



Středoškolská technika 2010

Setkání a prezentace prací středoškolských studentů na ČVUT

WEBOVÉ ROZHŘANÍ PRO PŘEKLADATELE GNU GETTEXT

Zdeněk Vinduška

Vyšší odborná škola a Střední průmyslová škola Varnsdorf
Mariánská 1100, Varnsdorf 5

Obsah

1. Úvod.....	3
2. Použité programovací jazyky	4
2.1. PHP.....	4
2.2. HTML.....	5
2.3. CSS.....	5
2.4. MySQL.....	6
3. Metodika a postup práce.....	6
3.1. Databáze	6
3.2. Funkce	11
3.3. Session.....	13
3.4. Menu.....	14
3.5. Registrace účtu	16
3.6. Přihlášení účtu	17
3.7. Odhlášení účtu.....	18
3.8. Smazání účtu	19
3.9. Uložení dokumentu	20
3.10. Schválení dokumentu	21
3.11. Překládání dokumentu.....	22
3.12. Čtení dokumentu	25
3.13. Ukládání dokumentu z databáze do souboru.....	25
3.14. Stažení dokumentu	25
3.15. Mazání dokumentu	26
3.16. Slovník a seznam frází	26
3.17. Uživatelé.....	27
4. Závěr.....	28
5. Seznam obrázků a diagramů.....	29
6. Použitá literatura.....	30

1. Úvod

Tato práce má sloužit jako online překladač licencí ve formě souborů XML. Licence jsou psány v anglickém jazyce a překládají se do ostatních potřebných jazyků. Tyto licence překládají překladatelé, kteří se tímto zabývají na svém PC a právě toto je důvod, proč byl navrhnout tento překladač, který by měl umožnit efektivní překládání z jakéhokoliv místa s připojením internetu.

Na začátku je nutné vymyslet databázi v MySQL pro ukládání a práci se soubory. Dále vymyslet jednoduché uživatelské rozhraní, které by mělo obsahovat vše potřebné pro práci s dokumenty. V této aplikaci jde především o překládání licencí. V této části se nechám inspirovat programem PoEdit, který je právě pro tuto práci používán překladateli. Nejtěžší zde bude právě vymyslet algoritmus pro vypsání potřebných částí souboru a ukládání následného upravení řetězců. Dále bude za potřebí slovník a seznam často používaných frází pro efektivnější překládání.

2. Použité programovací jazyky

Zde vám ve stručnosti popíši programovací jazyky, které jsem v této práci použil. Jedná se o jazyk pro skriptování *PHP*, pro tvorbu webových stránek *HTML*, pro grafickou úpravu stránek *CSS* a dále databázi *MySQL* pro práci s daty.

2.1. PHP

Vznik tohoto skriptovacího jazyku je datován rokem 1994, kdy jistý pan Rasmus Lerdorf se rozhodl vytvořit jednoduchý systém pro počítání přístupu ke stránkám. Po menším vývoji byla vydána sada skriptů pod názvem “*Personal Home Page Tools*“ neboli PHP. V polovině roku 1995 byl systém PHP spojen s nástrojem “*Form Interpreter*“ a vzniklo tak PHP/FI 2.0 a tento systém byl postupně velmi rozšířen a získal si velikou proslulost. Po 3 letech se objevuje další verze PHP a to verze 3.0, která měla více funkcí a byla rychlejší než předešlá verze. V dnešní době se používá už verze 4.0, ve které je i podpora potřebných XML souborů.

PHP je nezávislé na platformě a dnes se nejvíce používá se serverem Apache. Je to bezplatný software, který je dnes nejvíce využíván pro tvorbu dynamických webových stránek. Programování v PHP není nejtěžší, je zde mnoho užitečných funkcí, a pokud víme, jak to naprogramovat pomocí použitých funkcí, pak není problém s vyřešením zadaného úkolu. Lehce se implementuje do HTML kódu a pracuje s databázemi. V PHP se dají napsat různé aplikace, ať už to jsou internetové obchody, diskusní fóra, vyhledávače, různá počítačla a redakční systémy.

PHP kód je vždy ohraničen těmito znaky: `<?php` skript `?>`. Další informace o tomto skriptovacím jazyce naleznete na oficiálních stránkách <http://www.php.net/>.

2.2. HTML

Základem HTML (HyperText Markup Language) jsou hypertextové odkazy neboli linky, bez kterých by tento jazyk nebyl nikdy tak populární. Proto celá historie HTML začíná právě u vývoje hypertextů. Tento vývoj začal už ve 40. letech minulého století, kdy V. Bush představil hypertextový systém Memex. V. Bush byl poté následován dalšími vývojáři. První neoficiální verze HTML byla vyvinuta roku 1989 T. Bernersem-Leeem a R. Caillauem, pracovníky CERNu. Tato verze byla spojena s protokolem HTTP (HyperText Transfer Protocol). Další rozvoj byl umožněn po vytvoření prvního prohlížeče Mosaic. Oficiální verze HTML 2.0 byla po dlouhém vývoji vydána v roce 1996. Poslední verze, která byla schválena W3C (World Wide Web Consortium), byla vydána ke konci roku 1999 jako verze 4.01. Vývoj HTML tímto rokem skončil, ale i nadále má tento jazyk své následovníky např. XHTML (Extensive HyperText Markup Language) a XML (Extensive Markup Language).

HTML nám slouží pro tvorbu webových stránek pomocí určitých tagů, které používáme pro formátování textu, obrázků a tabulek. Pro správné zobrazení v různých prohlížečích musí být kód správně napsán podle norem. Kód můžeme validovat na stránkách <http://www.w3c.org/>.

2.3. CSS

První norma CSS (Cascading Style Sheets) byla zveřejněna roku 1994. V roce 1998 byla vydaná další specifikace CSS 2 a v dnešní době se pracuje na verzi CSS 3.

CSS se využívá ke grafické úpravě webových stránek, a tedy k formátování obsahu HTML dokumentů. Tento jazyk rozšiřuje schopnosti formátování. Použitím různých stylů u elementů lze určit, jak budou dané elementy vypadat. Na začátku si nadefinujeme různé elementy a dále používáme jen tagy přiřazené k nim. Docílíme tím přehlednějšího a kratšího kódu. Pokud chceme v celém kódu změnit barvu, velikost či typ písma u elementů, pak to bude otázka přepsání jednoho řádku, čímž si ušetříme dost času. Formátování lze napsat do souboru a následně importovat do kódu, tím pádem při potřebné změně změníme jen jeden soubor a na všech stránkách se změna patřičně objeví.

