



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

APLIKACE PRO SPRÁVU PŘEKLADŮ PRO REDAKČNÍ SYSTÉMY

TRANSLATING MANAGEMENT SYSTEM FOR PUBLISHING SYSTEMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PAVEL ŠTEFANÍK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ONDŘEJ KRAJSA, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Pavel Štefaník

ID: 78725

Ročník: 2

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Aplikace pro správu překladů pro redakční systémy

POKYNY PRO VYPRACOVÁNÍ:

Podrobně se seznámte s možnostmi multiplatformního programování pro Internet. Ve Vámi vybraném vývojovém nástroji navrhnete a realizujete webovou aplikaci pro pokročilou správu textů a jejich překladů. Pro aplikaci navrhnete modul pro import/export textů ze zvoleného redakčního systému.

DOPORUČENÁ LITERATURA:

- [1] KOSEK, Jiří. PHP :Tvorba interaktivních internetových aplikací. Podrobný průvodce /Praha :Grada Publishing,1998. 1. vyd. 490 s. ISBN 8071693731
- [2] NARAMORE, Elizabeth. PHP5, MySQL, Apache: vytváříme webové aplikace / Vyd. 1. Brno : Computer Press, 2006. 813 s. ISBN 80-251-1073-7
- [3] KOFLER, Michael; ÖGGL, Berndt .PHP 5 a MySQL 5 : průvodce webového programátora, Vyd. 1. Brno : Computer Press, 2007. 607 s., ISBN: 978-80-251-1813-9

Termín zadání: 10.2.2014

Termín odevzdání: 30.5.2014

Vedoucí práce: Ing. Ondřej Krajsa, Ph.D.

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Diplomová práce se zabývá vytvořením aplikace pro správu překladů pro redakční systémy. Aplikace bude pomáhat vedoucímu překladateli distribuovat požadavky na překlad překladatelům a následně korektorům. Vedoucí překladatelé mohou přímo v aplikaci importovat články z redakčního systému *WordPress* a po překladu ho exportovat. Redaktoři mohou v přímo v aplikaci články psát. Za schvalování účtů a jim přidělených práv budou zodpovědní uživatelé s účtem administrátora.

Klíčová slova:

Aplikace, redakční systém, publikační systém, HTML, CSS, PHP, SQL, web, překlad, korektura, redaktor, *WordPress*

Abstract

The diploma thesis is focused on creation of application for managing translation for content management systems. Application will help translation leader with distribution of requests for translation to translators and then to proofreader. Translation leader will be able to import articles from WordPress and export after translation. Editor will be able to write articles directly in application. For managing and granting access will be responsible user with administration rights.

Key words:

Application, content management systems, HTML, CSS, PHP, SQL, web, translation, proofreading, editor, *WordPress*

ŠTEFANÍK, P. *Aplikace pro správu překladů pro redakční systémy*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2014. 63 s. Vedoucí diplomové práce Ing. Ondřej Krajsa, Ph.D..

Prohlašuji, že svou diplomovou práci na téma Aplikace pro správu překladů pro redakční systémy jsem vypracoval samostatně pod vedením vedoucího semestrální práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této semestrální práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

podpis autora

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Ondřejovi Krajsovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracovávání mé diplomové práce.

V Brně dne

.....

podpis autora

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

OBSAH

Seznam obrázků	10
Seznam výpisů kódu	11
Úvod.....	12
1. Redakční systém.....	14
1.1. Definice redakčního systému	14
1.2. Základní funkce redakčního systému	14
1.3. Používané redakční systémy	15
1.3.1. WordPress	15
1.3.2. Joomla!	15
1.3.3. Drupal	16
1.3.4. GetSimple CMS	16
1.3.5. Ostatní	16
2. Použitý software a programovací jazyky	17
2.1. Apache	17
2.2. HTML	18
2.3. CSS.....	19
2.4. PHP.....	20
2.5. Smarty.....	21
2.6. MySQL	23
2.7. Zebra_database	24
3. Aplikace pro Správu překladů	26
3.1. Účel aplikace.....	26
3.2. Popis aplikace	26
3.3. Návrh aplikace.....	26
3.3.1. Prezentační vrstva	28
3.3.2. Logická vrstva	28
3.3.3. Databázová vrstva	29

3.4. Přístup do aplikace.....	30
3.5. Uživatelské účty	30
3.5.1. Administrátor systému.....	31
3.5.2. Správce překladu	31
3.5.3. Překladačel	32
3.5.4. Korektor	33
3.5.5. Redaktor.....	33
3.6. Rozbor kódu	34
3.6.1. Přístupový bod aplikace	34
3.6.2. Konfigurační soubor	34
3.6.3. Aplikační soubor.....	35
3.6.4. Soubory s uživatelskými kódy	35
3.6.5. Soubory s šablonami pro Smarty	36
3.6.6. Soubory knihoven	36
3.7. Struktura databáze	37
3.7.1. Tabulka article.....	37
3.7.2. Tabulka article_translate	38
3.7.3. Tabulka article_wp	38
3.7.4. Tabulka member	39
3.7.5. Tabulka member_accounts	40
3.7.6. Tabulka member_languages.....	40
3.7.7. Tabulka language.....	41
3.7.8. Tabulka account_list	41
3.8. Webhosting	42
3.9. Bezpečnost	42
3.9.1. Zabezpečení účtů.....	42
3.9.2. Zabezpečení hesla	42
3.9.3. Zabezpečení PHP	43
3.10. Uživatelský popis.....	43

3.10.1. Administrátor	45
3.10.2. Správce překladu	46
3.10.3. Redaktor.....	47
3.10.4. Překladač	48
3.10.5. Korektor.....	50
3.11. Instalace a použití	51
3.11.1. Prerekvizity.....	51
3.11.2. Instalace.....	51
3.11.3. Vytvoření MySQL databáze	51
3.11.4. Vytvoření administrátor uživatele	52
3.11.5. Vytvoření testovacích uživatelů.....	52
3.11.6. Zprovoznění emailového oznamování	52
3.11.7. Import testovacích článků.....	52
Závěr	53
Použitá literatura	55
Seznam použitých zkratk	57
PŘÍLOHA A – SQL kód pro přípravu databáze.....	58
Seznam příloh na CD	63

SEZNAM OBRÁZKŮ

obr. 2-1: Zastoupení webových serverů [9]	17
obr. 3-1: Propojení vrstev	27
obr. 3-2: Základní popis aplikace.....	27
obr. 3-3: Popis uživatelských práv	29
obr. 3-4: Login	43
obr. 3-5: Úvodní menu po přihlášení	43
obr. 3-6: Registrace	44
obr. 3-7: Změna hesla	44
obr. 3-8: Administrátora - list uživatelů	45
obr. 3-9: Úprava jazyků aplikace	45
obr. 3-10: Správce překladu – list článků aplikace	46
obr. 3-11: Správce překladu – list importovaných článků z <i>WordPress</i>	47
obr. 3-12: Redaktor – vytvoření článku.....	48
obr. 3-13: Redaktor – list článků	48
obr. 3-14: Překladař – list článků pro překlad.....	48
obr. 3-15: Překladař – list článků pro překlad.....	49
obr. 3-16: Překladař – list jazyků.....	49
obr. 3-17: Korektor – články na korekturu	50
obr. 3-18: Korektor – korektura.....	50

SEZNAM VÝPISŮ KÓDU

2-1: Struktura HTML stránky [16]	19
2-2: Struktura CSS kódu [17].....	19
2-3: Připojení CSS stylu do HTML stránky	20
2-4: Porovnání kódu <i>Smarty</i> a PHP [14]	22
2-5: Použití <i>Zebra_database</i>	24
2-6: <i>Zebra_database</i> SELECT požadavek	24
2-7: <i>Zebra_database</i> INSERT požadavek.....	25
2-8: <i>Zebra_database</i> UPDATE požadavek	25
2-9: <i>Zebra_database</i> QUERY požadavek	25
3-1: Kód souboru index.php	34
3-2: Kód souboru config.php	34
3-3: Kód pro vytvoření tabulky article	37
3-4: Kód pro vytvoření tabulky article_translate.....	38
3-5: Kód pro vytvoření tabulky article_wp.....	39
3-6: Kód pro vytvoření tabulky member.....	39
3-7: Kód pro vytvoření tabulky member_account	40
3-8: Kód pro vytvoření tabulky article_languages.....	41
3-9: Kód pro vytvoření tabulky language	41
3-10: Kód pro vytvoření tabulky account_list.....	41

ÚVOD

V dnešní době informačních technologií je publikování článků, obrázků a ostatních multimédií na webových stránkách velmi žádané. Internetový obsah se velmi rychle rozrůstá, a to nejen kvůli web developerům, ale i méně zkušeným uživatelům, kterým pomáhají s publikováním webového obsahu rozličné aplikace. Pro usnadnění publikace textů, obrázků a blogů se začaly vytvářet redakční systémy, které dovolují i méně zkušeným uživatelům jednoduché vkládání článků na web. Internet spojuje velké množství uživatelů do jednoho celku, ale tím, že uživatelé jsou z celého světa, nemusí mít společný jazyk a domluvit se. Proto je potřeba překladatelů a tlumočnicků, kteří tuto jazykovou bariéru pomáhají překonat.

Cílem mé diplomové práce je seznámení se s možnostmi multiplatformního programování pro internet a navrhnout a realizovat webovou aplikaci pro pokročilou správu textů a jejich překladů. Pro naprogramování aplikace jsem si zvolil programovací jazyk PHP, pro webovou prezentaci zde bude využit jazyk HTML/CSS a pro ukládání dat zde bude využita databáze MySQL, která je volně dostupná pro použití.

Vytvořená aplikace bude koordinovat a zpřehledňovat práci na překladech článků z redakčních systémů do různých jazyků. Aplikace by měla být schopná jak články importovat z určeného redakčního systému, tak by se v ní měly dát přímo vytvářet. Redakční systém pro import a export článků jsem zvolil jeden z nejpoužívanějších, a to je *WordPress*. Jedná se o velmi rozšířený redakční systém a je dostupný k použití zdarma.

Aplikace by měla fungovat jako úložiště článků, které budou uživatelům zobrazovány v listu. V aplikaci budou definovány specifické uživatelské role a každá z nich bude mít rozličná práva a povinnosti při práci. Uživatelské role, které budu v aplikaci nazývat uživatelskými účty, se mohou překrývat. I na to by měla být aplikace připravená. Kromě toho musí být také schopná, aby v ní uživatel nastavil jazyky, které má ovládat a nabízet pro překlad článků. Texty a články se v aplikaci budou překládat manuálně uživateli, kteří budou mít účet překladatele. Přeložené články pak následně projdou korekturou, kvůli správnosti jejich překladu. Aby byla korektura účinná, je potřeba zajistit, aby se v aplikaci nepřekrývaly různé role. Proto bude potřeba nastavit, aby jeden uživatel, který bude článek překládat, nemohl

překlad také zkontrolovat. Po zkontrolování překladu bude článek hotový a připravený pro export do redakčního systému *WordPress*. Články bude možné z aplikace vyexportovat jednotlivě nebo označit více článků zároveň, a ty pak vložit do jednoho výstupního souboru, který ponese příponu *.xml*.

1. REDAKČNÍ SYSTÉM

1.1. Definice redakčního systému

Redakční systém je aplikace spravující informační data různého charakteru, které pak následně prezentuje. V různých zdrojích se redakční systém označuje různými slovními spojeními či zkratkami jako RS, publikační systém a CMS Kontent Management System. Redakční systémy mají vstupní rozhraní, které se používá pro vkládání dat jak v textové, tak v jiné multimediální formě, a výstupní rozhraní, které se používá pro jejich prezentaci. Většina redakčních systémů obsahuje pokročilejší funkce, například anketní subsystémy, které je potřeba obsluhovat, a proto zde byl požadavek na určité administrační prostředí.

Administrační prostředí se nazývá backend. Používá se pro konfiguraci celého redakčního systému a mělo by také plnit určité části redakčního systému daty. Administrační prostředí se většinou ovládá přes webové stránky, které jsou chráněny autorizačním systémem, jež omezuje přístup k určitým částem redakčního systému podle přidělených rolí. [1]

1.2. Základní funkce redakčního systému

Mezi základní funkce redakčního systému patří: [1]

- Správa článků – přidávání, úprava a mazání článků
- Inteligentní správce uživatelů – delegace přístupových práv uživatelů
- Schopnost konfigurace zobrazení a rozložení obsahu na zobrazované stránce
- Definice rubrik – kategorizování článků do rubrik
- Vyhledávací subsystém – možnost vyhledat určitý článek dle parametrů
- Anketní subsystém – funkce pro vytvoření ankety
- Novinkový subsystém – přehled novinek v redakčním systému
- Komentářový subsystém – možnost vkládání komentářů
- Sekce pro download souborů

1.3. Používané redakční systémy

Dále v tomto odstavci rozebereme nejznámější a nepoužívanější redakční systémy. [2]

1.3.1. WordPress

Redakční systém *WordPress* je založený na PHP a je k dispozici zdarma na stránce www.wordpress.org.

WordPress je open source projekt, který je vyvíjen pod licencí GNU GPL. Open source značí projekt s otevřeným kódem, jež je vyvíjen komunitou programátorů, kteří na něm pracují zdarma, a proto je také dostupný zdarma. První verze *WordPress* byla zveřejněna v roce 2003. *WordPress* je nyní nepoužívanější blogovací systém. Využívá se na více než 60 milionech webových stránek. Poslední *WordPress* dostupný ve verzi 3.7.1.

