v předchozích kapitolách. Z nich je zřejmé k čemu má aplikace sloužit, jaké má mít funkce, jakých prostředků má využívat a jak má výsledný produkt vypadat.

3.1 Struktura aplikace

Aplikace se bude skládat z PHP, HTML, CSS, JS, obrázků a pomocných souborů. Mezi pomocné soubory je možné zařadit soubor .htaccess obsahující dodatečnou konfiguraci webového serveru Apache pro adresář a podadresáře, ve kterém je obsažen, dále pak soubory s maskou *.mo, které budou obsahovat překlady jednotlivých částí portálu. Všechny soubory budou uspořádány do struktury, která byla navržena v kapitole Bezpečnost.

Do adresáře vendor je nutné umístit adresář ZF2, jenž bude obsahovat knihovny Zend Framework 2. Vedle ZF2 se zde bude nacházet ještě adresář JustSpeaking, který bude obsahovat knihovní třídy vyvíjené aplikace.

Adresář scripts bude obsahovat jediný podadresář sql a v něm bude základní sada sql dotazů, které vytvoří základní strukturu databáze a naplní některé tabulky daty, například tabulku language, country, log_category, review_rate_criterion a user_type.

V adresáři public bude soubor index.php, který bude webový server spouštět při požadavku na zobrazení stránky portálu, dále pak zde bude několik podadresářů:

- css – obsahující soubory s kaskádovými styly,
- img – obsahující soubory s obrázky, např. loga, tlačítka atp.,
- js – obsahující soubory s javascriptovými funkcemi.

Adresář module bude sloužit k ukládání jednotlivých modulů aplikace. Aktuálně vyvíjená aplikace bude mít pouze jeden modul pojmenovaný Application uložený ve stejnojmenném adresáři, který se bude dále členit na adresáře config, language, src a view.

V rámci adresáře data se mohou uchovávat uživatelská a pomocná data, zatím jej bude aplikace využívat pouze pro ukládání uživateli nahraných obrázků.

V adresáři config budou konfigurační soubory, jak již bylo naznačeno a blíže popsáno v kapitole Bezpečnost.

3.1.1 Knihovny aplikace

Jak bylo uvedeno, knihovní třídy aplikace budou uloženy v adresáři vendor a podadresáři JustSpeaking. Z diagramu tříd, který byl ve stejnojmenné kapitole, vyplývá, že bude potřeba napříč aplikací využívat abstraktní třídy Table a Model. Kromě těchto dvou tříd bude knihovna aplikace ještě obsahovat třídu Adapter, která má za úkol specifikovat způsob ověření přihlašovaného uživatele a kterou bude využívat zendovská knihovna AuthenticationService pro realizaci autentizace uživatele. Dále zde bude abstraktní třída Form, která bude předkem všech formulářů aplikace a bude mít za úkol zastřešit základní nastavení realizovaných formulářů. Poslední knihovní třída, kterou bude nutné implementovat je třída Image, která bude obsahovat metody sloužící ke zpracování obrázků používaných uživateli portálu.

3.1.2 Modul Application

Základní a zatím jediný modul aplikace bude dále členěn tak, jak je v aplikacích založených na Zend Framework 2 obvyklé, tedy na podadresáře config, language, src a view. V adresáři config budou konfigurační soubory modulu, v adresáři language budou soubory s příponou mo obsahující překlady do podporovaných jazyků, adresář view bude obsahovat HTML soubory aplikace a to jak soubory definující rozložení stránek, tak i soubory definující obsah jednotlivých stránek, adresář src bude obsahovat zdrojové PHP soubory aplikace, které se budou dále členit do adresářů Controller, Form a Model, případně dalších. V adresáři Controller budou umístěny řídicí PHP skripty jednotlivých částí modulu, v adresáři Form budou skripty sloužící pro sestavení a zpracování jednotlivých formulářů napříč modulem a adresář Model bude zahrnovat všechny modelové třídy a jejich protějšky sloužící pro přístup k datům v databázi, jak byly vydefinovány v kapitole Diagram tříd.