2.4. MySQL

Databáze je místo, kam se ukládají potřebné údaje. Mezičlánkem mezi databází a aplikačním programem je SŘBD (Systém Řízení Báze Dat). SŘBD má za úkol spravovat data, sdílet je nejefektivnějším způsobem a zajišťovat jejich integritu. Existují placené programy SŘBD např. Oracle, MS SQL Server a nadále freeware programy např. MySQL a PostgreSQL. Data se ukládají do tabulek. S databází komunikujeme pomocí jazyka SQL.

MySQL vznikl v roce 1996. V dnešní době je to jedna z nejvíce rozšířených databází, která se používá na většině webhostingů. Oficiální stránka je <http://www.mysql.com/>.

3. Metodika a postup práce

V této kapitole vám vysvětlím věcnou stránku mé práce, databázi, skripty a mé myšlenkové postupy. Chtěl bych se také zmínit o tom, že jsem se PHP učil “za chodu“ pomocí seriálu o PHP na <http://www.linuxsoft.cz/php/> a moje praxe s PHP není tak veliká. Také jsem se nechal inspirovat tímto seriálem, pokud jsem nevěděl, jak daný úkol řešit.

3.1. Databáze

3.1.1. Slovní popis

V databázi máme tabulku pro uživatele, fráze, slovník a dokumenty. Při schválení dokumentu se vytvoří další tabulka s názvem *dok.číslo* (př. dok3), ve které je dokument rozdělen na řádky.

3.1.2. Datové prvky

U uživatelů evidujeme id, jméno, heslo, zda je admin či překladatel.

U dokumentů evidujeme id, název, schválen, dokončen.

Ve frázích evidujeme id, českou a anglickou frázi.

Ve slovníku evidujeme id, české a anglické slovo.

V tabulce schváleného dokumentu evidujeme id a řádek.

3.1.3. Vztahy

Ukládání dokumentu (UD) – Jeden uživatel může uložit jeden nebo více dokumentů.

Schválení dokumentu (SD) – Jeden admin může schválit jeden nebo více dokumentů vyhovujících pro překlad.

Přeložení dokumentu (PD) – Jeden překladatel může přeložit jeden nebo více dokumentů.

Čtení dokumentu (ČD) – Jeden uživatel může číst jeden nebo více dokumentů.

Stažení souboru (SS) – Jeden uživatel si může stáhnout jeden nebo více hotových překladů.

Vytvoření fráze (VF) – Jeden admin může vytvořit jednu nebo více frází.

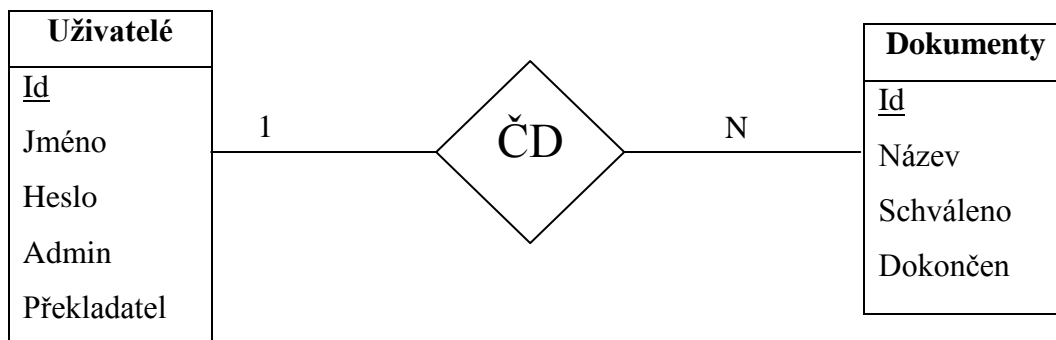
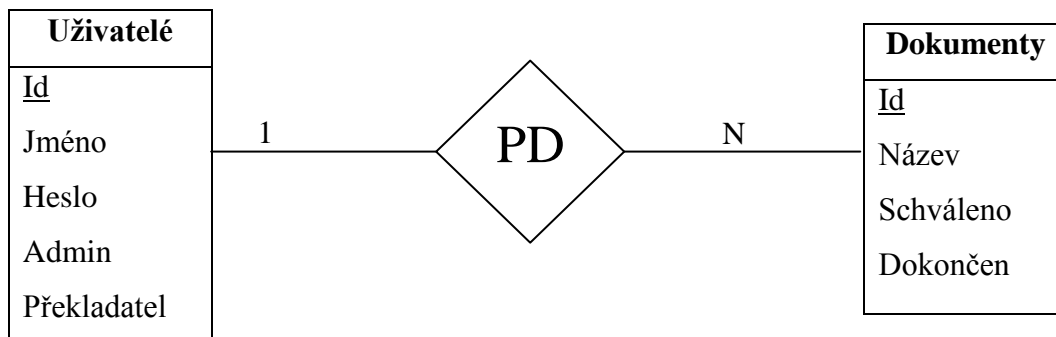
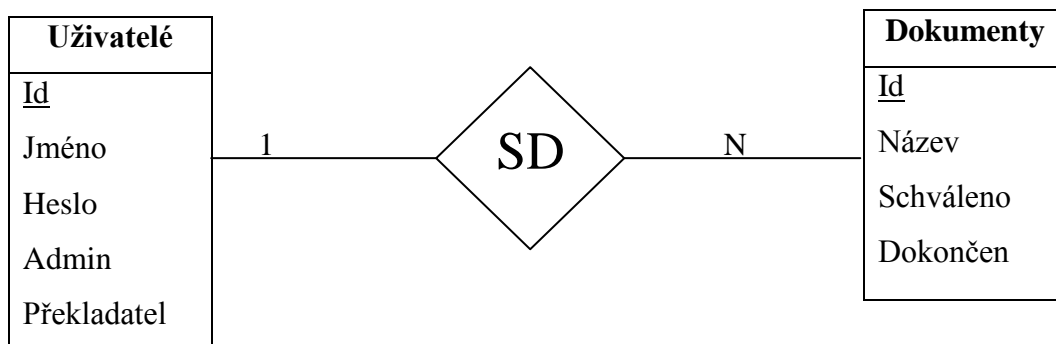
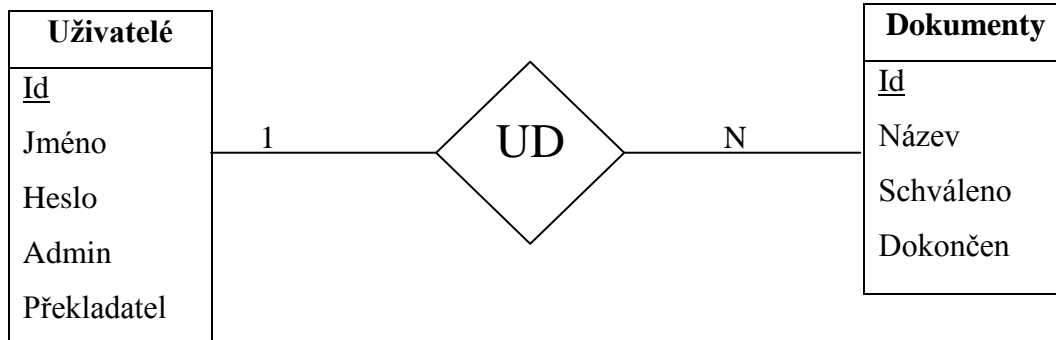
Čtení fráze (ČF) – Jeden uživatel může číst jednu nebo více frází.