Mezi výhody *WordPress* patří velká komunita vývojářů se značným množstvím dokumentace a návodů, bezplatné a placené pluginy, specializovaná témata pro tvorbu libovolných webů a uživatelsky přátelská nástěnka pro správu obsahu. Jako nevýhody zde můžeme uvést zbytečnou pokročilost pro základní weby, jeho zranitelnost bez přidání bezpečnostních opatření a chybějící oficiální podporu (která existuje jen ve formě uživatelských fór, kde se správce nebo uživatel nemusí dočkat oficiální odpovědi). [2] [3] [4]

1.3.2. Joomla!

Redakční systém *Joomla!* je napsaný v PHP a je možné si ho zdarma stáhnout z oficiálních stránek <http://www.joomla.org/>. Redakční systém *Joomla!* byl vydán v roce 2005 a nyní se nachází ve verzi 3.2.0. *Joomla!* používají pro webové stránky významné společnosti jako jsou MTV nebo Harvardská univerzita a jako backendové řešení ho používá například Citibank. *Joomla!* lze použít téměř na všechny různé typy webů od rezervačních systémů po systémy pro řízení majetku. Na rozdíl od *WordPress* za ním nestojí tak významná vývojářská komunita (přesto je však velmi početná) a existuje k němu velké množství rozšíření, dokumentace a témat. Jako nevýhodu můžeme zmínit malé množství kvalitních témat (na rozdíl od

ostatních redakčních systémů) a jeho zbytečnou pokročilost pro tvorbu jednoduchých webových stránek. [2] [5]

1.3.3. Drupal

Drupal je další open source redakční systém, který je založený na PHP, a jak už licence napovídá, je k dispozici zdarma na stránce <https://drupal.org/>. *Drupal* patří mezi populární redakční systémy, který používají velké společnosti jako New York Observer, Popular Science, MIT, Sony Music a další. Tento redakční systém je podporován velmi aktivní komunitou, která pořádá i osobní akce. K dispozici je také komunitou tvořená dokumentace, která se často aktualizuje. Pro *Drupal* je možné stáhnout až 6000 doplňků, které rozšiřují jeho schopnosti a funkčnosti. Jako nevýhodu můžeme zmínit složitost vytváření nových a kvalitních témat a tím pádem taky jejich nedostatek. [2]

1.3.4. GetSimple CMS

Tento redakční systém se zaměřuje na jednoduchost instalace, užívání a nastavení. Na rozdíl od předchozích redakčních systémů nepoužívá databázi MySQL, ale pro ukládání článků používá XML dokumenty. Je určen pro malé a nenáročné projekty a je k dispozici zdarma na stránce <http://get-simple.info/>. [7]

1.3.5. Ostatní

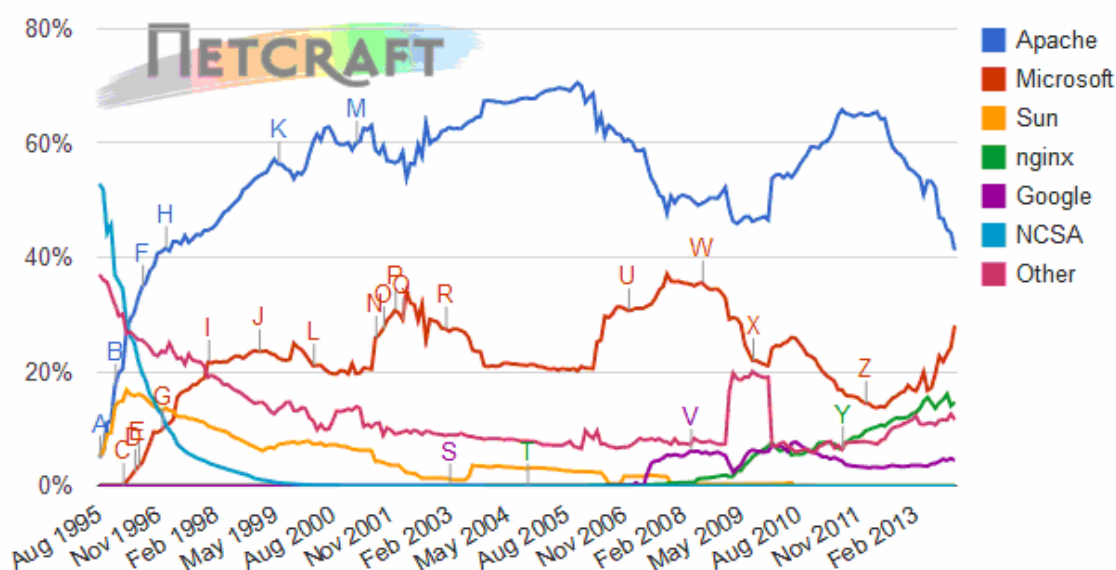
Mezi další redakční systémy patří například *SunLight CMS*, *PHP Vision* nebo *SilverStripe*. Tyto redakční systémy nejsou tak hojně používané, ale mají své zastánce, kteří si je oblíbili a budou je používat. [7]

2. POUŽITÝ SOFTWARE A PROGRAMOVACÍ JAZYKY

2.1. Apache

Apache je hodně výkonný a přitom jednoduchý web server, který je dostupný pro platformy *GNU/Linux*, *BSD*, *Solaris*, *Mac OS X* a *MS Windows*. Je to open source software a je k dispozici zdarma na stránkách <http://apache.org/>. V počátcích tento webový server vznikl jen jako patche pro původní *NCSA server* a tím tak vznikl jeho název a-patche. První oficiální *Apache* ve verzi 0.6.2 byl vydán v dubnu 1995 a od té doby je stále vylepšován až do současné verze 2.4.7., která byla uvedena na trh 25. 11. 2013.

Apache je v dnešní době stále nejpoblárnější webový server, jak je vidět i na grafu obr. 2-1 .



Obr. 2-1: Zastoupení webových serverů [9]

Apache v standardní konfiguraci přináší jen několik základních vlastností, které jsou nutné pro běh serveru. V případě potřeby rozšířit funkcionalitu je možno ho doplnit pomocí různých modulů. Přidáním modulů je možné do *Apache* zakomponovat podporu různých programovacích jazyků jako je například *Perl*, *Python*, *Tcl*, *C*, *C++* a také *PHP*. *Apache* poskytuje mnoho MultiProcessing modulů (MPM), které mu dovolují se přizpůsobit systému, kde běží. Výkon *Apache* pak

hodně závisí na nastavení těchto modulů. Za nejvýkonnější je považována vícevláknová verze kombinující více procesorů a více vláken. [8],[10]

Pro testování mého projektu používám na localhost *Apache* ve verzi 2.4.4.0.

2.2. HTML

HTML je zkratkou pro „*HyperText Markup Language*“ a jedná se o značkovací jazyk pro hypertext. Je to jazyk pro vytváření webových stránek, který umožňuje publikaci dokumentů na internetu.

Původní definici jazyka HTML vytvořil roku 1991 Tim Berners-Lee , který byl zaměstnán jako výzkumník v CERN. Ve stejném roce byl navržen také protokol HTTP pro přenos hypertextu, který umožňoval přenos HTML po síti. HTML jazyk se časem postupně vyvíjel. Verze 1.2 definovaná v roce 1993 ještě nepodporovala grafické rozhraní. Změna přišla v roce 1995 s verzí 2.0, kdy syntax zcela odpovídala SGML a přidávala k původní specifikaci interaktivní formuláře a podporu grafiky. Verze 3.0 nebyla přijata jako standard kvůli složitosti a v roce 1997 byla vydána verze 3.2. Dále byla vydána roku 1997 verze 4.0 a roku 1999 verze 4.1. Aktuálně se HTML nachází ve verzi 5.0 z roku 2014.

HTML jazyk obsahuje množinu značek, jež se označují v originále *tag*, a jejich vlastností (tzv. atributů). Značky se skládají z úhlových závorek a jsou otevírací <tag> a uzavírací </tag>. Mezi značky se vkládá text, který je na webové stránce zobrazen podle významu určité značky, například: ahoj znamená, že text „ahoj“ bude zobrazen výrazně [16].

Struktura základní HTML stránky je popsána ve výpisu kódu 2-1. Obsahuje hlavní HTML značku <html>, ve které je vložen HTML kód, dále je zde důležitá značka <head>, která obsahuje informaci o názvu stránky, kódování stránky a případně link na externí soubor kaskádovými styly. Po značce <head> následuje značka <body>, ve které je již obsažen HTML kód.

Výpis kódu 2-1: Struktura HTML stránky [16]

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titulek stránky</title>
  </head>
  <body>
    <h1>Nadpis stránky</h1>
    <p>Toto je <a href="http://example.com/">odkaz</a> v odstavci.</p>
    <!-- toto je komentář -->
  </body>
</html>
```

2.3. CSS

Kaskádové styly (v anglickém originále *Cascading Style Sheets* – CSS) jsou používány pro popis způsobu zobrazení stránek psaných v jazyce HTML. Jejich úkolem je oddělit kód stylu zobrazení od kódu HTML a rychle jej upravit nebo zaměnit za jiný styl. Jazyk byl navržen organizací W3C roku 1996 a autorem původního návrhu byl Håkon Wium Lie.

Kaskádové styly se definují pravidly. Každé pravidlo obsahuje selektor a blok deklarací. Selektoru předchází identifikátor, pro jakou část bude následující blok deklarací platit. Blok deklarací obsahuje deklarace, které se skládají z identifikátoru vlastnosti, dvojtečky a hodnoty vlastnosti. Deklarace jsou od sebe odděleny středníky. Příklad selektoru a bloku deklarací je uveden ve výpisu kódu 2-2 [17].

Výpis kódu 2-2: Struktura CSS kódu [17]

```
body {
  background-color: white;
  color: black;
  padding: 10px;
}
```

CSS styl se dá v HTML kódu použít třemi způsoby, které jsou zobrazeny ve výpisu kódu 2-3:

- a) Přímý inline zápis pomocí atributu `style`
- b) Zápis stylu na začátek HTML stránky
- c) Připojení externího .css souboru pomocí elementu *link*

Výpis kódu 2-3: Připojení CSS stylu do HTML stránky

```
a)
<p style="color: red; text-decoration: underline">Tento odstavec
bude červený a podtržený.</p>

b)
<style type="text/css">
#hlavicka{
    width: 200px;
    height: 450px;
}
</style>

c)
<head>
    <link rel='stylesheet' href='styley.css' type='text/css'>
</head>
```

Ve své diplomové práci jsem použil CSS styl ze stránky <http://johnsardine.com/example/simple-little-table/> , který jsem následně zmodifikoval pro svou potřebu.

2.4. PHP

Jedná se o skriptovací jazyk pro tvorbu pokročilejších dynamických webových stránek, kde je potřeba používat složitějších funkcí jako je například počítadlo, kniha návštěv, diskusní fórum nebo anketa.

Vznikl v roce 1994, kdy tato zkratka původně znamenala „*Personal Home Page*“ a tehdy bylo PHP napsané v PERLu. Následně bylo přepsáno do jazyka C, protože PERL hodně zatěžoval procesor. Posléze se spojil s nástrojem *Form Interpreter* FI a vzniklo PHP/FI 2.0. V roce 1998 byla uvedena verze PHP 3.0, která byla k dispozici taky pro *Windows*, a používání PHP se začalo rozšiřovat. Zkratka PHP dostala po velkých úpravách kódu nový význam, a to „*Hypertext preprocessor*“. [13]

Nejnovější verze PHP nese pořadové číslo 5.5.7 a je dostupná ke stažení z oficiálních stránek <http://www.php.net/>.

Skriptovací jazyk PHP je v dnešní době velmi oblíbený, a to z mnoha důvodů:

- je relativně jednoduchý,
- má podobnou syntax s jazykem C,
- má velkou podporu vývojářské komunity,
- komunikuje snadno s většinou běžných databází,
- je multiplatformní.

Má ovšem také své nevýhody, mezi které můžeme zařadit, že:

- se jedná o jazyk interpretovaný,
- je definován pouze jedinou svou implementací,
- se může verze od verze lišit funkcemi a stejné funkce mohou mít jiné chování v různých verzích,
- má horší podporu pro Unicode,
- v základu chybí debugovací nástroj.

PHP se zapisuje do souboru společně s HTML. V případě spouštění webové stránky se PHP zpracovává na straně serveru, přeloží se do HTML a klientovi pošle jen výsledek v podobě HTML. Původní HTML kód přijde klientovi vždy ve stejné podobě, jak je napsán. V prohlížeči klienta už žádný PHP kód nenajdeme, vše je serverem přeloženo do HTML a stránka se z pohledu prohlížeče jeví jako statická.

PHP je všestranný jazyk, ve kterém je možné naprogramovat téměř libovolnou aplikaci. Pro ukládání dat lze v PHP použít buď libovolných textových souborů nebo běžně používaných databází. Od HTML kódu se rozezná tak, že mu předchází značka „<?php” a ukončuje ho značka „?>” (nebo pouze <? a ?>) jak lze vidět na výpisu kódu 2-4: . Dále je potřeba, aby soubor obsahující PHP kód měl příslušnou příponu jako .php, nebo .phtml .

2.5. Smarty

Smarty je šablonová nástavba pro PHP, která odděluje prezentační vrstvu (HTML/CSS) od aplikační logiky PHP. Toto rozdělení v praxi znamená, že PHP kód je v jiném souboru než kód pro HTML/CSS a propojuje se pomocí šablon. [14]

Smarty se dá shrnout do následujících bodů:

- Úplné oddělení prezentační vrstvy od aplikační vrstvy
- Doplněk k PHP, nenahrazuje ho

- Lehce se udržuje
- Jednoduchá syntax
- Flexibilita pro další vývoj

Smarty je open source a je možné ho používat zdarma. Do prezentační vrstvy HTML kódu se integruje pomocí následující syntaxe {\$proměnná}. Příklad uvádím na výpisu kódu 2-4:

Výpis kódu 2-4: Porovnání kódu *Smarty* a PHP [14]

```

PHP
<?php for($x=0; $x<20; $x+=2): ?>
    <php echo $x+1; ?>
<?php endfor; ?>

Smarty
{for $x = 1 to 20 step 2}
    {$x}
{/for}

PHP
<a href="<?php echo $bar['zig']; ?>"><?php echo $bar['zag']; ?></a>
<a href="<?php echo $bar['zig2']; ?>"><?php echo $bar['zag2'];
?></a>
<a href="<?php echo $bar['zig3']; ?>"><?php echo $bar['zag3'];
?></a>