3.2 Příprava layoutů

Layouty neboli rozložení stránek se budou ukládat v rámci modulů do adresáře view a podadresáře layout. Přípona souboru s layoutem stejně jako přípona všech dalších HTML souborů v rámci portálu bude phtml, jak je zvykem u HTML souborů obsahujících PHP.

V rámci této aplikace bude použit pouze jeden layout, neboť rozložení stránek je velmi podobné a vždy se skládá z hlavičky, hlavního obsahu a patičky, přičemž patička bude vždy stejná a hlavička se bude lišit jen minimálně.

Layout bude definovat základní strukturu HTML 5 dokumentu skládající se s definice dokumentu, základní HTML struktury s použitím tagů `<html>`, `<head>`, `<body>`, `<nav>`, `<header>`, `<main>` a `<footer>`. Bude řešit připojení CSS souborů v sekci `<head>` vypisování struktury menu, záhlaví a zápatí stránky, dále připojení javascriptu na konci kódu layoutu.

Javascript bude připojen až na konci kódu, neboť prohlížeče zpracovávají kód stránek shora dolů, proto zpracování javascriptu nebude zdržovat vykreslení stránky.

V rámci layoutu bude rovněž vložen měřicí kód projektu Google Analytics, který má za úkol zaznamenávat provoz na portále a je potřeba jej vložit do každé stránky, tudíž ideálně do souboru layoutu. Google Analytics je javascriptový projekt určený pro sledování trendu návštěvnosti internetových stránek s možností segmentace návštěvníků dle různých kritérií. Více se o projektu dozvíte z [19].

3.3 Příprava modelových tříd

Jak bylo popsáno v kapitole Diagram tříd, modelové třídy budou dědit třídu Model a jejich protějšky pro sloužící pro manipulaci s daty v databázi budou dědit třídu Table.

Pro všechny modelové třídy bude platit to, budou obsahovat sadu atributů, které budou přesně kopírovat atributy neboli sloupce tabulky, kterou modelují.

Všechny třídy vycházející z abstraktní třídy Table budou implementovat tři základní metody:

- `public getItem(array $id)` – metoda s jediným vstupním parametrem `$id`, kterým bude do metody vstupovat asociativní pole identifikátorů požadovaného záznamu. Návrátová hodnota metody bude instance PHP třídy `ArrayObject`. V případě nenalezení záznamu v databázi bude metoda vyvolávat výjimku.
- `public saveItem(Model $item)` – tato metoda bude mít rovněž jeden vstupní parametr, jež bude očekávat instanci potomka třídy Model. Na základě hodnot atributů vstupního parametru dojde k vložení nebo uložení záznamu do databázové tabulky. Pokud ukládání selže, vyvolá se výjimka.
- `public deleteItem(array $id)` – stejně jako metoda `getItem` má tato metoda jeden vstupní parametr obsahující asociativní pole s identifikátory záznamu, která má být smazán. Metoda nebude vracet žádnou hodnotu.

Dále budou jednotlivé modelové třídy obsahovat specifické metody tak, jak je naznačeno v kapitole Diagram tříd.

3.4 Příprava formulářů

Portál bude využívat několika formulářů, všechny z nich budou potomky knihovny třídy projektu Form, která bude dědit ze zendovské třídy Form, která společně s dalšími formulářovými třídami projektu Zend poskytuje spolehlivou a jednoduchou cestu, jak vytvářet a zpracovávat formuláře.

Třída Form z knihovny projektu JustSpeaking má definovaný konstruktor, jehož hlavní úloha spočívá ve vytvoření tří formulářových polí, které budou společné pro všechny formuláře napříč portálem:

- `imp_field_t` – Skryté pole formuláře s tímto jménem bude nést informaci o času, kdy byl formulář sestaven. Tato hodnota bude sloužit k ochraně formuláře.
- `imp_field_h` – Formulářové pole typu text, které má funkci honey potu a rovněž slouží k ochraně formuláře.
- `imp_field_name` – Slouží k identifikaci formuláře.