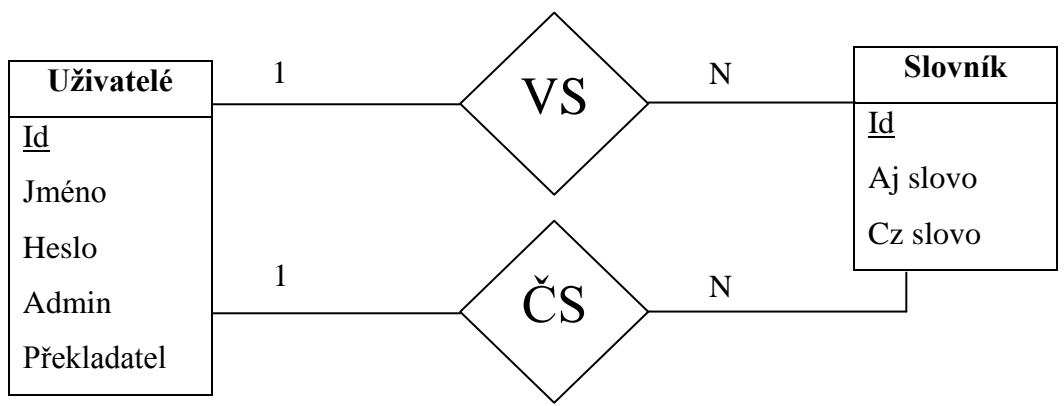
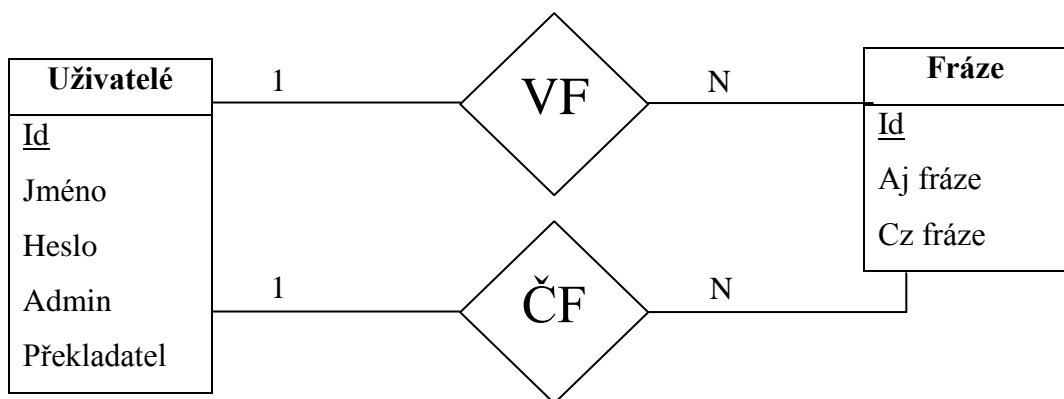
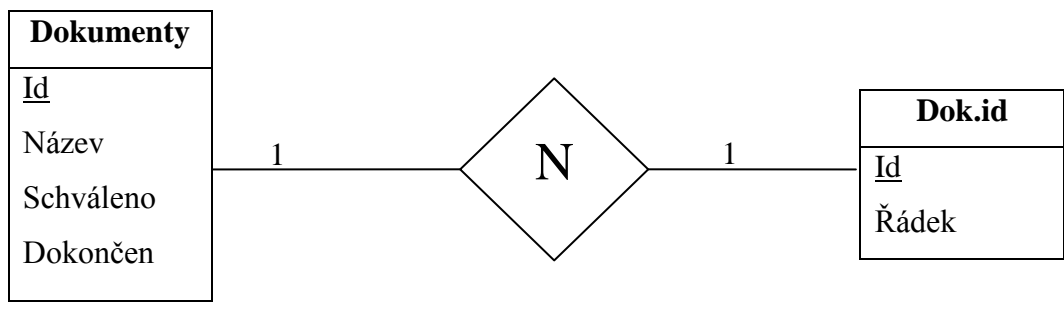
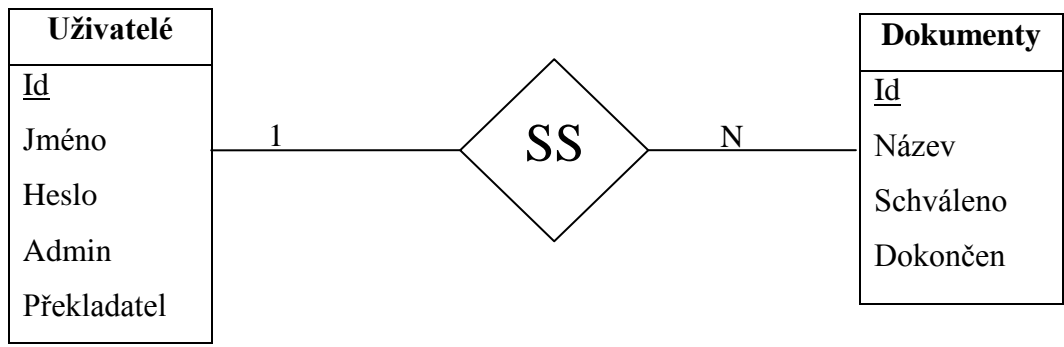
Vytvoření slovní dvojice (VS) – Jeden admin může vytvořit jednu nebo více slovních dvojic ve slovníku.