Smarty
<a href="{ $bar.zig}">{$bar.zag}</a>
<a href="{ $bar.zig2}">{$bar.zag2}</a>
<a href="{ $bar.zig3}">{$bar.zag3}</a>

```

Mezi další výhodu *Smarty* patří takzvaný „sandboxing“, to znamená, že *Smarty* izoluje prezentační šablony od PHP a tím zamezuje neautorizovanému zásahu do řídicí logiky PHP přes prezentační vrstvu. *Smarty* také dále obsahuje bezpečnostní prvky, které mohou dále omezovat práva na používání šablon.

Smarty sestavuje kopie šablony jako PHP skript a dále používá jen tuto transformovanou kopii. Tímto způsobem jsou zkombinované výhody šablon a rychlosti PHP. Šablona se zkompiluje jen jednou po prvním použití a pak jsou dále používány jen tyto zkompilované šablony. [14]

2.6. MySQL

MySQL je relační databáze typu DBMS (*database management system*). Databáze MySQL je dostupná jako open source, který je možné používat zdarma, a proto je databáze MySQL hodně oblíbená, tedy i rozšířená. Je možnost ji ale také používat pod komerční placenou licenci, která v sobě zahrnuje i oficiální podporu. Obě verze je možné stáhnout ze stránek <http://www.mysql.com>.

MySQL je malá, rychlá a jednoduchá databáze, kterou obsahují jiné databázové systémy, například Oracle [11]. MySQL je multiplatformní databáze, kde komunikace probíhá pomocí jazyka SQL.

První verze MySQL byla vydána v roce 1995 a v roce 2008 byla odkoupena společností SUN, kterou od roku 2009 vlastní firma Oracle.

Architektura MySQL se skládá ze tří vrstev. Horní vrstva obsahuje služby obsluhující většinu potřebných nástrojů klient/server. Ve druhé vrstvě se nachází hlavní funkcionality databáze a kódy pro parsing, analýzu a optimalizaci zabudovaných funkcí. Třetí vrstva obsahuje úložné enginy, které mají na starost ukládání a nahrávání všech dat v databázi. Bezpečnost je zajištěna autentizací podle jména, hesla a pro každý požadavek je ověřováno, jestli má oprávnění pro příslušnou akci. MySQL se dá ovládat jen pomocí příkazového řádku, ale existuje mnoho grafických nadstavb. Při tvorbě mé aplikace jsem využíval grafickou nadstavbu *Heidisql*.

MySQL je napsáno v programovacím jazyce C a C++ a existuje pro ni mnoho knihoven, které umožňují komunikaci dalších různých programovacích jazyků s MySQL databází.

MySQL má také i své limitace:

- nejsou podporovány foreign keys
- umožňuje jen jeden trigger na akci
- nejsou možné triggers u views

Databáze MySQL může běžet i na cloudu, jak to provozuje Amazon, kde uživatelé platí podle toho, jak ji využívají, a nemusí se starat o hosting a údržbu. [12]

2.7. Zebra_database

Zebra_database je pokročilá, kompaktní nástavba nad PHP a MySQL. Jedná se o rozšíření, které poskytuje metody pro interakci z MySQL databází. Je bezpečnější a výkonnější než základní metody PHP. Při transakcích si ukládá výsledky do mezipaměti na disku nebo použitím memcache. *Zebra_database* je velmi dobře komentována a informuje o varování, chybách a oznámení varovnými hlášeními, pokud je v PHP nastaveno hlášení chyb na E_ALL.

Zebra_database je možno stáhnout z webové adresy stefangabos.ro/php-libraries/zebra-database/. Základní použití příkazů pro *Zebra_database* je zobrazeno na výpisech kódů 2-6, 2-7, 2-8 a 2-9.

Výpis kódu 2-5: Použití *Zebra_database*

```
<?php

require 'path/to/Zebra_Database.php';

$db = new Zebra_Database();

$db->debug = true; // turn debugging on

$db->connect('host', 'username', 'password', 'database');

$db->show_debug_console();

?>
```

Výpis kódu 2-6: *Zebra_database* SELECT požadavek

```
<?php
$db->select(
    'column1, column2',
    'table',
    'criteria = ?',
    array($criteria)
);

$records = $db->fetch_assoc_all();
while ($row = $db->fetch_assoc()) { }

?>
```

Výpis kódu 2-7: *Zebra_database* INSERT požadavek

```
<?php
$db->insert(
    'table',
    array(
        'column1' => $value1,
        'column2' => $value2,
    )
);
?>
```

Výpis kódu 2-8: *Zebra_database* UPDATE požadavek

```
<?php
$db->update(
    'table',
    array(
        'column1' => $value1,
        'column2' => $value2,
    ),
    'criteria = ?',
    array($criteria)
);
?>
```

Výpis kódu 2-9: *Zebra_database* QUERY požadavek