Dále tato třída definuje metodu `getInputFilter()`, která nastavuje vstupní filtry a validátory výše uvedených polí formuláře.

3.4.1 Filtry

Zend Framework 2 obsahuje sadu filtrů, které mohou automatizovaně filtrovat vstupy formulářových polí. Za všechny je možné jmenovat např. filtr `Alnum` filtrující ze vstupu všechny znaky kromě písmen a čísel, `Digits` filtrující vše kromě čísel nebo `Int` filtrující celočíselnou hodnotu.

3.4.2 Validátory

Zend Framework nabízí vývojáři mnoho předdefinovaných validátorů, které se dají použít v kombinaci s formulářovými prvky a třídou formuláře ke kontrole, zda vstupní hodnoty formuláře odpovídají zadaným požadavkům. Za všechny lze jmenovat např. `Alnum` povolující pouze alfanumerické znaky, `Date` povolující pouze údaje ve formátu data, `Digits` povolující pouze čísla, nebo `EmailAddress` povolující vstup pouze ve formátu emailové adresy.

3.5 Controllery

Controller má za úkol řídit operace předcházející vygenerování finální webové stránky. Tato aplikace bude implementovat několik specifických controllerů, které budou všechny dědit z jedné abstraktní třídy controlleru, která bude sama potomkem zendovské abstraktní třídy controlleru, která poskytuje spoustu užitečných metod. Abstraktní třída controlleru bude mít za úkol několik důležitých funkcí např. sestavení položek menu v závislosti na přihlášeném uživateli či připravení různých pluginů. Zároveň bude definovat několik často využívaných metod napříč controllery, např. `sendMail()` zprostředkující odeslání emailové zprávy nebo `getTable()` vracející požadovanou instanci potomka třídy `Table`.

Aplikace bude obsahovat následující controllery:

- `ArticleController` – Připraví stránky s články např. časté dotazy.
- `CronController` – Připraví stránky prováděcí opakované operace např. stažení aktuálního kurzovního lístku ze stránek ČNB.
- `ImageController` – Připraví požadavky na zobrazení obrázku.
- `IndexController` – Připraví úvodní stránku portálu.
- `LessonController` – Připraví stránku pro realizaci výuky.
- `PayoutRequestController` – Připraví stránky pro správu požadavků na výběr prostředků.
- `PaypalController` – Zpracuje požadavky z portálu PayPal.
- `UserController` – Připraví stránky detailů či správy uživatelů.

3.6 Popis stěžejních funkcí portálu

3.6.1 Registrace / Přihlášení uživatele

Při registraci, přihlášení, ale také při editaci profilu uživatele budou klíčovými operacemi kontrola a vygenerování zašifrovaného hesla uživatele. Jak bylo uvedeno v kapitole Bezpečnost, bude se heslo šifrovat algoritmem BCrypt. Tento algoritmus implementuje zendovská třída `Bcrypt`, kterou pro vygenerování šifrovaného obrazu hesla využije metoda `cryptPassword()` modelové třídy `User`.

3.6.2 Zpracování obrázků

Zpracování obrázků bude zastřešovat knihovní třída Image. Každý obrázek vložený uživatelem do systému ať už profilový obrázek nebo certifikát uživatele bude mít v databázové tabulce picture svůj celočíselný identifikátor. Tento identifikátor bude použit jako název souboru s obrázkem. Třída image bude definovat kromě pomocných metod následující klíčové metody:

- setResource(\$resource) – nastaví obrázek obsažený v jediném atributu metody,
- save() – uloží originál obrázku do adresáře data/contentimgs/orig/ID.jpg, kde ID je celočíselný identifikátor obrázku,
- get(\$width, \$height) – vrátí cestu k obrázku v požadovaném rozlišení, pokud neexistuje, tak jej vytvoří z jeho originálu a uloží do adresáře data/contentimgs/wXXXhYYY/ID.jpg, kde XXX je šířka obrázku v pixelech, YYY je výška obrázku v pixelech a ID je celočíselný identifikátor obrázku,
- delete() – smaže všechny výskyty obrázku napříč podadresáři adresáře data/contentimgs.