Čtení slovní dvojice (ČS) – Jeden uživatel může číst jednu nebo více slovních dvojic ze slovníku.

Náleží (N) – Jednomu schválenému dokumentu náleží jen jedna tabulka, ve které je rozdělen.

3.1.4. E-R Schéma





3.1.5. Lineární vztahy schématu

E:

uživatelé (id, jméno, heslo, admin, překladatel);

dokumenty (id, název, schváleno, dokončen);

dok.id (id, řádek);

fráze (id, aj fráze, cz fráze);

slovník (id, aj slovo, cz slovo);

R:

UD (uživatelé, dokumenty);

SD (uživatelé, dokumenty);

PD (uživatelé, dokumenty);

ČD (uživatelé, dokumenty);

SS (uživatelé, dokumenty);

N (dokumenty, dok.id);

VF (uživatelé, fráze);

ČF (uživatelé, fráze);

VS (uživatelé, slovník);

ČS (uživatelé, slovník);

3.2. Funkce

Všechny funkce jsou uloženy v souboru *function.php*, který je vložen do hlavní stránky *index.php*. Lze tak rychle a přehledně upravovat a dopisovat funkce. Funkce jsou seřazeny chronologicky za sebou, podle postupu v práci.

```
<?
INCLUDE "./CONFIG.php";
$GLOBALS["link"]=mysql_connect(SQL_HOST, SQL_USERNAME,
SQL_PASSWORD);
mysql_select_db(SQL_DBNAME);
?>
```

Tato proměnná nám slouží při práci s databází pomocí SQL dotazů. Vždy musíme do SQL dotazu napsat k jaké databázi se chceme připojit. Tento systém nám chrání naše údaje, protože se vždy napíše a zobrazí jen *\$GLOBALS["link"]*, ve kterém jsou uloženy přístupové údaje k databázi a tudíž je nikdo nevidí.

Pokud potřebujeme rychle změnit tyto údaje, pak přepíšeme jen jeden soubor a vše je změněno, takže si ušetříme mnoho času, a také nemusíme do SQL dotazů psát dlouhé přístupové údaje.

Tato funkce *ukazatclanek()* nám zajišťuje zobrazení zvolené stránky. Pokud je zvolena nějaká stránka, tak se její název uloží do odkazu takovýmto způsobem *index.php?stranka=registrace*, tato funkce přečte parametr *stranka* a dále zjistí, zda stránka existuje, pokud ano, tak se zobrazí, pokud ne, tak se zobrazí stránka informující o nenalezení stránky. Jako základní stránka je nastavena stránka *novinky.php*.

Funkce *iduzivatele()* se nám táže databáze, zda existuje účet, který má zadané jméno, uložené v *\$jmeno*, a heslo, uložené v *\$heslo*. Pokud tento účet existuje, vrátí se číselná hodnota *id*, při nevyhovění se vrátí hodnota FALSE. Tato funkce se využívá při nastavování SESSION (Kap. 3.3).

Funkce *nazevuzivatele()* nám vrátí název uživatele neboli jeho jméno. Databáze se tážeme, zda existuje uživatel s *id*, a pokud existuje, vrátí se nám jeho uživatelské jméno.

Funkce *jmenovdb()* nám zjišťuje, zda existuje v databázi zadané jméno *\$jmeno*. Po odpovědi se vyhodnotí, zda existuje (TRUE) nebo neexistuje (FALSE).

Tyto dvě funkce *jeadmin ()* a *jeprekladatel ()* nám vrací logickou hodnotu po dotazu, zda je daný účet s daným *id* nastaven na admina či překladatele. Tyto funkce se využívají při zobrazování nabídek v menu (Kap. 3.4).

Funkce *dokumentvdb ()* nám vrátí logickou hodnotu TRUE (1) nebo FALSE (0), podle odpovědi na dotaz, zda dokument s názvem v *\$nazev_dokumentu* již existuje na portále, či nikoliv. Zde jsou logické stavy invertně, protože se ptáme, zda je volno pro daný dokument.

Funkce *ukazatdokument ()* nám stejně jako funkce *ukazatclanek ()* čerpá parametr z odkazu, který je zapsán ve tvaru *index.php?stranka=dokument_prekladani2&doku=mixer_applet2.HEAD.po* a vrací celý název dokumentu v proměnné, s kterým dále pracujeme.

Funkce *iddokument()* se váže na předešlou. Název souboru je v proměnné *\$nazev* a pomocí dotazu zjistíme *id*, které si vrátíme jako výsledek.

Funkce *delka_textu ()* se používá při vypisování zkráceného textu při překládání a čtení souborů online. Tato funkce zjistí, zda daný řetězec *\$delka* má více znaků než je 39. Delší řetězec se zkrátí na hodnotu 39 znaků a pokud je řetězec kratší, vrací se zpět nezměněná hodnota délky.

Funkcí *zbyva ()* se tážeme, zda existují dokumenty, které ještě nejsou schváleny. Počet neschválených dokumentů se pak ukáže jako číselný údaj, který je uložen v proměnné *\$pocet*. Tato funkce je použita v menu pro adminy, aby věděli, zda přibyly nějaké nové dokumenty ke schválení.

Funkce *iduzivatele2()* nám jako předešlé *ukazatdokument ()* a *ukazatclanek ()* čerpá parametr z odkazu, který je ve tvaru *index.php?stranka=uzivatel&uz=22*, a vrací *id* hodnotu přiřazenou k určitému uživateli. Tato funkce se využívá při změně práv (Kap. 3.17).