```
<?php
$db->query('
//zde je možno zapsat libovolnou SQL query
');
?>
```

3. APLIKACE PRO SPRÁVU PŘEKLADŮ

3.1. Účel aplikace

Účelem aplikace je zrychlení managementu pro rozdělování článků určených k překladu. Do aplikace se budou vkládat články, které bude potřeba přeložit. Aplikace umožní u článků zvolit jazyk překladu, přidělovat články překladatelům a korektorům a nastavit datum, do kterého se musí přidělený článek přeložit a zkontrolovat. Po překladu nabídne export přeloženého článku pro použití v redakčním systému *WordPress*. List jazyků aplikace je libovolně rozšiřitelný podle aktuálních požadavků.

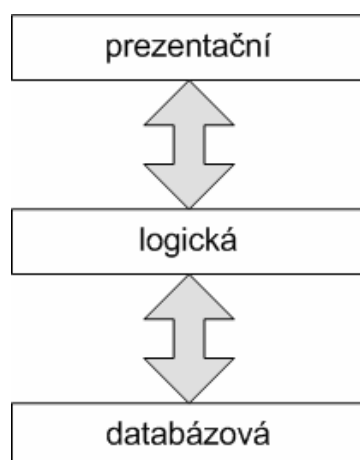
3.2. Popis aplikace

Aplikace je členěna podle typů uživatelských účtů. V aplikaci se nachází pět rozdílných uživatelských účtů, které jsou členěny podle uživatelských rolí a mají různé pravomoci. Mezi uživatelskými účty je možné se libovolně přepínat, ale to jen v případě, že je účet schválen administrátorem. Aplikace pro každý účet zobrazí uživateli jiné uživatelské prostředí s různou funkcionalitou. Nově přihlášenému uživateli aplikace nabídne jen velmi omezenou funkcionalitu jako je zažádání o další účet a změnu hesla. Uživatelský účet „Správce překladu“ umožní články importovat z redakčního systému *WordPress* a zobrazí informace o vložených článcích. Umožní nastavit jazyk překladu, překladatele, korektora a datum, do kdy se má článek přeložit. „Správce překladu“ bude moci přidělit článek na překlad nebo korekturu jen těm uživatelům, kteří mají nastavené, že požadovaný cizí jazyk ovládají. Účet „Překladatel“ a „Korektor“ zobrazí články, které má daný uživatel přeložit nebo zkontrolovat a umožní mu nastavit ovládané cizí jazyky. Účet „redaktor“ zpřístupní články v aplikaci psát a účet administrátora je zodpovědný za schvalování nových účtů. Podrobnější popis uživatelských účtů je popsán v kapitole 3.5.

3.3. Návrh aplikace

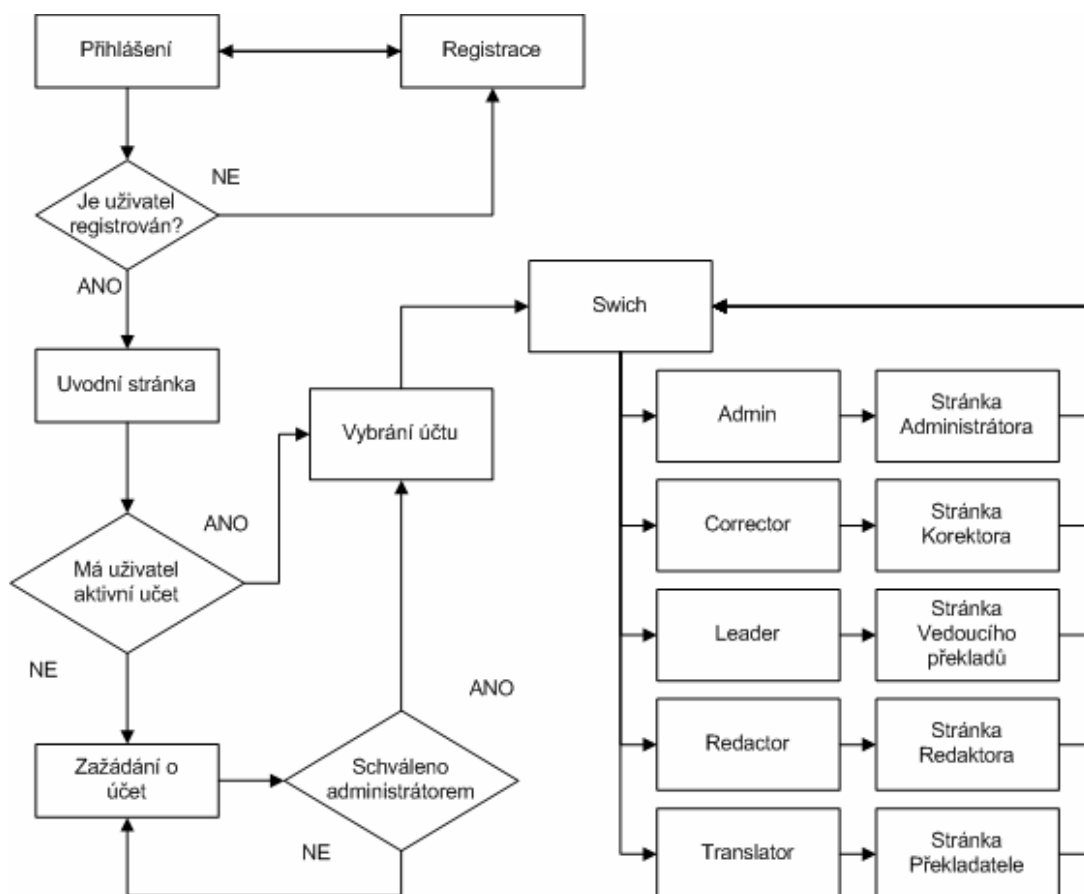
Aplikace je rozdělena na tři vrstvy. Tyto vrstvy spolu komunikují podle schématu, které je zobrazeno na obr. 3-1 . Určitá vrstva je schopna komunikovat jen s vrstvou sousedící. Mezi prezentační a logickou vrstvou se používá doplněk *Smarty*

a mezi logickou a databázovou se používá doplněk *Zebra_database*. Oba tyto doplňky je možné zdarma využívat.



Obr. 3-1: Propojení vrstev

Základní fungování aplikace je zobrazeno na obr. 3-2.



Obr. 3-2: Základní popis aplikace

Po přihlášení do aplikace je uživateli nabídnuto vybrání účtu. Pokud jej nemá přidělený, musí o něj zažádat u administrátora. Po schválení se uživatel může mezi účty přepínat a každý z nich mu zobrazí jinou stránku s odlišnými funkcemi, které jsou popsány v dalších odstavcích. Pokud nemá uživatel účet schválený, nebude mu nabídnuto se do neschváleného účtu přepnout. I kdyby se o to pokusil modifikováním adresy v příkazovém řádku, jeho požadavek bude odmítnut.

3.3.1. Prezentační vrstva

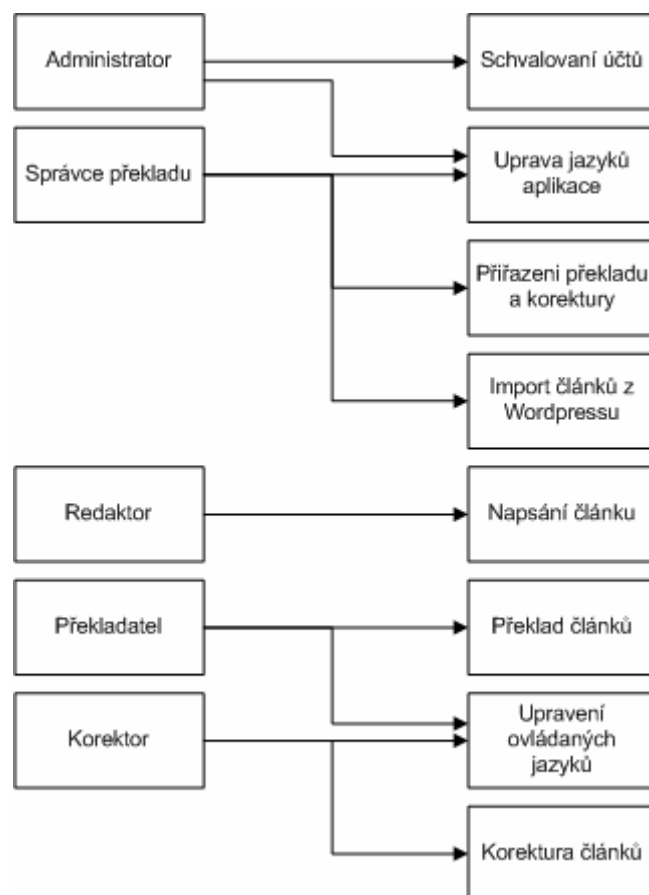
Prezentační vrstva se stará o komunikaci s uživateli. Data zadaná v prezentační vrstvě jsou posílána na vrstvu logickou ke zpracování a následně logická vrstva odpoví zpátky vrstvě prezentační, která je zobrazí uživateli. Informace bude zobrazena uživateli na základě přidělených oprávnění (tzv. účtů). Každý účet, o který uživatel zažádá, musí být prvně schválen administrátorem, čímž uživateli dá určitá oprávnění. Dle přidělených oprávnění pak putují příkazy do logické vrstvy, která je zpracuje. Prezentační vrstva používá jazyk HTML/CSS.

3.3.2. Logická vrstva

V logické vrstvě se bude odehrávat veškeré řízení aplikace. Zde se bude zprostředkovávat komunikace mezi prezentační a databázovou vrstvou. Tato vrstva má za úkol:

- Autentizaci uživatele
- Autorizaci uživatele
- Přidělování oprávnění pro uživatele
- Vytvoření článku, načtení z databáze a uložení do databáze
- Import článku a export článku z redakčního systému *WordPress*
- Přidání a odebrání jazyků používaných aplikací
- Nastavení cílového jazyka pro překlad článku
- Zobrazení překladů pro překladatele a korektory
- Zobrazení postupu překladu a korektury

Kompletní přehled uživatelských účtů a jim přidělených práv je vyobrazeno na obr. 3-3.



Obr. 3-3: Popis uživatelských práv

Logická vrstva komunikuje s prezentační vrstvou pomocí šablon *Smarty* a s databázovou vrstvou přes doplněk *Zebra_database*. Použití těchto doplňků přináší vyšší bezpečnost před napadnutím systému. Popis složení kódu pro logickou vrstvu je podrobněji probrán v kapitole 3.6.

3.3.3. Databázová vrstva

Databázová vrstva má na starost veškeré ukládání a načítání dat pro logickou vrstvu. Logická vrstva posílá databázové vrstvě SQL dotazy, na které bude odpovídat a vracet výsledky do logické vrstvy. Využívá se zde databáze MySQL. Komunikace zde probíhá pomocí doplňku *Zebra_database*, který zrychluje komunikaci s databází, stará se o navazování a ukončování spojení a také o formátovou konverzi dat pro speciální charaktery, které se mohou v textu článků nacházet. Podrobná struktura databáze je popsána v odstavci 3.7.

3.4. Přístup do aplikace

Po zadání adresy aplikace <http://31.31.76.16/stefanik/> do prohlížeče se uživateli nabídne přihlašovací okno, které je zobrazeno na obrázku obr. 3-4, kde uživatel vyplní svůj přihlašovací login a heslo.

Pokud ještě do aplikace není registrován, může se zaregistrovat kliknutím na „registrace“, který ho přesměruje na registrační okno zobrazené na obrázku obr. 3-6. Zde vyplní své jméno, příjmení, email, login a heslo. Dále si vybere účet a cizí jazyky nabídnuté aplikací, které ovládá.

Po přihlášení si může uživatel zvolit účet, se kterým chce pracovat. Nabídka pro vybrání účtu je zobrazena na obr. 3-5.

Správce překladu má pravomoc do aplikace vložit článek a nastavit jazyk, do kterého se má článek přeložit. Článek pak dále může přiřadit překladatelům a korektorům, kteří mají nastaveno, že ovládají jak původní jazyk článku, tak cílový jazyk překladu. Překladatel si může článek prohlédnout a začít s překladem. Ve chvíli přiřazení se uloží datum, kdy překlad začal. Po ukončení překladu se uloží datum konce překladu a článek se začne zobrazovat korektorovi na korekturu. Pro správce překladu nebude možné přiřadit stejný článek pro překlad a korekturu jedné osobě. Po úspěšné korektuře bude možno článek vyexportovat do formátu pro redakční systém *WordPress*.

Všichni uživatelé mají možnost si změnit heslo. Po kliknutí na odkaz „Změnit heslo“ se uživateli zobrazí okno, které je zobrazeno na obrázku obr. 3-7, kde provedou změnu hesla.

3.5. Uživatelské účty

Uživatelské účty jsou rozděleny do následujících kategorií a je možné přiřadit jednomu uživateli více účtů. Každý uživatelský účet má podle typu určitá oprávnění pro modifikace v aplikaci. Uživatel si vybírá přidělené účty na hlavní liště aplikace, která je zobrazena na obr. 3-5. Pokud uživatel nemá přidělený uživatelský účet, může o něj zažádat kliknutím na „zažádat o účet“.

3.5.1. Administrátor systému

Administrátor má administrativní práva k systému a má možnost přidávat a odebírat práva ostatním uživatelům viz obr. 3-8. Aby mohl administrátor přiřadit práva, musí si o něj prvně uživatel zažádat. Dále také může přidat či odebrat další jazyky v aplikaci.

Aktivace/deaktivace účtu uživatele se provede kliknutím na příslušný odkaz v určitém sloupci. Pokud je pole prázdné, znamená to, že uživatel o příslušný účet zatím nezažádal. Po zažádání uživatele o účet se zde zobrazí položka „aktivovat“. Následně po potvrzení administrátorem se položka přepíše na „deaktivovat“ pro možnost pozdější deaktivace účtu a tím odebrání práv uživateli. V posledním sloupci se nachází odkaz na smazání celého uživatele, kterého pak ze systému úplně odstraní, tím se zruší celý uživatelův účet a uživatel se pak nebude moci do systému přihlásit.

Přidání jazyka se provede zapsáním zkratky to textového pole a potvrdí tlačítkem „Přidat jazyk“. Odebrání jazyka pak kliknutím na odkaz odebrat na příslušném řádku. Odebrání jazyků v aplikaci je ošetřeno tak, že není možné odebrat poslední dva jazyky. Pokud by byl v aplikaci nastaven jen jeden jazyk, aplikace pro překlad by ztratila smysl.

3.5.2. Správce překladu

Správce překladu má na starost určit datum, do kdy se má článek přeložit, jazyk překladu a uživatele, který ho bude překládat a kontrolovat. Nastavení článku pro překlad se provede vybráním jazyka ve sloupci „Překládá se do“ a v příslušném řádku pro požadovaný článek. Po nastavení cílového jazyka se dále vybere uživatel, který bude článek překládat a korektor pro následnou kontrolu. Při vybrání překladatele se uloží datum, kdy se překladatel určil. Po dokončení překladu se vyplní datem pole „Korektura zadána“ - to je datum, od kdy může začít korektor opravovat překlad.

List článků aplikace je zobrazen na obr. 3-10. V tomto listu jsou zobrazeny nejdůležitější informace o článcích, jako jsou:

- Název článku
- Jazyk článku
- Cílový jazyk překladu

- Jméno přiděleného překladatele a korektora
- Procentuální postup překladu nebo korektury
- Datum začátku překladu a korektury
- Datum, do kdy se má článek přeložit

Mezi další práva správce překladu patří import článku z redakčního systému *WordPress*. Importované články se zobrazí po kliknutí na odkaz „Import WordPress“ viz obr. 3-11. Import dalších článků se provede vybráním *WordPress* článku tlačítkem „Browse“ a následným potvrzením tlačítkem „Načíst WordPress článek do listu“ se články načtou.

Zde si správce překladu vybere článek, který chce importovat a import do aplikace provede kliknutím na nabízený jazyk, který je přidán do aplikace. Je zde možnost si článek před importem do aplikace zobrazit nebo ho smazat z listu importovaných článků. Z důvodu, že *WordPress* neumožňuje export jednoho přesně určeného článku, ale exportuje více článků zároveň do jednoho výstupního .xml souboru, existuje také možnost smazat všechny importované články z *WordPress* po importu požadovaného článku do aplikace. U importovaných článků se zobrazuje informace o názvu článku, autor a jazyk. Z důvodu, že jazyk článku ve *WordPress* nemusí být kompatibilní s jazykem článku aplikace, je potřeba, aby hlavní překladatel vybral stejný jazyk nabídnutý aplikací. Pokud jazyk článku importovaného z *WordPress* v aplikaci chybí, je potřeba, aby ho správce překladu do aplikace přidal.

3.5.3. Překladatel

Každý překladatel si při vytváření účtu, případně později v aplikaci, nastaví, které jazyky ovládá. Podle nastavených jazyků se budou správci překladu nabízet uživatelé, kterým může článek na překlad přiřadit. Seznam článků přiřazených na přeložení si překladatel zobrazí kliknutím na „Přeložit článek“ viz obr. 3-14.

Překladatel si zde může článek nejprve prohlédnout kliknutím na odkaz „zobrazit“ nebo začít s překladem po kliknutí na odkaz ve sloupci „Začít s překladem“. Po prokliknutí tohoto odkazu se překladateli zobrazí okno překladu. V záhlaví se zobrazí název článku a vedle textové pole, kam překladatel zapíše překlad a potvrdí tlačítkem. Dokud překladatel neoznačí článek za hotový, může libovolně měnit přeložené věty. Tlačítko pro označení hotového překladu se zobrazí v záhlaví až po přeložení všech částí článku. Článek se pro překlad rozdělí podle HTML značek

a každou část překládá překladatel zvlášť. Tyto části jsou zobrazeny vedle sebe v textových polích. V levém textovém poli je zobrazen původní text a do pravého pole ho překladatel překládá. Tlačítko pro uložení všech řádků překladu se nachází v záhlaví. V horní liště se vypisuje název článku, původní a cílový jazyk a hotovost překladu v procentech. Překladatel může překlad kdykoli přerušit a případně se i odhlásit. Po následném přihlášení a zobrazení příslušného překladu se vrátí na stejné místo, kde skončil, a může v překladu pokračovat.

Aby se překladateli článek dal přidělit, musí ovládat jak původní jazyk článku, tak také jazyk překladu. Nastavení jazyků, které ovládá, provede kliknutím na odkaz „Upravit ovládané jazyky“, kde se překladateli nabídne list jazyků, jak je vidět na obr. 3-16.

Podle toho, jaké nabízené jazyky překladatel umí, klikne na odkaz přidat nebo odebrat vedle zkratky jazyka. Zde se zobrazují jen jazyky, které jsou do aplikace vloženy.

3.5.4. Korektor

Korektor má za úkol překlady článků kontrolovat po překladatelích. Okno korektury vypadá podobně jak okno překladu, je stejně rozvržené, jen s tím rozdílem, že neopravené části článku jsou zobrazeny červeně a po potvrzení korektury se podbarví zeleně. Případnou chybu v překladu může korektor okamžitě opravit a po potvrzení se oprava uloží do databáze. Stejně jak překladatel, tak i korektor může korekturu přerušit a později se vrátit na stejné místo, kde přestal. Okno korektury je zobrazeno na obr. 3-18.

Na obrázku obr. 3-17 je zobrazen list článků přidělených na korekturu. Zde může korektor vidět, kdy byla korektura zadána, do kdy se má článek zkontrolovat a kolik dní zbývá do uzávěrky.

3.5.5. Redaktor

Redaktor může články v systému psát. Článek začne vytvářet po kliknutí na odkaz „Vytvořit článek“. Po prokliknutí odkazu se zobrazí okno, kde zadá název článku a jazyk a do textového pole zapíše text. Okno pro psaní článků je zobrazeno na obr. 3-12.

Pod textovým polem jsou zobrazena dvě tlačítka. První „Uložit pro editaci“ - jak už název napovídá - uloží článek do databáze a označí ho jako nekompletní a redaktor ho bude moci v budoucnu ještě opravit. Takovýto článek se ještě nebude nabízet pro překlad. Po kliknutí na tlačítko „Uložit jako finální článek“ se článek označí jako hotový a začne se nabízet pro překlad. Takto uložený článek již nebude možné dále upravovat. Pokud se již nezačalo s překladem, může ho redaktor ještě smazat. List vytvořených článků si může redaktor zobrazit kliknutím na tlačítko „články“, jak je vidět na obr. 3-13.

3.6. Rozbor kódu

3.6.1. Přístupový bod aplikace

Přístupový bod PHP aplikace je soubor index.php, který spouští webový prohlížeč. V případě této aplikace soubor index.php obsahuje jen načtení souboru s konfigurací config.php a soubor s kódem aplikace app.php, jak je zobrazeno na výpisu kódu 3-1.

Výpis kódu 3-1: Kód souboru index.php

```
<?php
require_once ("./app.php");
require_once ("./include/config.php");

$app = new App();

$app->run();
?>
```

3.6.2. Konfigurační soubor

Konfigurační soubor config.php se nachází v adresáři include. Obsahuje cesty k použitým knihovnám a souborům s kódy, které aplikace používá. Obsah souboru config.php je zobrazen na výpisu kódu 3-2.

Výpis kódu 3-2: Kód souboru config.php

```
<?php
require './libs/smarty/Smarty.class.php';
require './libs/zebra/Zebra_Database.php';
require './include/login.php';
```