3.6.3 Dobití / výběr kreditu

V první verzi portálu bude možné dobit kredit pouze prostřednictvím platebního systému PayPal, výběr se bude realizovat ručně na základě požadavku na výběr kreditu. Do budoucna bude vhodné implementovat další platební možnosti, např. platbu mobilem, převodem na účet nebo přímou platbu kartou. Do budoucna bude rovněž vhodné na základě používané banky vytvořit proces, který vygeneruje data pro hromadné zpracování příkazů k úhradě.

Použití platebního systému PayPal má několik výhod:

1. Nezatěžuje paušálními poplatky,
2. Jednoduchá integrace,
3. Používá se a je použitelný globálně.

Integrace platebního systému spočívá v několika nezbytných krocích:

1. Registrace na www.paypal.com – Je vhodné vytvořit si registraci také na sandbox.paypal.com, aby bylo možné na testovacím kontu integraci odladit.

2. Formulář – Je potřeba vytvořit HTML formulář obsahující všechna potřebná pole pro úspěšné přesměrování uživatele na PayPal pro dokončení platby.

Název pole	Příklad hodnoty pole	Popis pole
Cmd	_xclick	Značí, že návštěvník klikl na tlačítko „koupit nyní“.
business	8MA55U8TG82GE	Třináctimístný kód obchodníka získaný od PayPal.
notify_url	http://www.example.com/en/paypal/ipn	IPN – Instant Payment Notification – adresa, na kterou bude PayPal zasílat potvrzení platby.
item_name	Nákup kreditu	Popis položky, za kterou návštěvník platí.
tax_rate	4	Procentuální vyjádření poplatku za platbu.
currency_code	CZK	Třípísmenný kód měny, ve které je prováděna platba.
custom	just-speaking@gmail.com	Identifikace návštěvníka.
image_url	http://www.example.com/img/logo_half.png	Obrázek, který se bude zobrazovat na stránce platebního portálu.
Lc	cs	Dvoupísmenný identifikátor jazyka
Return	http://www.example.com/en/paypal/success	Stránka, na kterou má PayPal přesměrovat po dokončení platby.
cancel_return	http://www.example.com/en/paypal/error	Stránka, na kterou má PayPal přesměrovat po stornování platby.

Tab. 1 – Seznam proměnných posílaných portálu PayPal

3. IPN – Je nutné vytvořit obsluhu požadavku, který bude PayPal zasílat po úspěšném provedení platby. Data zasláná v požadavku je potřeba poslat zpět na k tomu určenou adresu PayPal domény a pokud PayPal potvrdí správnost údajů, lze považovat platbu za správně provedenou.
4. Nakonec je potřeba zprovoznit dvě adresy, na které bude PayPal směřovat uživatele po odchodu ze stránek PayPal zpět na stránky portálu.

Více informací o použití PayPal jako platební metody webových portálů lze nalézt v [20-21].

3.6.4 Aktualizace kurzovního lístku

Portál bude pracovat s finančními prostředky ve formě kreditu, který budou mít možnost uživatelé vést v různých měnách. Aby mohl být realizován převod kreditu od jednoho uživatele k druhému v případě, že vedou kredit v různých měnách, je nutné ukládat kromě seznamu měn i jejich kurzy.

Kurzy je nutné aktualizovat na denní bázi. K tomuto budou využívána data poskytovaná Českou národní bankou na adrese http://www.cnb.cz/cs/financni_trhy/devizovy_trh/kurzy_devizoveho_trhu/denni_kurz.txt.

Na této adrese je každý den po 14:30 k dispozici aktualizovaný lístek kurzu české měny k cizím měnám.

Skript, který bude provádět zpracování kurzovního lístku, bude k dispozici na adrese <http://www.example.com/en/cron/currency>. Spuštění bude možné vyvolat buď ručně, nebo automatizovaně voláním této stránky pomocí CRON.