3.3. Session

Mohli jsme si vybrat mezi cookies a sessions. Vybrali jsme sessions, protože cookies jsou složitější a pro naši práci postačí jednoduché sessions.

Zde máte vysvětlení funkce sessions:

- Pokud PHP dostane příkaz, že má začít session, nejprve zjistí, zda session již neběží. Pokud neexistuje, vytvoří ji, pokud existuje, připojí se k ní.
- Nastavíme si libovolnou proměnnou pro tuto session a server si ji pamatuje mezi stránkami. Nepřenášejí se všechny údaje spojené s touto session, nýbrž jen identifikátor samotné session.
- O vše se stará PHP, takže nemusíte měnit odkazy pro přenesení identifikátoru.
- Session lze kdykoliv ukončit. Pokud ji neukončíte, ukončí se zavřením prohlížeče.

```
<?
ob_start();
session_start();

IF ($_GET["stranka"]=="prihlaseni")
{
    $id=iduzivatele($_POST["jmeno"], $_POST["heslo"]);
    IF ($id>=1)
    {
        $_SESSION["id"]=$id;
    }
}
ELSEIF ($_GET["stranka"]=="odhlaseni")
{
    UNSET($_SESSION["id"]);
}

ob_end_flush();
?>
```

Pokud při přihlášení souhlasí jméno `$_POST["jmeno"]` a heslo `$_POST["heslo"]`, pak se vrátí hodnota `id` daného uživatele, která se uloží do session. S touto proměnnou `$_SESSION["id"]` pak dále pracujeme při ověřování, zda je někdo přihlášen a jaké má pravomoce, např. v menu (Kap. 3.4).

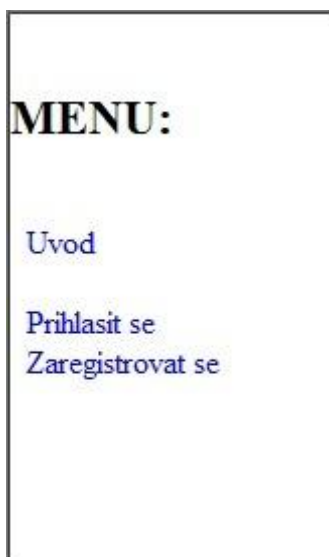
Pro odhlášení stačí pouze navštívit stránku *odhlaseni* a session se smaže.

```
<?
IF (!ISSET($_SESSION["id"]))
{
    ECHO "<BR><P align=\"center\">Pro pokračovani se musite
prihlasit!<BR>
    <A
href=\"index.php?stranka=prihlaseni\">Prihlaseni</A></P>";
}
ELSE
{
    IF (jeprekladatel($_SESSION["id"])==FALSE)
    {
        ECHO "<BR><P align=\"center\">Nemate dostatecna
prava!<BR>";
    }
    ELSE
    {
        //zobrazená stránka
    }
}
?>
```

Zde máme příklad využití proměnné `$_SESSION["id"]`. Pokud je session nastavená, uživatel je tedy přihlášen, pokračuje dále, jinak zobrazí varovnou hlášku. Dále zde v příkladu zjišťujeme, zda je daný účet nastaven na překladatele. Pokud je tento účet nastaven, zobrazí se stránka pro práci, při nevyhovění se zobrazí varovná hláška.

3.4. Menu

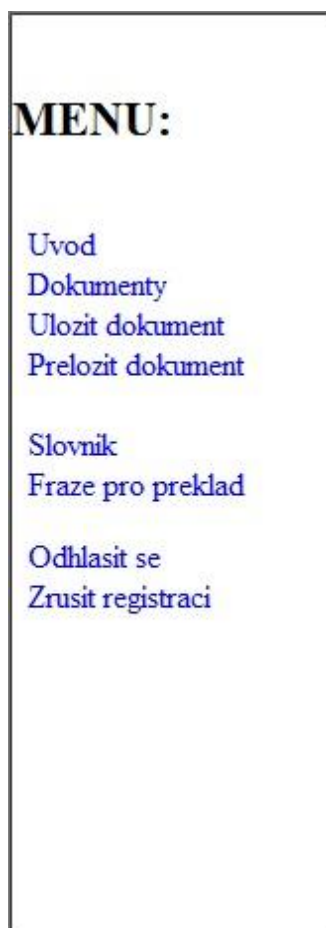
Menu jsem se snažil udělat co nejpřehlednější, aby související odkazy byly u sebe a nemuseli jste je v menu hledat. O to složitější je zdrojový kód, který jsem musel pro toto menu napsat.



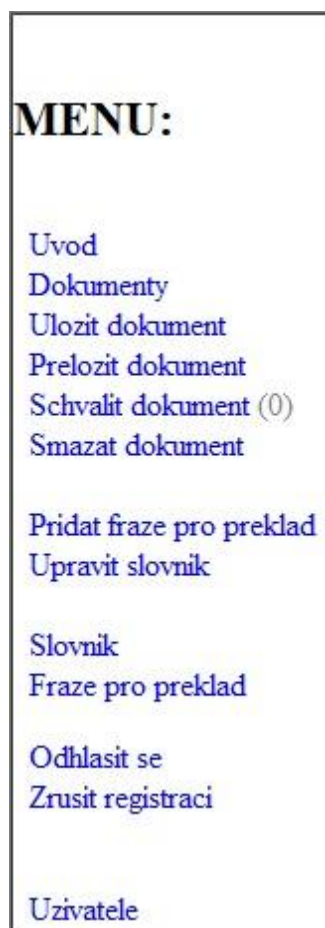
[Obr. 3.4.1. Menu nepřihlášeného uživatele]



[Obr. 3.4.2. Menu uživatele bez práv]