```
require './include/user.php';
require './include/article.php';
require './include/redactor.php';
require './include/translator.php';
require './include/leader.php';
require './include/admin.php';
require './include/corrector.php';
?>
```

3.6.3. Aplikační soubor

Soubor obsahuje základní řídicí logiku aplikace. Hlavní funkce *run* si pomocí příkazu `$_GET['action']` načte požadovanou akci a podle ní se rozhoduje. Na začátku si aplikace zkontroluje, jestli je uživatel přihlášený. Jestli uživatel přihlášený není, nabídne přihlašovací okno, a pokud je, postupuje dále v programu k požadované akci. V případě, že by uživatel ručně měnil informace obsažené v adrese, aplikace si zkontroluje platnost, a jestli je příkaz nesprávný nebo odkazuje na účet, ke kterému nemá uživatel práva, přesměruje se na domovskou stránku.

3.6.4. Soubory s uživatelskými kódy

Pro zpřehlednění a zjednodušení úpravy kódů aplikace je PHP kód členěný do souborů podle toho, jaký kód obsahuje a pro co se používá. Kódy pro jednotlivé uživatelské účty se nacházejí v souborech se jménem podle jména účtu:

- admin.php
- leader.php
- redactor.php
- translator.php
- corrector.php

Další soubory, které se v aplikaci používají, jsou společné pro všechny typy účtů:

- login.php – obsahuje php kód pro přihlášení do aplikace
- user.php – obsahuje php kód pro operace související s uživatelem

Všechny výše zmíněné soubory se nacházejí v adresáři „include“ a jsou logicky rozděleny na dvě části z důvodu přehlednosti. V první horní části se nachází kódová logika pro řízení aplikace. Jsou zde zapsány funkce pro všechny uživatelské akce a každá z nich je pojmenována s předponou „do_“ a následuje název účtu,

například „do_leader_import_article“. Ve druhé spodní části se nacházejí funkce, ve kterých jsou obsaženy query pro práci s databází. Jména těchto funkcí uvádí předpona podle vlastností funkce:

- get – vrátí data z databáze
- set – změní data v databázi
- insert – vloží data do databáze

Pokud je funkce složitější a vykoná více částečných kroků, je pojmenována podle toho, co přesně provádí.

3.6.5. Soubory s šablonami pro Smarty

Šablony pro doplněk *Smarty* se nachází v adresáři „templat“ a „template_c“. Ve složce „template“ jsou uloženy šablony vytvořené uživatelem a do složky „template_c“ ukládá doplněk *Smarty*. Uživatelská šablona obsahuje HTML a *Smarty* kód. *Smarty* tento kód zpracuje a vytvoří z něj plnohodnotný HTML/PHP kód, který ukládá do složky „template_c“. Kód je následně odeslán na zobrazení do prohlížeče. Tento systém šablon zpřehlední HTML kód a oddělí PHP od HTML. Ve *Smarty* šablonách je čistě jen HTML a potřebné proměnné jsou do něj propagovány skrz logiku *Smarty*. Každý uživatelský účet má jednu základní šablonu pojmenovanou podle jména účtu a pak několik dalších podle toho, jaké funkce plní. Tyto šablony jsou pojmenované podle jména, funkce a mají příponu .tpl. (např. „corrector_display_article.tpl“).

3.6.6. Soubory knihoven

Knihovny použité v aplikaci pro správu překladů se nacházejí v adresáři libs. Jedná se o knihovny:

- *Smarty*
- *Zebra_database*

Použití a chování obou knihoven je popsáno v předchozích odstavcích této diplomové práce.

3.7. Struktura databáze

3.7.1. Tabulka article

V tabulce „article“ jsou uloženy všechny články aplikace. Každý článek v tabulce je identifikován unikátním klíčem „article_id“, který se používá jako klíč v SQL dotazech při vybírání článků a využívá se zde funkce „auto_increment“, která každému nově vloženému článku přiřadí ID o jedno větší, tudíž má každý vložený článek různé id. Mezi další informace obsažené v article je ID autora, jazyk, název, vlastní text a zda je článek označen za hotový. Pro překlad tabulka obsahuje další informace: cílový jazyk překladu, jméno překladatele a korektora, postup překladu v procentech, přeložený název a text článku. SQL query pro vytvoření tabulky je uvedeno ve výpisu kódu 3-3.

Výpis kódu 3-3: Kód pro vytvoření tabulky article

```
CREATE TABLE `article` (  
  `article_id` INT(11) NOT NULL AUTO_INCREMENT,  
  `origin_id` INT(11) NULL DEFAULT NULL,  
  `source` VARCHAR(50) NULL DEFAULT NULL,  
  `redactor_id` INT(11) NOT NULL,  
  `article_origin_language` VARCHAR(50) NULL DEFAULT NULL,  
  `is_final` VARCHAR(50) NULL DEFAULT NULL,  
  `article_name` VARCHAR(50) NOT NULL,  
  `article_content` VARCHAR(10000) NOT NULL,  
  `translate_to` VARCHAR(50) NULL DEFAULT NULL,  
  `translate_progress` FLOAT NULL DEFAULT NULL,  
  `translate_final` VARCHAR(50) NULL DEFAULT NULL,  
  `article_name_translated` VARCHAR(100) NULL DEFAULT NULL,  
  `article_content_translated` VARCHAR(10000) NULL DEFAULT NULL,  
  `translator_id` INT(11) NULL DEFAULT NULL,  
  `translator_name` VARCHAR(50) NULL DEFAULT NULL,  
  `corrector_id` INT(11) NULL DEFAULT NULL,  
  `corrector_name` VARCHAR(50) NULL DEFAULT NULL,  
  `corrector_progress` FLOAT NULL DEFAULT NULL,  
  `corrected` VARCHAR(1) NULL DEFAULT NULL,  
  `tran_date` VARCHAR(50) NULL DEFAULT NULL,  
  `corr_date` VARCHAR(20) NULL DEFAULT NULL,  
  `end_date` VARCHAR(20) NULL DEFAULT NULL,  
  INDEX `ID` (`article_id`)  
)  
COLLATE='latin1_swedish_ci'
```

```
ENGINE=InnoDB  
AUTO_INCREMENT=1;
```

3.7.2. Tabulka article_translate

Do této tabulky se ukládá finální článek, který je označen k překladu. Text článku se rozdělí podle HTML značek a každá HTML značka nebo její obsah se uloží na samostatný řádek. Identifikaci článku napomáhá sloupec `article_id`, kde se ukládá ID článku vygenerované v tabulce „`article`“ a „`line_id`“, kde má každý řádek jednoho článku různý identifikátor. Identifikátor 0 označuje název článku a text článku nabývá hodnot od 1 výše. Originální text článku je uložen ve sloupci „`text`“ a přeložený text se ukládá do sloupce „`translated`“. Další sloupec „`is_tag`“ označuje, zda se na řádku nachází HTML značka, která se nebude překládat, nebo text určený pro přeložení. Sloupec „`translated_to`“ označuje cílový jazyk překladu a „`corrected`“ značí, jestli byl už daný řádek zkontrolován. Pomocí sloupců „`is_tag`“, „`translated`“ a „`corrected`“ se vypočítává procento hotovosti překladu nebo korektury, které se zobrazují hlavnímu překladateli v aplikaci. SQL query pro vytvoření tabulky je uvedeno ve výpisu kódu 3-4.

Výpis kódu 3-4: Kód pro vytvoření tabulky `article_translate`

```
CREATE TABLE `article_translate` (  
  `article_id` INT(11) NULL DEFAULT NULL,  
  `line_id` INT(11) NULL DEFAULT NULL,  
  `text` VARCHAR(10000) NULL DEFAULT NULL,  
  `is_tag` VARCHAR(1) NULL DEFAULT NULL,  
  `translate_to` VARCHAR(10) NULL DEFAULT NULL,  
  `translated` VARCHAR(10000) NULL DEFAULT NULL,  
  `corrected` VARCHAR(1) NULL DEFAULT NULL  
)  
COLLATE='latin1_swedish_ci'  
ENGINE=InnoDB;
```

3.7.3. Tabulka article_wp

Tabulka „`article_wp`“ obsahuje články importované z redakčního systému *WordPress*. Každému vloženému článku z *WordPress* je přiřazeno unikátní ID funkcí „`auto_increment`“, které je uloženo ve sloupci „`article_wp_id`“. První ID začíná na čísle 9000000. Identifikátor *WordPress* článku v této tabulce je o několik řádů vyšší z toho důvodu, protože se po importu přesune do tabulky „`article`“, kde nemůžou mít dva

články stejné id. Dále se zde nachází sloupec „in_use“, který označuje, jestli už se článek v aplikaci používá. Tím se znemožní jeho smazání a případný pokus vložit do aplikace stejný článek vícekrát. Tato tabulka se používá jako mezikrok pro import *WordPress* článku. *WordPress* neumožňuje exportovat přesně vybraný článek, ale jen články určité kategorie nebo časového rozmezí, a proto se první importuje list článku do této tabulky a až po vybrání určitého článku se článek přesune do tabulky „article“. SQL query pro vytvoření tabulky je uvedeno ve výpisu kódu 3-5.

Výpis kódu 3-5: Kód pro vytvoření tabulky article_wp

```
CREATE TABLE `article_wp` (  
  `article_wp_id` INT(11) NOT NULL AUTO_INCREMENT,  
  `in_use` VARCHAR(1) NULL DEFAULT 'N',  
  `language1` VARCHAR(50) NULL DEFAULT NULL,  
  `wxr_version` VARCHAR(50) NULL DEFAULT NULL,  
  `title` VARCHAR(50) NULL DEFAULT NULL,  
  `creator` VARCHAR(50) NULL DEFAULT NULL,  
  `encoded` MEDIUMTEXT NULL,  
  `status1` VARCHAR(1000) NULL DEFAULT NULL,  
  `post_type` VARCHAR(1000) NULL DEFAULT NULL,  
  INDEX `ID` (`article_wp_id`)  
)  
COLLATE='latin1_swedish_ci'  
ENGINE=InnoDB  
AUTO_INCREMENT=9000000;
```

3.7.4. Tabulka member

V tabulce „member“ je uložen list všech uživatelů, kteří se do aplikace zaregistrovali. Obsahuje informace o uživateli jako „id“, „jméno“, „příjmení“, „email“, „login“ a „heslo“. Heslo v tabulce je šifrováno kryptovacím algoritmem md5. Tato tabulka dále obsahuje informaci o zažádaných účtech a jejich aktivaci. O aktivaci účtu si může požádat přihlášený uživatel u administrátora. Nový uživatel se bude moci do aplikace přihlásit, ale bez aktivovaného účtu si nemůže zobrazit data. SQL query pro vytvoření tabulky je uvedeno ve výpisu kódu 3-6.

Výpis kódu 3-6: Kód pro vytvoření tabulky member

```
CREATE TABLE `member` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT,  
  `username` VARCHAR(30) NOT NULL,  
  `password` VARCHAR(100) NOT NULL,
```

```

`fname` VARCHAR(30) NOT NULL,
`lname` VARCHAR(30) NOT NULL,
`email` VARCHAR(100) NOT NULL,
`admin` VARCHAR(1) NULL DEFAULT NULL,
`leader` VARCHAR(1) NULL DEFAULT NULL,
`redactor` VARCHAR(1) NULL DEFAULT NULL,
`translator` VARCHAR(1) NULL DEFAULT NULL,
`corrector` VARCHAR(1) NULL DEFAULT NULL,
PRIMARY KEY (`id`)
)
COLLATE='latin1_swedish_ci'
ENGINE=MyISAM
AUTO_INCREMENT=1;

```

3.7.5. Tabulka member_accounts

Tato tabulka obsahuje informace o účtech uživatelů. Je provázána s tabulkou „member“, ale data o účtech jsou zde uloženy v jiném formátu z důvodu jednodušších SQL dotazů. Na rozdíl od tabulky „member“, kde jsou účty uloženy na jednom řádku pro každého uživatele, zde každý řádek představuje jeden uživatelský účet a informaci, zda je účet aktivován. SQL query pro vytvoření tabulky je uvedeno ve výpisu kódu 3-7.

Výpis kódu 3-7: Kód pro vytvoření tabulky member_account

```

CREATE TABLE `member_accounts` (
  `account_id` INT(11) NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(30) NOT NULL,
  `account` VARCHAR(10) NOT NULL,
  `account_active` VARCHAR(1) NOT NULL,
  PRIMARY KEY (`account_id`)
)
COLLATE='latin1_swedish_ci'
ENGINE=MyISAM
AUTO_INCREMENT=1;

```

3.7.6. Tabulka member_languages

Uchovává informaci o jazycích uživatelů, které mají nastavené jako ovládané. Obsahuje sloupce „user_id“, „username“ a „language“. Každý jazyk určitého uživatele je zde uložen na jednom řádku. SQL query pro vytvoření tabulky je uvedeno ve výpisu kódu 3-8.

Výpis kódu 3-8: Kód pro vytvoření tabulky article_languages

```
CREATE TABLE `member_languages` (  
  `user_id` INT(11) NULL DEFAULT NULL,  
  `username` VARCHAR(50) NULL DEFAULT NULL,  
  `language` VARCHAR(50) NULL DEFAULT NULL  
)  
COLLATE='latin1_swedish_ci'  
ENGINE=InnoDB;
```

3.7.7. Tabulka language

Obsahuje jazyky vložené do aplikace. Aplikace nabízí možnost překladu jen do jazyků, které jsou uloženy v této tabulce. Před uvedením aplikace do chodu na novém hostingu je potřeba do této tabulky vložit minimálně jeden jazyk. Jazyky aplikace se nabízí na základě této tabulky. Kód pro vytvoření a vložení jednoho jazyka je uveden ve výpisu kódu 3-9.

Výpis kódu 3-9: Kód pro vytvoření tabulky language

```
CREATE TABLE `language` (  
  `language_id` VARCHAR(50) NULL DEFAULT NULL,  
  UNIQUE INDEX `language_id` (`language_id`)  
)  
COLLATE='latin1_swedish_ci'  
ENGINE=InnoDB;  
  
INSERT INTO `language` (`language_id`) VALUES ("cz");  
INSERT INTO `language` (`language_id`) VALUES ("en");  
INSERT INTO `language` (`language_id`) VALUES ("de");  
INSERT INTO `language` (`language_id`) VALUES ("fr");
```

3.7.8. Tabulka account_list

Tato pomocná tabulka obsahuje list všech účtů aplikace. Je vytvořena pro zjednodušení SQL dotazů používaných v aplikaci a obsahuje jména účtů. SQL query pro vytvoření tabulky a vložení účtů je uvedeno ve výpisu kódu 3-10.

Výpis kódu 3-10: Kód pro vytvoření tabulky account_list

```
CREATE TABLE `account_list` (  
  `accounts` VARCHAR(50) NULL DEFAULT NULL,  
  `accountcz` VARCHAR(50) NULL DEFAULT NULL,  
)  
COLLATE='latin1_swedish_ci'
```

```
ENGINE=InnoDB;

INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("admin","Administr&acute;tor");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("leader","Spr&acute;vce p&#x0159;kladu");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("corrector","Korektor");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("redactor","Redaktor");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("translator","P&#x0159;kladatel");
```

3.8. Webhosting

Aplikace je nahrána na serveru, kde se používá PHP verze 5.3.10-1 a je dostupná na adrese <http://31.31.76.16/stefanik/>. Pro administraci je použita aplikace phpMyAdmin ve verzi 3.4.10.1.

3.9. Bezpečnost

Zabezpečení aplikace pro správy překladů je složeno s více bezpečnostních prvků, které jsou popsány v následujících odstavcích.

3.9.1. Zabezpečení účtů

Pokud je uživatel přihlášený do aplikace, může se přepnout jen do těch účtů, které mu schválil administrátor. Jestli se uživatel pokusí modifikovat url adresu, aby se dostal na stránku neschváleného účtu, bude jeho požadavek odmítnut. Tento útok je v aplikaci zabezpečen tak, že před každým načtením stránky se zkontroluje, jestli má na ní uživatel právo. Pokud ne, stránka se nezobrazí.

3.9.2. Zabezpečení hesla

Bezpečnost hesla je v aplikaci zajištěna hashovacím algoritmem Message-Digest (zkráceně md5). Algoritmus vytváří hash o délce 128 bitů a minimální změna vstupu do tohoto algoritmu velmi pozmění výstup.

3.9.3. Zabezpečení PHP

Dalším prvkem zabezpečení je použití doplňku *Smarty*. Šablonová nástavba *Smarty* izoluje prezentační šablony od PHP a tím zamezuje neautorizovanému zásahu do řídicí logiky PHP přes prezentační vrstvu.

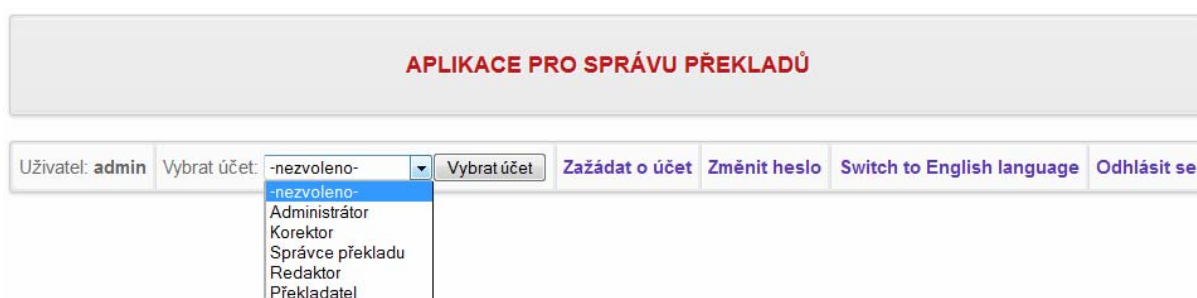
3.10. Uživatelský popis

Aplikace je psána tak, aby byla uživatelsky přívětivá a jednoduchá. Po zapsání adresy aplikace do prohlížeče se zobrazí přihlašovací stránka, která je zobrazená na obr. 3-4.



Obr. 3-4: Login

Pokud ještě není uživatel registrován, je zde odkaz na registraci a změnu jazyka aplikace. V registračním okně (viz obr. 3-6) vyplní jméno, příjmení, email, login, heslo a vybere účet a jazyky, které ovládá. Po registraci se uživatel může přihlásit, ale bez schváleného účtu administrátorem nemůže dál s aplikací pracovat. O další uživatelský účet může zažádat kliknutím na odkaz v hlavní liště, viz obr. 3-5.



Obr. 3-5: Úvodní menu po přihlášení

Po schválení účtu se už do něho uživatel může přepnout a pracovat. Na hlavní liště je dále odkaz na změnu hesla a odhlášení se z aplikace.

V následujících podkapitolách jsou popsány úkony pro každý účet a popis, jak úkon provede.

Registrace

Jméno:

Příjmení:

Email:

Login ID:

Heslo:

Účet:

Jazyk:

☐ cz

☐ de

☐ en

☐ fr

Vyplň následující položky: login, heslo, účet, jméno, příjmení, email, jazyk

[Login](#) [Switch to English](#)

Obr. 3-6: Registrace

Uživatel: admin Vybrat účet: [Zažádat o účet](#) [Změnit heslo](#) [Switch to English language](#) [Odhlásit se](#)

ZMĚNA HESLA

Původní heslo:

Nové heslo:

Nové heslo znova:

Obr. 3-7: Změna hesla

3.10.1. Administrátor

Schválit účet uživatele – v nabídce „List uživatelů“ se klikne na odkaz „aktivovat“ na řádku s požadovaným uživatelem a ve sloupci s příslušným účtem.

Odebrat účet uživatele – v nabídce „List uživatelů“ se klikne na odkaz „deaktivovat“ na řádku s požadovaným uživatelem a ve sloupci s příslušným účtem.

Uživatel: admin

Vybrat účet: Administrátor

Vybrat účet

Zažádat o účet

Změnit heslo

Switch to English language

Odhlásit se

LIST UŽIVATELŮ

UPRAVIT JAZYKY APLIKACE

JMÉNO	PŘÍJMENÍ	LOGIN	EMAIL	ADMINISTRÁTOR	SPRÁVCE PŘEKladU	REDAKTOR	PŘEKladATEL	KOREKTOR	SMAZAT ÚČET
adminjmeno	adminprijmeni	admin	admin@admin.cz	deaktivovat	deaktivovat	deaktivovat	deaktivovat	deaktivovat	smazat
uzivatel1jmeno	uzivatel1prijmeni	uzivatel1	uzivatel1@uzivatel1.cz		deaktivovat	deaktivovat			smazat
uzivatel2jmeno	uzivatel2prijmeni	uzivatel2	uzivatel2@uzivatel2.cz				deaktivovat	deaktivovat	smazat
uzivatel3jmeno	uzivatel3prijmeni	uzivatel3	uzivatel3@uzivatel3.cz				deaktivovat	deaktivovat	smazat
uzivatel4jmeno	uzivatel4prijmeni	uzivatel4	uzivatel4@uzivatel4.cz				deaktivovat	deaktivovat	smazat
uzivatel5jmeno	uzivatel5prijmeni	uzivatel5	uzivatel5@uzivatel5.cz				deaktivovat	deaktivovat	smazat

Obr. 3-8: Administrátora - list uživatelů

Smazat uživatele – v nabídce „List uživatelů“ se klikne na odkaz „smazat“ na řádku s požadovaným uživatelem a ve sloupci „Smazat účet“.

Přidat nový jazyk do aplikace – v nabídce „Upravit jazyky aplikace“ se zapíše zkratka nového jazyka do textového pole ve sloupci „Zkratka jazyka“ a potvrdí se kliknutím na „Přidat Jazyk“.

Odebrat existující jazyk z aplikace – v nabídce „Upravit jazyky aplikace“ se klikne na odkaz „odebrat“ na řádku požadovaného jazyka.

Uživatel: admin	Vybrat účet: Administrátor	Vybrat účet	Zažádat o účet	Změnit heslo	Switch to English language	Odhlásit se
LIST UŽIVATELŮ	ZKRATKA JAZYKA	MOŽNOSTI				
UPRAVIT JAZYKY APLIKACE		Přidat jazyk				
	cz	odebrat				
	de	odebrat				
	en	odebrat				
	fr	odebrat				

Obr. 3-9: Úprava jazyků aplikace

3.10.2. Správce překladu

Nastavit jazyk překladu článku – v nabídce „Články“ se vybere jazyk z dropdown menu ve sloupci „Přeložit do jazyka“ a potvrdí se tlačítkem „Překlad“.

Nastavit překladatele článku – v nabídce „Články“ se vybere překladatel z dropdown menu ve sloupci „Překladatel“ a potvrdí tlačítkem „Překladatele“.

Nastavit korektora článku – v nabídce „Články“ se vybere korektor z dropdown menu ve sloupci „Korektor“ a potvrdí tlačítkem „Korektora“.

Nastavit cílový datum překladu článku – v nabídce „Články“ se vybere datum z dropdown menu ve sloupci „Přeložit do“ a potvrdí tlačítkem „Potvrdit“.

Zrušit překlad – v nabídce „Články“ se klikne na „zrušit překlad“ ve sloupci „Zrušit překlad“ na příslušném řádku článku.

Uživatel: admin

Vybrat účet:

Správce překladu

Vybrat účet

Zažádat o účet

Změnit heslo

Switch to English language

Odhlásit se

ČLÁNKY

IMPORT WORDPRESS

UPRAVIT JAZYKY APLIKACE

NÁZEV ČLÁNKU	PŮVODNÍ JAZYK	PŘEKLÁDÁ SE DO	PŘEKLADATEL	PŘEKLAD	PŘEKLAD ZADAN	KOREKTOR	KOREKTURA	KOREKTURA ZADANA	PŘELOŽIT DO	ZRUŠIT PŘEKLAD	Export do WordPress	SMAZAT
test claneek 1	cz	en	admin	16.67%	2014-05-22	uzivatel2	0%		2014-05-31 Potvrdit	zrusit preklad		smazat
test claneek 3	cz	fr	admin	0%	2014-05-22	uzivatel4	0%		2014-05-29 Potvrdit	zrusit preklad		smazat
test claneek 4	cz	de	uzivatel3	100%	2014-05-22	admin	66.67%	2014-05-22	Potvrdit	zrusit preklad		smazat
test claneek 5	cz	en	uzivatel2	100%	2014-05-22	admin	100%	2014-05-22	2014-06-04 Potvrdit	zrusit preklad		smazat
test claneek 6	cz	vybrat překlad		0%			0%		Potvrdit			smazat
test claneek 7	cz	en	vybrat překladaatele	0%			0%		Potvrdit	zrusit preklad		smazat

Obr. 3-10: Správce překladu – list článků aplikace

Exportovat články pro WordPress – v nabídce „Články“ ve sloupci „Exportovat do WordPress“ se označí články, které chce uživatel exportovat a export provede kliknutím na tlačítko „Exportovat do WordPress“.

Smazat článek z aplikace – v nabídce „Články“ se klikne na odkaz „Smazat“ ve sloupci „Smazat“ na příslušném řádku článku.

Import WordPress článku do aplikace – v nabídce „Import WordPress“ se vybere soubor s články exportovanými z *WordPress* tlačítkem „Browse“ a potvrdí se načtení do listu tlačítkem „Načíst WordPress článek do listu“. V listu importovaných článků se vybere požadovaný článek a import do aplikace se provede

kliknutím na zkratku jazyka ve sloupci „Importuj kliknutím“. Článek si před importem může uživatel zobrazit kliknutím na odkaz „zobrazit“ nebo smazat kliknutím na odkaz „smazat“. Smazání celého listu provede kliknutím na odkaz „Smazat vše“ v záhlaví tabulky.

Uživatel: admin

Vybrat účet:

Správce překladu

Vybrat účet

Zažádat o účet

Změnit heslo

Switch to English language

Odhlásit se

ČLÁNKY

IMPORT WORDPRESS

UPRAVIT JAZYKY APLIKACE

Browse_

No file selected.

Načíst WordPress články do listu

NÁZEV	AUTOR	JAZYK	IMPORTUJ KLIKNUTIM	ZOBRAZIT	SMAZAT VSE
test clanek 1	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 2	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 3	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 4	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 5	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 6	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 7	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat
test clanek 8	bagrov	en-US	Článek je napsán v jazyce: cz de en fr	Zobrazit	Smazat

Obr. 3-11: Správce překladu – list importovaných článků z *WordPress*

Přidat nový jazyk do aplikace – v nabídce „Upravit jazyky aplikace“ se zapíše zkratka nového jazyka do textového pole ve sloupci „Zkratka jazyka“ a potvrdí se kliknutím na „Přidat Jazyk“.

Odebrat existující jazyk z aplikace – v nabídce „Upravit jazyky aplikace“ se klikne na odkaz „odebrat“ na řádku požadovaného jazyka.

3.10.3. Redaktor

Napsat článek – v nabídce „Vytvořit článek“ se zapíše název článku do textového pole v záhlaví, vybere se jazyk s dropdown menu a napíše se text článku. Pro editaci se text uloží tlačítkem „uložit pro editaci“. Uložení finálního článku se provede kliknutím na tlačítko „Uložit jako finální článek“.

Uživatel: **admin**
Vybrat účet: Redaktor
Vybrat účet
Zažádat o účet
Změnit heslo
Switch to English language
Odhlásit se

VYTVOŘIT ČLÁNEK
ČLÁNKY

NÁZEV ČLÁNKU

JAZYK ČLÁNKU

cZ

TEXT ČLÁNKU

test článek admin1

Uložit pro editaci

Uložit jako finální článek

Obr. 3-12: Redaktor – vytvoření článku

Zobrazit napsané články – klikne se na nabídku „Články“.

Uživatel: **admin**
Vybrat účet: Redaktor
Vybrat účet
Zažádat o účet
Změnit heslo
Switch to English language
Odhlásit se

VYTVOŘIT ČLÁNEK	NÁZEV ČLÁNKU	JAZYK ČLÁNKU	JAZYK PŘEKladU	FINÁLNÍ ČLÁNEK	ZAPOČAL PŘEKlad	MOŽNOSTI	
ČLÁNKY	test článek admin1	cZ		Ne	Ne	zobrazit	smazat
	test článek admin2	cZ		Ano	Ne	zobrazit	smazat
	test článek admin3	cZ		Ano	Ne	zobrazit	smazat
	test článek admin4	cZ		Ne	Ne	zobrazit	smazat

Obr. 3-13: Redaktor – list článků

3.10.4. Překladatel

Zobrazit články pro překlad – klikne se na nabídku „Přeložit článek“.

Zobrazit obsah článku – v nabídce „Přeložit článek“ se klikne na odkaz „zobrazit“ na příslušném řádku článku.

Uživatel: **admin**
Vybrat účet: Překladatel
Vybrat účet
Zažádat o účet
Změnit heslo
Switch to English language
Odhlásit se

PŘELOŽIT ČLÁNEK	NÁZEV ČLÁNKU	JAZYK	PŘELOŽIT DO	PŘEKlad ZADÁN	UZÁVĚRKA	DNÍ DO UZÁVĚRKY	POSTUP PŘEKladU	ZOBRAZIT	ZaČIT S PŘEKladEM
UPRAVIT OVLÁDANÉ JAZYKY	test clanek 1	cZ	en	2014-05-22	2014-05-31	9	16.67%	zobraz	Přeložit
	test clanek 3	cZ	fr	2014-05-22	2014-05-29	7	0%	zobraz	Přeložit

Obr. 3-14: Překladatel – list článků pro překlad

Přeložit článek – v nabídce „Přeložit článek“ se klikne na odkaz „Přeložit“ ve sloupci „začít s překladem“ na příslušném řádku článku. Zobrazí se okno s překladem, kde v levé půlce je zobrazen původní text a do pravé půlky zadává překladatel překlad. Překlad uloží tlačítkem „Uložit překlad“. Pokud je celý článek přeložen, zobrazí se v záhlaví tlačítko „Označit překlad jako hotový“, kterým uživatel překlad potvrdí a ukončí.

The screenshot shows a web interface for translating articles. At the top, there is a navigation bar with the following elements: "Uživatel: admin", "Vybrat účet: Překladatel" (with a dropdown arrow), "Vybrat účet", "Zažádat o účet", "Změnit heslo", "Switch to English language", and "Odhlásit se". Below this, a status message reads: "Překládá se článek 'test claneek 1' z jazyka cz do jazyka en, překlad je hotový na 16.67%". The main content area is titled "NÁZEV ČLÁNKU: TEST CLANEK 1" and includes a "Uložit překlad" button. On the left, there are two buttons: "PŘELOŽIT ČLÁNEK" and "UPRAVIT OVLÁDANÉ JAZYKY". The main table displays a list of articles with two columns: the original text in Czech and the translated text in English. The first article is "test claneek 1" / "test article 1". The second article is about USB Thunderbolt, with the Czech text "USB čelí vylepšené konkurenci. Thunderbolt je rychlejší" and the English text "USB article 1". The third article is dated "13. května 2014" / "13. May 2014". The fourth article is about Intel Thunderbolt, with the Czech text "Intel představil třetí generaci rozhraní Thunderbolt, které známe především z" and the English text "Intel article 1". The fifth article is "Další 1 fotografie v galerii" / "Further 1 photograph in the gallery". The sixth article is about LightPeak, with the Czech text "Rozhraní LightPeak, které nyní využívá systém Thunderbolt, Intel představil již v roce 2010." and the English text "LightPeak article 1".

Obr. 3-15: Překladatel – list článků pro překlad

Přidat ovládaný jazyk – v nabídce „Upravit ovládané jazyky“ se klikne na odkaz „Přidat“ na řádku s příslušným jazykem.

Odebrat ovládaný jazyk – v nabídce „Upravit ovládané jazyky“ se klikne na odkaz „Odebrat“ na řádku s příslušným jazykem.