3.6.5 Realizace video hovoru

K realizaci video hovoru bude použito projektu WebRTC, konkrétně jeho aplikaci v podobě knihovny SimpleWebRTC, o kterém je možné se více dozvědět na adrese <http://simplewebrtc.com>, který je zatím podporován nejnovějšími desktopovými verzemi internetových prohlížečů Google Chrome a Mozilla Firefox. Uživatelé disponující jinými webovými prohlížeči budou muset využít k realizaci video hovoru jiného software např. Skype. Pro vytvoření spojení je potřeba vytvořit dva HTML blokové objekty a jejich identifikátory včetně identifikátoru komunikace předat inicializační funkci komunikátoru. Každá rezervace termínu výuky bude mít v databázi svůj jednoznačný identifikátor zašifrovaný algoritmem BCrypt. Pomocí tohoto identifikátoru se bude identifikovat spojení mezi studentem a lektorem.

3.7 Obsluha JavaScriptem

K obsluze akcí na straně klienta se bude využívat javascriptová knihovna JQuery a další knihovny na ní založené. Javascript vytvořený pro účely této aplikace bude uložen v adresáři `/public/js/page` v podadresářích pojmenovaných podle controllerů, jejichž stránky budou

obsluhovat. Názvy jednotlivých souborů s javascriptem budou vycházet z názvů metod controllerů, které budou stránky využívající tyto skripty obsluhovat.

3.7.1 JQuery

Tato knihovna v kombinaci s některými volitelnými moduly bude používána pro základní manipulaci s objekty na stránce, dále pak pro validaci formulářů na straně klienta, řešení dialogových oken, spouštění AJAX požadavků apod.

3.7.2 RequireJS

Knihovna sloužící k načítání javascriptových souborů a knihoven. Hlavním přínosem RequireJS je modulárnost, tedy možnost rozdělit svůj javascriptový kód na moduly a ty načítat snadno na jednotlivých stránkách podle potřeby.

3.7.3 TinyMCE

Tento WYSIWYG editor se bude hodit při editaci detailních informací uživatele a také při editaci článku.

3.7.4 Lightbox

Tato knihovna umožňuje snadné listování obrázky a i jejich služeb bude na portále využito, konkrétně při realizaci prohlížení certifikátů uživatele.

4 TESTOVÁNÍ APLIKACE

Testování aplikace bude vycházet z funkčních požadavků, které by měly být splněny a jednotlivé funkce by měly pracovat bezchybně.

4.1 Testovací scénáře

Test se bude skládat z několika ucelených bloků, v rámci nichž se otestují funkčně blízké celky.

4.1.1 Registrace a editace uživatele

V tomto kroku se provede registrace nového studenta, lektora i administrátora, vyzkouší se funkce validátorů, zkontroluje se odesílání potvrzovacích emailů, deaktivace profilu a také přihlášení, odhlášení a zpracování požadavku na zaslání nového hesla.

4.1.2 Správa konta

Pro účely testu vkladu prostředků na konto uživatele bude sloužit testovací účet na `sandbox.paypal.cz`. Přepnutí mezi testovacím a ostrým PayPal účtem bude probíhat v konfiguračním souboru portálu tedy `config/application.config.php`.

V rámci testu výběru prostředků bude potřeba otestovat zadání požadavku a jeho potvrzení administrátorem.

Kromě vkladu a výběru bude nutné otestovat převod prostředků mezi konty uživatelů, správná konverze mezi měnami a aktualizace kurzovního lístku.

4.1.3 Výuka

V tomto kroku bude potřeba otestovat založení, úpravu a zrušení termínu výuky, rezervaci a zrušení rezervace termínu výuky, spuštění výuky, hodnocení výuky a to včetně všech potvrzovací a notifikačních emailů, které s těmito úkony souvisejí.

4.1.4 Články, sociální sítě, statistiky

V tomto závěrečném kroku se otestuje zobrazení a editace článků a „lajkování“ jednotlivých stránek na sociálních sítích.

5 NASAZENÍ APLIKACE DO PROVOZU

Pro zprovoznění aplikace bude nutné provést několik kroků.

5.1 Nahrání souborů aplikace

Do adresáře určeného pro soubory webové aplikace bude potřeba nahrát všechny soubory.