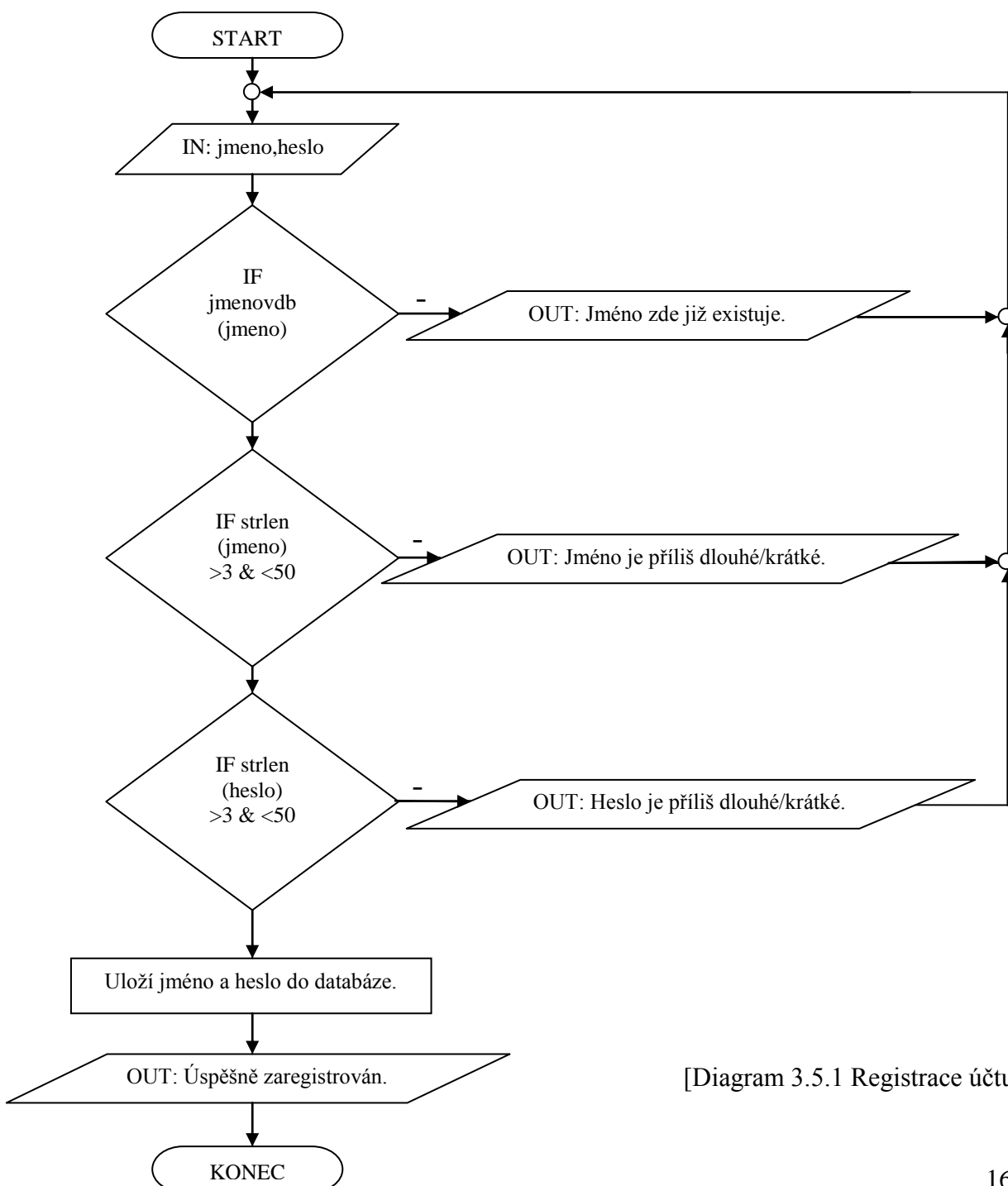
[Obr. 3.4.3. Menu překladatele]



[Obr. 3.4.4. Menu admina]

Neděste se tohoto zápisu menu, doopravdy to je přehlednější. Jak si můžete všimnout, používáme zde několik funkcí, o kterých jsme se již zmiňovali (Kap. 3.2). Pomocí `ISSET($_SESSION["id"])` zjišťujeme, zda je session vytvořena a uživatel tedy přihlášen. Můžeme poté ověřovat, zda je či není adminem nebo překladačem, a pak podle výsledku vypisovat menu. Jak si můžete všimnout, funkce `jeadmin()` je zde několikrát a to právě z důvodu přehlednosti, protože by musely být u sebe všechny odkazy pro admina a pak by vše bylo na přeskáčku.