The screenshot shows a web interface for managing languages. At the top, there is a navigation bar with the following elements: "Uživatel: admin", "Vybrat účet: Překladatel" (with a dropdown arrow), "Vybrat účet", "Zažádat o účet", "Změnit heslo", "Switch to English language", and "Odhlásit se". Below this, there are two buttons: "PŘELOŽIT ČLÁNEK" and "UPRAVIT OVLÁDANÉ JAZYKY". The main content area is titled "JAZYKY" and displays a list of languages with buttons to add or remove them. The languages listed are "cz - odebrat", "de - odebrat", "en - odebrat", and "fr - odebrat".

Obr. 3-16: Překladatel – list jazyků

3.10.5. Korektor

Zobrazit články pro korekturu – klikne se na nabídku „Zobrazit články na korekturu“.

Zobrazit obsah článku – v nabídce „Zobrazit články na korekturu“ se klikne na odkaz „zobrazit“ na příslušném řádku článku.

Uživatel: admin	Vybrat účet: Korektor	Vybrat účet	Zažádat o účet	Změnit heslo	Switch to English language	Odhlásit se
-----------------	-----------------------	-------------	----------------	--------------	----------------------------	-------------

ZOBRAZIT ČLÁNKY NA KOREKTURU	NÁZEV ČLÁNKU	JAZYK	OPRAVIT DO	PROGRES	OPRAVA ZADÁNA	UZÁVĚRKA	DNÍ DO UZÁVĚRKY	ZOBRAZIT	ZAČÍT S OPRAVOU
UPRAVIT OVLÁDANÉ JAZYKY	test clanek 4	CZ	de	66.67%	2014-05-22	2014-05-28	6	zobrazit	zkontrolovat
	test clanek 2	CZ	de	0%	2014-05-22			zobrazit	zkontrolovat
	test clanek 3	CZ	de	0%	2014-05-22			zobrazit	zkontrolovat

Obr. 3-17: Korektor – články na korekturu

Přeložit článek – v nabídce „Přeložit článek“ se klikne na odkaz „zkontrolovat“ ve sloupci „začít s opravou“ na příslušném řádku článku. Zobrazí se okno s korekturou, kde v levé půlce je zobrazen původní text a v pravé půlce je zobrazen překlad. Korekturu potvrdí tlačítkem „Potvrdit opravu“ pod každým textovým polem. Neopravená část se v textovém poli podbarví červeně, opravená část zeleně. Pokud je celý článek zkontrolován, zobrazí se v záhlaví tlačítko „Označit hotovou korekturu“, kterým uživatel korekturu potvrdí a ukončí.

Uživatel: admin	Vybrat účet: Korektor	Vybrat účet	Zažádat o účet	Změnit heslo	Switch to English language	Odhlásit se
-----------------	-----------------------	-------------	----------------	--------------	----------------------------	-------------

Opravuje se článek "test clanek 4" z jazyka cz do jazyka de, oprava je hotova na 66.67%

ZOBRAZIT ČLÁNKY NA KOREKTURU	NÁZEV ČLÁNKU: TEST CLANEK 4
UPRAVIT OVLÁDANÉ JAZYKY	
test clanek 4	test article 5
Potvrdit opravu	
USB čelí vylepšené konkurenci. Thunderbolt je rychlejší	USB facing better competition. Thunderbold is faster.
Potvrdit opravu	
13. května 2014	13. May 2014
Potvrdit opravu	
Intel představil třetí generaci rozhraní Thunderbolt, které známe především z	Intel introduce next generation of Thunderbold.
Potvrdit opravu	
Další 1 fotografie v galerii	Next 1 foto in gallery
Potvrdit opravu	

Obr. 3-18: Korektor – korektura

Přidat ovládaný jazyk – v nabídce „Upravit ovládané jazyky“ se klikne na odkaz „Přidat“ na řádku s příslušným jazykem.

Odebrat ovládaný jazyk – v nabídce „Upravit ovládané jazyky“ se klikne na odkaz „Odebrat“ na řádku s příslušným jazykem.

3.11. Instalace a použití

3.11.1. Prerekvizity

Pro použití aplikace je potřeba mít nainstalováno PHP v minimální verzi 3 a mySQL ve verzi 5 nebo novější. Instalaci PHP a MySQL lze dohromady provést nainstalováním software XAMPP, který je dostupný ke stažení na stránce <https://www.apachefriends.org/download.html>. Pro jednoduchou správu MySQL databáze doporučuji nainstalovat taktéž aplikaci *HeidiSQL*, kterou je možné stáhnout ze stránky <http://www.heidisql.com/download.php>.

3.11.2. Instalace

Na přiloženém CD se nachází soubor *apsp.zip*, který obsahuje kompletní kód aplikace včetně knihoven *Smarty* a *Zebra_database*. V případě použití defaultní instalace XAMPP pro Windows je potřeba rozbalit obsah souboru *apsp.zip* do adresáře `C:\xampp\htdocs\apsp\`. Pro správný chod aplikace požadují PHP soubory knihovny *Smarty* právo zápisu. To je nutné po instalaci zkontrolovat, jinak aplikace zobrazí chybové hlášení „Fatal error: Uncaught --> Smarty: unable to write file `./templates_c`“. V dalším kroku je potřeba spustit *Apache* server a MySQL databázi - to se provede spuštěním aplikace „XAMPP Control Panel“ a kliknutím na tlačítko „start“ na řádku Apache a MySQL.

3.11.3. Vytvoření MySQL databáze

Po spuštění aplikace *HeidiSQL* se připojení k databázi provede kliknutím na tlačítko „nové“. Do položky „Hostitel / IP“ se vyplní „localhost“ a do „Port“ „3306“.

Kód pro vytvoření potřebných tabulek je uveden v příloze A1, který se zkopíruje do pole SQL dotazu v *HeidiSQL* a spustí.

CD příloha také obsahuje soubor *all.sql* s kódem, který vytvoří databázi a nahraje do ní administrátora, testovací uživatele a články.

3.11.4. Vytvoření administrátor uživatele

Pro fungování aplikace je nutné vytvořit minimálně jednoho uživatele s administrátorským právem, který může schvalovat další přístupy. V příloze A2 je kód rozdělen na dvě části. V první části, kterou je nutné spustit, se provede vložení administrátora s přihlašovacím jménem/ „admin“ a heslem „admin“.

Tímto je aplikace připravena a dostupná na adrese <http://localhost/apsp/>.

3.11.5. Vytvoření testovacích uživatelů

V druhé části přílohy A2 je kód pro vytvoření testovacích uživatelů s přihlašovacími jmény: uzivate1 až uživatel5. Hesla jsou stejná jako jména.

3.11.6. Zprovoznění emailového oznamování

Pokud je požadováno emailové oznamování, je potřeba ho nastavit v konfiguračním souboru „C:\xampp\php\php.ini“ nebo v souboru aplikace C:\xampp\htdocs\apsp\include\user.php ve funkci send_email(), do kterého aplikace směřuje tři proměnné: \$to, \$subject a \$message pro vytvoření emailu. Pro lokální testování stačí v souboru „C:\xampp\php\php.ini“ odkomentovat parametr „sendmail_path="C:\xampp\mailtodisk\mailtodisk.exe"“ a generované emaily se budou ukládat do složky „C:\xampp\mailoutput“. Při nasazení aplikace na server je potřeba vyplnit adresu požadovaného SMTP serveru a jeho port.

3.11.7. Import testovacích článků

Na CD příloze se nachází soubor, který obsahuje osm testovacích článků vyexportovaných z redakčního systému *WordPress*. Články jsou převzaté ze stránky <http://technet.idnes.cz/thunderbolt-3-od-intelu-07b>.

ZÁVĚR

Úkolem mé diplomové práce bylo podrobné seznámení se s možnostmi multiplatformního programování pro internet, navrhnutí a naprogramování webové aplikace pro pokročilou správu textů a jejich překladů, včetně navrhnutí modulu pro import a export textů ze zvoleného redakčního systému.

Jako vývojový nástroj jsem zvolil programovací jazyk PHP, který je úzce navázán s jazyky HTML a CSS. Dále jsem využil software XAMPP, který v sobě obsahuje *apache* pro vytvoření lokálního serveru pro spouštění PHP kódu, a MySQL databázi, ve které jsou ukládány data z aplikace. Pro usnadnění práce jsem použil *HeidiSQL*, což je grafická nástavba pro práci s databází. *HeidiSQL* přehledně zobrazuje data uložená v databázi, jednoduše se v ní dají vytvořit a modifikovat tabulky a automaticky hned nabízí kód pro vytvoření tabulky v jiné databázi. Dále jsem ve své diplomové práci využil doplněk *Smarty*, což je šablonový systém, který odděluje PHP od HTML kódu a tím zpřehledňuje práci a případnou další modifikaci kódu. Tím, že oddělí tyto kódy, přináší do aplikace také vyšší bezpečnost. Dalším doplňkem použitým při vytváření aplikace je *Zebra_database*. Tento doplněk usnadňuje a zrychluje práci s databází. Zajišťuje textovou konverzi, urychluje přístup a stará se o navazování a ukončování spojení s databází.

Aplikace je rozdělena na části podle uživatelských účtů. Každý uživatelský účet v aplikaci má různá oprávnění a možnosti nastavené podle toho, co je úkolem specifického typu uživatelského účtu. Nejdůležitějším uživatelským účtem z pohledu aplikace je typ „administrátor“. Hlavním úkolem administrátora je schvalování dalších uživatelských účtů. Nový uživatel, který se do aplikace zaregistruje a přihlásí, nemá ze začátku žádná práva a nemůže s aplikací pracovat. Může však o specifický účet požádat a po jeho schválení může začít v aplikaci pracovat. Dalším důležitým účtem je typ „vedoucí překladatel“, který má za úkol definovat, do jakého jazyka se bude určitý článek překládat. Má možnost články importovat a po přeložení také exportovat do formátu pro redakční systém *WordPress*. Uživatel s účtem redaktora má možnost články v systému vytvářet. Mezi další účty patří účet „překladaře“ a „korektora“. Překladaři se po přihlášení aplikace zobrazí články, které jsou určené pro překlad. Části článků publikovaných na webu se uzavírají do HTML značek, které popisují význam části textu. Tyto značky se během překladu odstraní a po překladu se zpátky

doplní na původní místa. Z tohoto důvodu je článek při překladu rozdělen na části podle těchto značek a každá část (nadpis, odstavec, případně popis obrázku) se překládá zvlášť. Podobně to má i korektor, kterému se články zobrazují stejně rozděleně jako překladateli. Každou tuto část pak kontroluje zvlášť a případně ji může ještě opravit a opravu uložit.

Závěrem bych chtěl zmínit, že návrh a vývoj aplikace pro správu překladů redakčních systémů mi přišlo jako velmi zajímavý způsob, jak se seznámit s programováním webových aplikací. Bylo zde potřeba aplikaci vhodně navrhnout, přehledně naprogramovat, vhodně definovat tabulky v databázi pro ukládání článků a také vytvořit modul pro import a export pro spolupráci s jinou webovou aplikací.

POUŽITÁ LITERATURA

- [1] LUKÁŠ, Jiří. *Co je to redakční systém?* [online]. 05.05.2005 [cit. 2013-12-08]. Dostupné z: <http://www.supersvet.cz/view.php?cisloclanku=2005050501>
- [2] CHAPMAN, Cameron. Top 10 content management systems. [online]. 10.31.2011 [cit. 2013-12-08]. Dostupné z: <http://www.webdesignerdepot.com/2011/10/top-10-content-management-systems/>
- [3] WordPress. *About WordPress* [online]. [cit. 2013-12-08]. Dostupné z: <http://wordpress.org/about/>
- [4] WordPress. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-12-08]. Dostupné z: <http://en.wikipedia.org/wiki/WordPress>
- [5] JOOMLA. *Joomla! The CMS Trusted By Millions for their Websites* [online]. 2005 [cit. 2013-12-08]. Dostupné z: <http://www.joomla.org/>
- [6] DRUPAL. *Drupal - Open Source CMS | Drupal.orgs for their Websites* [online]. 2001 [cit. 2013-12-08]. Dostupné z: <https://drupal.org/>
- [7] KRAUS, Josef. *Nejlepší redakční a publikační systém – Živě.cz* [online]. 2012 [cit. 2013-12-08]. Dostupné z: <http://www.zive.cz/clanky/nejlepsi-redakcni-a-publikacni-system/sc-3-a-162731/default.aspx>
- [8] Apache. *The Apache HTTP Server Project* [online]. 1995 [cit. 2013-12-22]. Dostupné z: <http://httpd.apache.org/>
- [9] Web server developers: Market share of all sites. In: *Netcraft* [online]. 2013 [cit. 2013-12-22]. Dostupné z: <http://news.netcraft.com/archives/category/web-server-survey/>
- [10] BROŽEK, Erik. Server Apache, co je vlastně zač a co dokáže. [online]. 1998 [cit. 2013-12-22]. Dostupné z: <http://www.zive.sk/server-apache-co-je-vlastne-zac-a-co-dokaze/sc-3-a-153277/default.aspx>
- [11] Co je to databáze MySQL?. *ARTIC STUDIO* [online]. 2011 [cit. 2013-12-22]. Dostupné z: <http://www.artic-studio.net/slovnicek-pojmu/databaze-mysql/>

- [12] MySQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013 [cit. 2013-12-22]. Dostupné z: <http://cs.wikipedia.org/wiki/MySQL>
- [13] Historie jazyka PHP. *PHP (1) - Historie a budoucnost* [online]. 2004 [cit. 2013-12-22]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=171
- [14] *Smarty-Template engin* [online]. 2002 [cit. 2014-03-10]. Dostupné z: <http://www.smarty.net>
- [15] *Zebra_Database* [online]. 2006 [cit. 2014-03-11]. Dostupné z: <http://stefangabos.ro/php-libraries/zebra-database/>
- [16] HyperText Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-05-05]. Dostupné z: http://cs.wikipedia.org/wiki/HyperText_Markup_Language
- [17] Kaskádové styly. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2014-05-05]. Dostupné z: http://cs.wikipedia.org/wiki/Kask%C3%A1dov%C3%A9_styly
- [18] KOSEK, Jiří. PHP :Tvorba interaktivních internetových aplikací. Podrobný průvodce /Praha :Grada Publishing,1998. 1. vyd. 490 s. ISBN 8071693731
- [19] NARAMORE, Elizabeth. PHP5, MySQL, Apache: vytváříme webové aplikace / Vyd. 1. Brno:Computer Press, 2006. 813 s. ISBN 80-251-1073-7
- [20] KOFLER, Michael; ÖGGL, Berndt .PHP 5 a MySQL 5 : průvodce webového programátora, Vyd. 1. Brno : Computer Press, 2007. 607 s., ISBN: 978-80-251-1813-9

SEZNAM POUŽITÝCH ZKRATEK

CMS	Kontent Management Systém
CSS	Cascading Style Sheets
DBMS	Database managment system
FI	Form Interpreter
GNU GPL	GNU General Public License
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
MD5	Message-Digest algoritmus
MPM	MultiProcessing modulů
PHP	Hypertext Preprocessor
RS	Redakční systém
SGML	Standard Generalized Markup Language
SQL	Structured Query Language
URL	Uniform Resource Locator
XML	Extensible Markup Language

PŘÍLOHA A – SQL KÓD PRO PŘIPRAVENÍ DATABÁZE

A1 – Vytvoření databáze