5.2 Konfigurace

Bude potřeba nastavit konfigurační soubory v adresáři config/autoload. V souboru config/autoload/local.php je nutné zadat název databáze, v souboru config/autoload/global.php je nutné nastavit přístupové údaje k databázi.

5.3 Příprava databáze

Pro naplnění databáze bude nutné spustit jednotlivé skripty z adresáře scripts/sql a to v abecedním pořadí. Následně bude potřeba spustit aktualizaci kurzů měn.

5.4 Nastavení serveru

Buď pomocí .htaccess souboru nebo pomocí konfiguračních souborů serveru bude potřeba přesměrovat všechny požadavky na doménu portálu na soubor public/index.php.

5.5 Změna hesla hlavního administrátora

Hlavní administrátor, jehož účet bude vytvořen inicializačními SQL dotazy, bude mít uživatelské jméno root@just-speaking.com a heslo r00tr00t. Toto heslo je z bezpečnostních důvodů nutné ihned po spuštění portálu změnit.

6 DALŠÍ VÝVOJ PORTÁLU

Jak bylo uvedeno, portál byl nasazen do zkušebního provozu na adrese www.just-speaking.com a na této adrese bude také provozován.

Před uvedením aplikace do ostrého provozu bude nutné pečlivě zpracovat copywriting klíčových textů portálu, tedy třech hlavních článků, hlavních popisných textů a textů emailových zpráv. Tyto texty bude potřeba postupně překládat do dalších jazyků, ve kterých bude portál provozován, zprvu to budou jazyky český, slovenský a anglický, v další fázi budou velmi pravděpodobně následovat německý, francouzský a španělský, postupně budou doplňovány i další jazykové verze portálu.

Po uvedení aplikace do provozu bude nutné začít se sbíráním registrací lektorů cizích jazyků. Pro tento úkol bude potřeba zpracovat analýzu mapující možnosti, jak jich na portál efektivně přivést dostatečné množství. Současně bude provedena SEO analýza a optimalizace portálu pro internetové vyhledávače. Kromě organického zviditelnění portálu prostřednictvím vyhledávačů bude vhodné zvážit investici do reklamy.

Dále bude nutné se zaměřit na postupné zprovoznění real-time komunikace v co nejširším spektru internetových prohlížečů. Rychlost zpracování tohoto úkolu bude záviset na rychlosti integrace potřebných prostředků do jednotlivých prohlížečů jejich vývojáři.

Dalším úkolem bude analýza a integrace dalších platebních metod tak, aby byl pokryt dostatečně velký záběr potenciálních uživatelů portálu.

Průběžně bude potřeba zlepšovat zabezpečení aplikace. Jako první bezpečnostní prvek bude vhodné implementovat zabezpečenou komunikaci s využitím protokolu HTTPS a šifrované komunikace na stránkách, na kterých bude docházet k odesílání citlivých dat z klientského počítače na server či naopak, především na stránkách sloužících k autentizaci uživatele a stránkách pro správu konta.

ZÁVĚR

Cílem této práce bylo vyvinout webový portál sloužící jako podpora online výuky cizích jazyků. V teoretické části této práce byly navrženy a stručně popsány technologie, které byly následně využity k vývoji aplikace a jejichž podmínkou pro využití byla možnost jejich bezplatného využití pro realizaci projektu tohoto typu.

V praktické části byla zpracována úvodní studie následovaná podrobnou analýzou požadavků kladených na tuto aplikaci, rozbořem otázek bezpečnosti aplikace, možných rizik a způsobů jak je řešit. Pomocí UML diagramů byl navržen model případů užití a diagram tříd, dále byla navržena struktura relační databáze. Samotnému programování aplikace ještě předcházela návrh designu, který se skládal z návrhu drátěných modelů a finálního grafického řešení.

Jádrem aplikace se stal Zend Framework 2, jenž dobře posloužil pro vývoj tohoto portálu a svojí širokou škálou modulů je vhodný pro použití i u mnohem větších projektů. Kromě tohoto frameworku bylo využito dalších aplikací třetích stran např. JQuery, RequireJS, TinyMCE, Lightbox, Bootstrap či SimpleWebRTC.