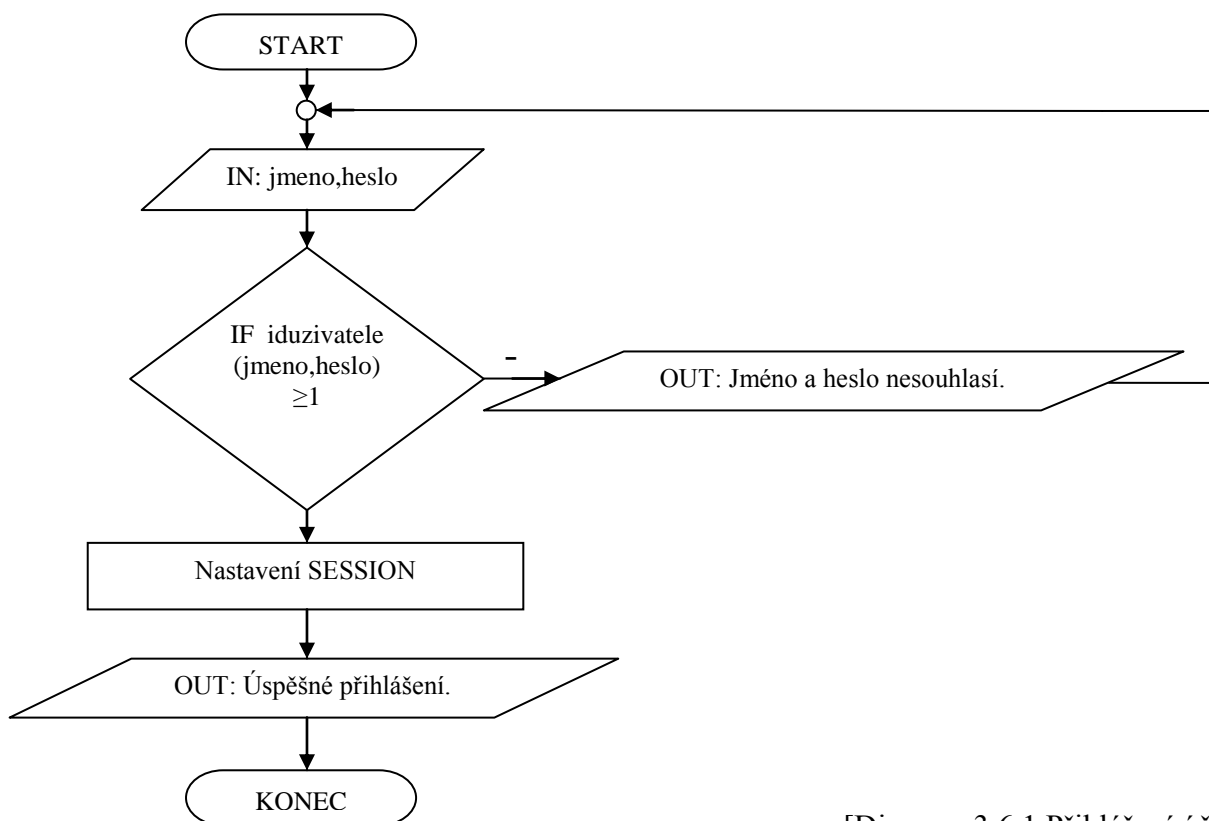
3.5. Registrace účtu



[Diagram 3.5.1 Registrace účtu]

Při registraci zadáváme pouze uživ. jméno a heslo. V prvním kroku zjistíme, zda uživ. jméno již není v databázi. Pokud není v databázi, testujeme délku vloženého jména a hesla, pokud vyhovují, jsou údaje uloženy do databáze a uživatel se může přihlásit. Při nevyhovění se vypíše varovná hláška a uživatel je vyzván k opravě údajů.

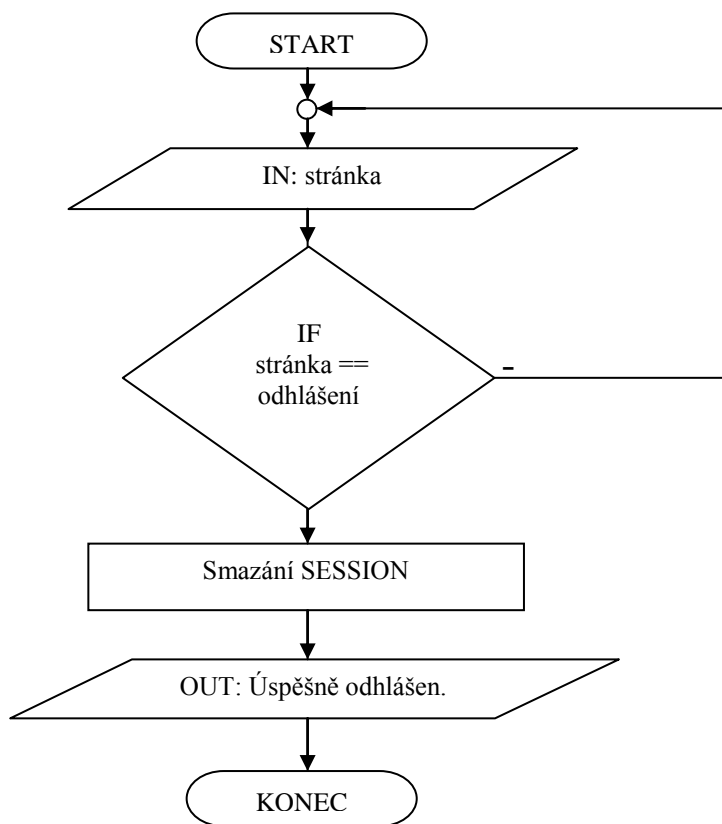
3.6. Přihlášení účtu



[Diagram 3.6.1 Přihlášení účtu]

Při přihlášení zadáme naše uživ. jméno a heslo, pokud se vrátí naše id, při správném zadání údajů, jsme přihlášení, v opačném případě se vypíše varovná zpráva.