```
CREATE DATABASE IF NOT EXISTS `stefanik`;
USE `stefanik`;

CREATE TABLE IF NOT EXISTS `article` (
  `article_id` INT(11) NOT NULL AUTO_INCREMENT,
  `origin_id` INT(11) NULL DEFAULT NULL,
  `source` VARCHAR(50) NULL DEFAULT NULL,
  `redactor_id` INT(11) NOT NULL,
  `article_origin_language` VARCHAR(50) NULL DEFAULT NULL,
  `is_final` VARCHAR(50) NULL DEFAULT NULL,
  `article_name` VARCHAR(50) NOT NULL,
  `article_content` VARCHAR(10000) NOT NULL,
  `translate_to` VARCHAR(50) NULL DEFAULT NULL,
  `translate_progress` FLOAT NULL DEFAULT NULL,
  `translate_final` VARCHAR(50) NULL DEFAULT NULL,
  `article_name_translated` VARCHAR(100) NULL DEFAULT NULL,
  `article_content_translated` VARCHAR(10000) NULL DEFAULT NULL,
  `translator_id` INT(11) NULL DEFAULT NULL,
  `translator_name` VARCHAR(50) NULL DEFAULT NULL,
  `corrector_id` INT(11) NULL DEFAULT NULL,
  `corrector_name` VARCHAR(50) NULL DEFAULT NULL,
  `corrector_progress` FLOAT NULL DEFAULT NULL,
  `corrected` VARCHAR(1) NULL DEFAULT NULL,
  `tran_date` VARCHAR(50) NULL DEFAULT NULL,
  `corr_date` VARCHAR(20) NULL DEFAULT NULL,
  `end_date` VARCHAR(20) NULL DEFAULT NULL,
  INDEX `ID` (`article_id`)
)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB
AUTO_INCREMENT=1;

CREATE TABLE IF NOT EXISTS `article_translate` (
  `article_id` INT(11) NULL DEFAULT NULL,
  `line_id` INT(11) NULL DEFAULT NULL,
  `text` VARCHAR(10000) NULL DEFAULT NULL,
  `is_tag` VARCHAR(1) NULL DEFAULT NULL,
  `translate_to` VARCHAR(10) NULL DEFAULT NULL,
  `translated` VARCHAR(10000) NULL DEFAULT NULL,
  `corrected` VARCHAR(1) NULL DEFAULT NULL
```

```

)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `article_wp` (
  `article_wp_id` INT(11) NOT NULL AUTO_INCREMENT,
  `in_use` VARCHAR(1) NULL DEFAULT 'N',
  `language1` VARCHAR(50) NULL DEFAULT NULL,
  `wxr_version` VARCHAR(50) NULL DEFAULT NULL,
  `title` VARCHAR(50) NULL DEFAULT NULL,
  `creator` VARCHAR(50) NULL DEFAULT NULL,
  `encoded` MEDIUMTEXT NULL,
  `status1` VARCHAR(1000) NULL DEFAULT NULL,
  `post_type` VARCHAR(1000) NULL DEFAULT NULL,
  INDEX `ID` (`article_wp_id`)
)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB
AUTO_INCREMENT=9000000;

CREATE TABLE IF NOT EXISTS `member` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(30) NOT NULL,
  `password` VARCHAR(100) NOT NULL,
  `fname` VARCHAR(30) NOT NULL,
  `lname` VARCHAR(30) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `admin` VARCHAR(1) NULL DEFAULT NULL,
  `leader` VARCHAR(1) NULL DEFAULT NULL,
  `redactor` VARCHAR(1) NULL DEFAULT NULL,
  `translator` VARCHAR(1) NULL DEFAULT NULL,
  `corrector` VARCHAR(1) NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
)
COLLATE='latin1_swedish_ci'
ENGINE=MyISAM
AUTO_INCREMENT=1;

CREATE TABLE IF NOT EXISTS `member_accounts` (
  `account_id` INT(11) NOT NULL AUTO_INCREMENT,
  `username` VARCHAR(30) NOT NULL,
  `account` VARCHAR(10) NOT NULL,
  `account_active` VARCHAR(1) NOT NULL,
  PRIMARY KEY (`account_id`)
)

```

```

COLLATE='latin1_swedish_ci'
ENGINE=MyISAM
AUTO_INCREMENT=1;

CREATE TABLE IF NOT EXISTS `member_languages` (
  `user_id` INT(11) NULL DEFAULT NULL,
  `username` VARCHAR(50) NULL DEFAULT NULL,
  `language` VARCHAR(50) NULL DEFAULT NULL
)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `language` (
  `language_id` VARCHAR(50) NULL DEFAULT NULL,
  UNIQUE INDEX `language_id` (`language_id`)
)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB;

INSERT INTO `language`(`language_id`) VALUES ("cz");
INSERT INTO `language`(`language_id`) VALUES ("en");
INSERT INTO `language`(`language_id`) VALUES ("de");
INSERT INTO `language`(`language_id`) VALUES ("fr");

CREATE TABLE IF NOT EXISTS `account_list` (
  `accounts` VARCHAR(50) NULL DEFAULT NULL,
  `accountcz` VARCHAR(50) NULL DEFAULT NULL
)
COLLATE='latin1_swedish_ci'
ENGINE=InnoDB;

INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("admin","Administr&aacute;tor");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("leader","Spr&aacute;vce p&#x0159;ekladu");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("corrector","Korektor");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("redactor","Redaktor");
INSERT INTO `account_list`(`accounts`, `accountcz`)
VALUES ("translator","P&#x0159;ekladatel");

```

A2 – Vložení uživatelů

```
-----vložení administrátora-----
INSERT INTO `member` (`id`, `username`, `password`, `fname`,
`lname`, `email`, `admin`, `leader`, `redactor`, `translator`,
`corrector`) VALUES
    (1, 'admin', '21232f297a57a5a743894a0e4a801fc3', 'adminjmeno',
'adminprijmeni', 'admin@admin.cz', 'Y', 'Y', 'Y', 'Y', 'Y');

INSERT INTO `member_accounts` (`account_id`, `username`, `account`,
`account_active`) VALUES
    (1, 'admin', 'admin', 'Y'),
    (2, 'admin', 'leader', 'Y'),
    (3, 'admin', 'redactor', 'Y'),
    (4, 'admin', 'translator', 'Y'),
    (5, 'admin', 'corrector', 'Y');

INSERT INTO `member_languages` (`user_id`, `username`, `language`)
VALUES
    (1, 'admin', 'cz'),
    (1, 'admin', 'de'),
    (1, 'admin', 'en'),
    (1, 'admin', 'fr');

-----vložení testovacích uživatelů-----
INSERT INTO `member` (`id`, `username`, `password`, `fname`,
`lname`, `email`, `admin`, `leader`, `redactor`, `translator`,
`corrector`) VALUES
    (2, 'uzivatel1', '5e351ae0e063c2504468e6e8a31cf4fc',
'uzivatel1jmeno', 'uzivatel1prijmeni', 'uzivatel1@uzivatel1.cz',
NULL, 'Y', 'Y', NULL, NULL),
    (3, 'uzivatel2', '372641748bbe6888688935ea0bcd8091',
'uzivatel2jmeno', 'uzivatel2prijmeni', 'uzivatel2@uzivatel2.cz',
NULL, NULL, NULL, 'Y', 'Y'),
    (4, 'uzivatel3', '7903f786b8164c458b3665edbf2d3279',
'uzivatel3jmeno', 'uzivatel3prijmeni', 'uzivatel3@uzivatel3.cz',
NULL, NULL, NULL, 'Y', 'Y'),
    (5, 'uzivatel4', 'e811b717fda20cd3cde943eddbadc261',
'uzivatel4jmeno', 'uzivatel4prijmeni', 'uzivatel4@uzivatel4.cz',
NULL, NULL, NULL, 'Y', 'Y'),
    (6, 'uzivatel5', 'e4a3bbcdde85aaec48c99949cedf4188',
'uzivatel5jmeno', 'uzivatel5prijmeni', 'uzivatel5@uzivatel5.cz',
NULL, NULL, NULL, 'Y', 'Y');

INSERT INTO `member_accounts` (`account_id`, `username`, `account`,
`account_active`) VALUES
    (6, 'uzivatel1', 'leader', 'Y'),
    (7, 'uzivatel2', 'translator', 'Y'),
```

```

(8, 'uzivatel3', 'translator', 'Y'),
(9, 'uzivatel4', 'translator', 'Y'),
(10, 'uzivatel5', 'translator', 'Y'),
(11, 'uzivatel1', 'redactor', 'Y'),
(12, 'uzivatel2', 'corrector', 'Y'),
(13, 'uzivatel3', 'corrector', 'Y'),
(14, 'uzivatel4', 'corrector', 'Y'),
(15, 'uzivatel5', 'corrector', 'Y');

INSERT INTO `member_languages` (`user_id`, `username`, `language`)
VALUES
(2, 'uzivatel1', 'cz'),
(2, 'uzivatel1', 'de'),
(2, 'uzivatel1', 'en'),
(2, 'uzivatel1', 'fr'),
(3, 'uzivatel2', 'cz'),
(3, 'uzivatel2', 'en'),
(4, 'uzivatel3', 'cz'),
(4, 'uzivatel3', 'de'),
(4, 'uzivatel3', 'en'),
(5, 'uzivatel4', 'cz'),
(5, 'uzivatel4', 'en'),
(5, 'uzivatel4', 'fr'),
(6, 'uzivatel5', 'cz'),
(6, 'uzivatel5', 'de'),
(6, 'uzivatel5', 'fr');

```

SEZNAM PŘÍLOH NA CD

- Text diplomové práce – DP_Pavel_Stefanik_xstefa06_78725.pdf
- Kompletní kód aplikace včetně knihoven – soubor: apsp.zip
- SQL kód pro vytvoření databáze – soubor: database.sql
- SQL kód pro vytvoření testovacích uživatelů – soubor: users.sql
- SQL kód pro vytvoření kompletní databáze včetně uživatelů a článků – soubor: all.sql
- XML soubor s osmi testovacími články pro import – soubor: article_wp.xml