Vyvinutá aplikace splňuje všechny kladené požadavky a v současné době byla nasazena do zkušebního provozu na adrese www.just-speaking.com. Tímto ovšem její vývoj zdaleka nekončí. Kromě velkého úsilí, které bude nyní kladeno na získání co největšího povědomí mezi lektory i potenciálními studenty, bude potřeba dále pracovat na jejím vývoji, pilovat funkční i grafickou stránku aplikace a v neposlední řadě také copywriting tak, aby používání portálu bylo srozumitelné, bezproblémové a pohodlné. Dále bude potřeba neustále pracovat na zabezpečení portálu, co nejdříve opatřit certifikát od certifikační autority k ověření domény, aby bylo možno začít bezproblémově využívat šifrovanou komunikaci pomocí protokolu HTTPS na stránkách, kde dochází k odesílání citlivých dat.

ZÁVĚR V ANGLIČTINĚ

This thesis was aimed to create the web portal to support online teaching foreign languages. In the theoretical part of this thesis there were chosen and described technologies which were used for the developing of the application. It was necessary to use free tools and technologies.

In the practical part there was created an initial study followed by the detailed analysis of requirements. Next there were analyzed security questions about the using of this application and there were designed some solutions how to prevent the treat of attack. Application requirements were modeled by UML USE CASE diagram, application structure were modeled by UML class diagram and the model of the database. There was also created some wireframes and designs before writing the code.

Zend Framework 2 becomes the core of the application. It offers various modules so it's suitable for creating more complex applications than this one. Instead of ZF2 there were used many other third party apps such as JQuery, RequireJS, TinyMCE, Lightbox, Bootstrap or SimpleWebRTC in this application.

Final application meets all requirements and was launched online at www.just-speaking.com. By this thesis work on this application does not end. There is a lot of work to do, for example to propagate the application, to collect teachers' registrations, to improve design and functionality, to do copywriting etc. It's necessary to do the using of this application clear, smooth and comfortable. Finally this application should use HTTPS protocol to send sensitive information so in the future it's necessary to get SSL certificate and crypt this data.

SEZNAM POUŽITÉ LITERATURY

- [1] HOGAN, Brian P. HTML5 a CSS3. Brno: Computer Press, 2011. ISBN 978-80-2513-576-1.
- [2] ACHOUR, Mehdi, Friedhelm Betz, Antony Dovgal, Nuno Lopes, Hannes Magnusson, Georg Richter, Damien Seguy a Jakub Vrana. PHP Manual. Hypertext Preprocessor. [online]. [cit. 2014-01-19]. Dostupné z: <http://www.php.net/manual/en>.
- [3] VRÁNA, Jakub. 1001 tipů a triků pro PHP. Brno: Computer Press, 2011. ISBN 978-80-251-2940-1.
- [4] KOFLER, Michael a Bernd Öggl. PHP 5 a MySQL 5: průvodce webového programátora. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.
- [5] MySQL Reference Manuals. Developer Zone. [online]. [cit. 2014-01-19]. Dostupné z: <http://dev.mysql.com/doc>.
- [6] KOFLER, Michael. Mistrovství v MySQL 5. Brno: Computer Press, 2007. ISBN 978-80-251-1502-2.
- [7] Programmers Reference Guide of Zend Framework 2. Zend Framework. [online]. [cit. 2014-01-19]. Dostupné z: <http://framework.zend.com/manual/2.2/en/index.html>.
- [8] SHASANKAR, Krishna. Zend Framework 2.0 by Example: Beginner's Guide. United States: Packt Publishing Limited, 2013. ISBN 978-17-821-6192-9.
- [9] Getting started. Bootstrap. [online]. 2014 [cit. 2014-04-10]. Dostupné z: <http://getbootstrap.com/getting-started>.
- [10] jQuery API Documentation. jQuery. [online]. [cit. 2014-01-19]. Dostupné z: <http://api.jquery.com>.
- [11] CHAFFER, Jonathan, Karl Swedberg. Mistrovství v jQuery. Brno: Computer Press, 2013. ISBN 978-80-251-4103-8.
- [12] jQuery. Brno: Computer Press, 2010. ISBN 978-80-251-3152-7.
- [13] CONOLLY, Thomas. jQuery. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- [14] ODELL, Den. JavaScript. Brno: Computer Press, 2010. ISBN 978-80-251-2733-9.

- [15] LACKO, Ľuboslav. Ajax. Brno: Computer Press, 2008. ISBN 978-80-251-2108-5.
- [16] WebRTC API. Your Web, documented. [online]. [2014] [cit. 2014-01-19]. Dostupné z: <http://docs.webplatform.org/wiki/apis/webrtc>.
- [17] NEUSTADT, Ila, Jim Arlow. UML 2 a unifikovaný proces vývoje aplikací: Objektově orientovaná analýza a návrh prakticky. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [18] CASTLEDINE, Earle, Myles Eftos, Max Wheeler. Vytváříme mobilní web a aplikace pro chytré telefony a tablety. Brno: Computer Press, 2013. ISBN 978-80-251-3763-5.
- [19] CLIFTON, Brian. Google Analytics. Brno: Computer Press, 2009. ISBN 978-80-251-2231-0.
- [20] PayPal APIs Getting Started Guide - PayPal Developer. PayPal Developer [online]. [2014] [cit. 2014-04-19]. Dostupné z: https://developer.paypal.com/docs/classic/api/gs_PayPalAPIs.
- [21] BALDERAS, Michael. PayPal APIs Up and Running. United States: O'Reilly Media, Inc, USA, 2011. ISBN 978-14-4939-612-1.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SVN	Subversion. Jedná se o systém pro správu a verzování zdrojových kódů
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
cookies	Malé množství dat, která jsou serverem posílána prohlížeči a následně ukládána do klientského počítače
CRON	Plánovač úloh UNIXový a Linuxových systémech
CSS	Cascading Style Sheets
copywritting	Tvůrčí činnost spočívající v tvorbě čtivých a poutavých textů
framework	Softwarová struktura sloužící jako podpora při vývoji dalšího software
front-end	Pojem používaný pro označení části webu viditelné běžnými uživateli
GIT	Systém pro správu a verzování zdrojových kódů
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
LG	Large – použito pro označení zobrazovacích zařízení s vysokým rozlišením
MD	Medium – použito pro označení zobrazovacích zařízení se středním rozlišením
MVC	Model View Controller
MySQL	Jeden ze systémů řízení báze dat
NF	Normální forma. Zde použita při definici normálních forem struktury databáze
PC	Personal computer
PHP	Hypertext Preprocessor
SEO	Search Engine Optimalization
SM	Small – použito pro označení zobrazovacích zařízení s malým rozlišením
SQL	Structured Query Language neboli strukturovaný dotazovací jazyk
SŘBD	Systém řízení báze dat
request	Požadavek na server
UML	Unified Modeling Language
WYSIWYG	What you see is what you get – označení skupiny textových editorů
XS	Extra small - použito pro označení zobrazovacích zařízení s velmi malým rozlišením
ZF2	Zend Framework 2

SEZNAM OBRÁZKŮ

Obr. 1 - Aktéři systému.....	38
Obr. 2 – USE CASE diagram	39
Obr. 3 – Datový model	40
Obr. 4 – Diagram tříd.....	43
Obr. 5 – Drátěný model úvodní stránky	48
Obr. 6 – Drátěný model přihlašovacího formuláře	49
Obr. 7 – Drátěný model detailu lektora	50
Obr. 8 – Drátěný model editace uživatele.....	51
Obr. 9 – Drátěný model článku.....	52
Obr. 10 – Drátěný model správy konta.....	52
Obr. 11 – Drátěný model správy lekcí	53
Obr. 12 – Grafický návrh.....	54
Obr. 13 – Návrh grafických prvků.....	55

SEZNAM TABULEK

Tab. 1 – Seznam proměnných posílaných portálu PayPal	62
--	----