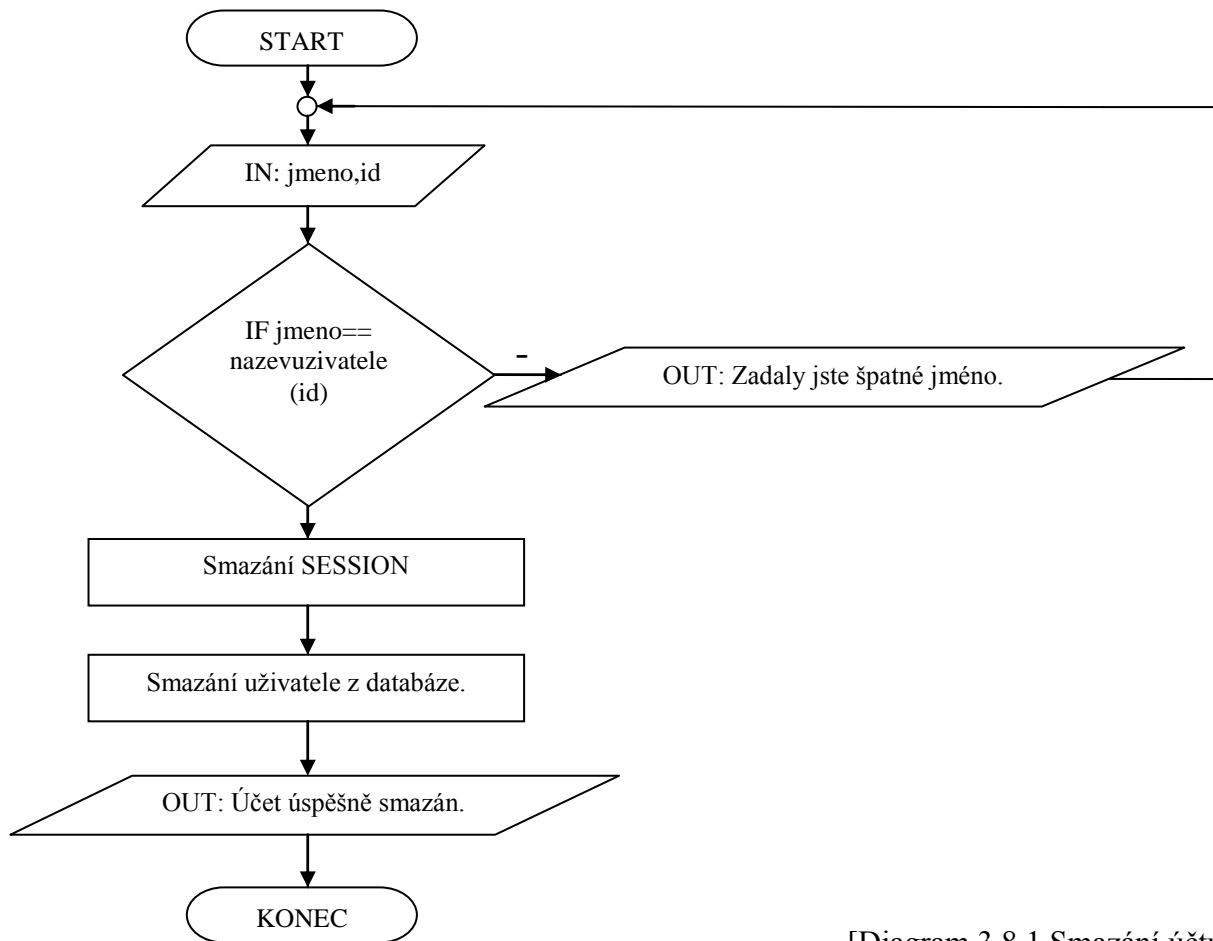
3.7. Odhlášení účtu



[Diagram 3.7.1 Odhlášení účtu]

Zde se čeká, až se navštíví stránka *odhlaseni*, až se tak stane, smaže se session a uživatel je odhlášen.

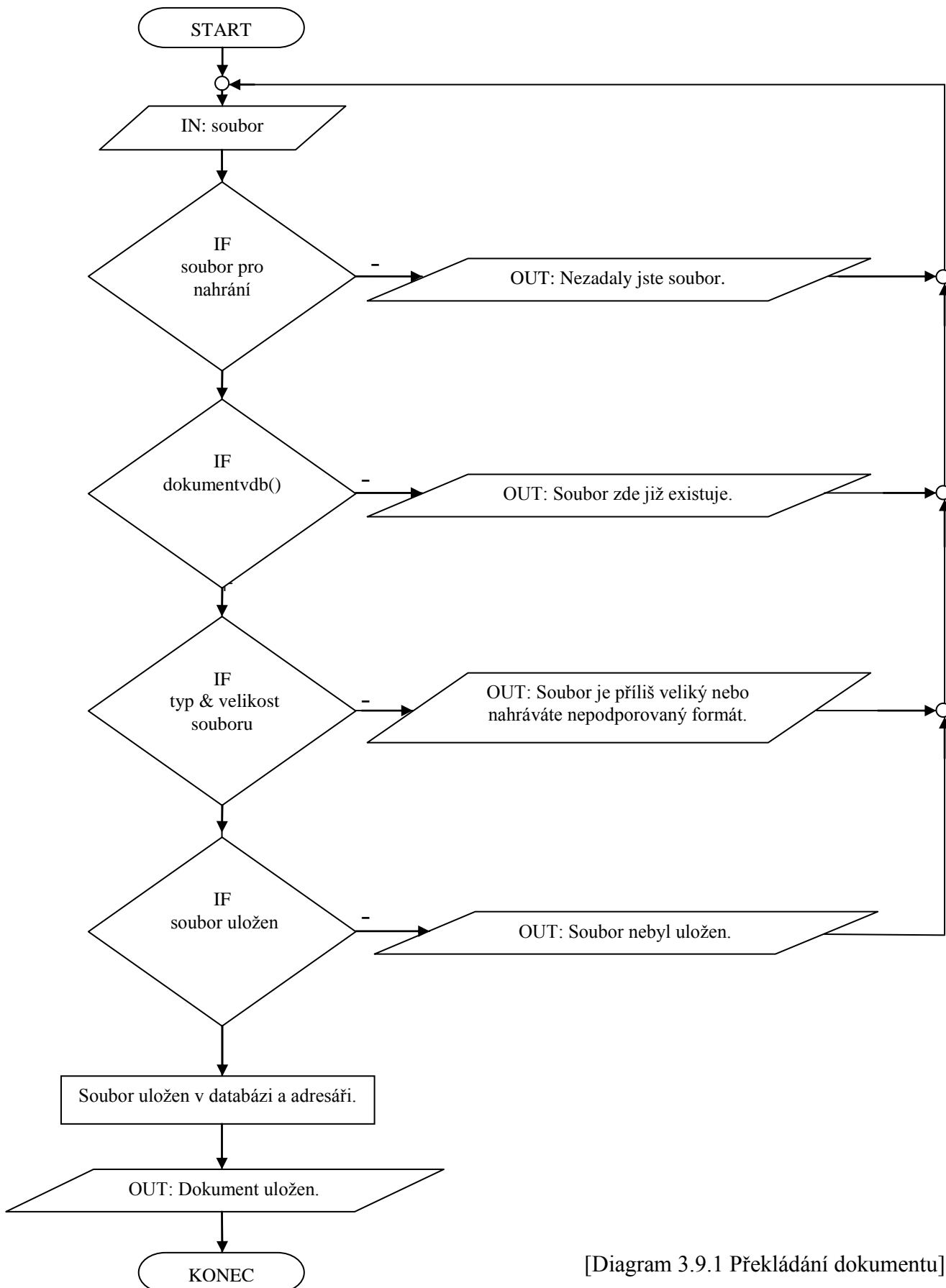
3.8. Smazání účtu



[Diagram 3.8.1 Smazání účtu]

Při smazání účtu je pro uživatele nutné, aby napsal své uživ. jméno. Po napsání uživ. jména se porovnává se jménem, které je výsledkem funkce *nazevuzivatele ()* (Kap 3.2), do které se jako vstupní hodnota zadává proměnná *id*, která je uložena v session. Shodují-li se jména, smaže se session, účet z databáze a vypíše se, že byl účet úspěšně smazán. Při neshodě jmen je vrácena varovná hláška a uživatel je opět tázán na jeho uživ. jméno.

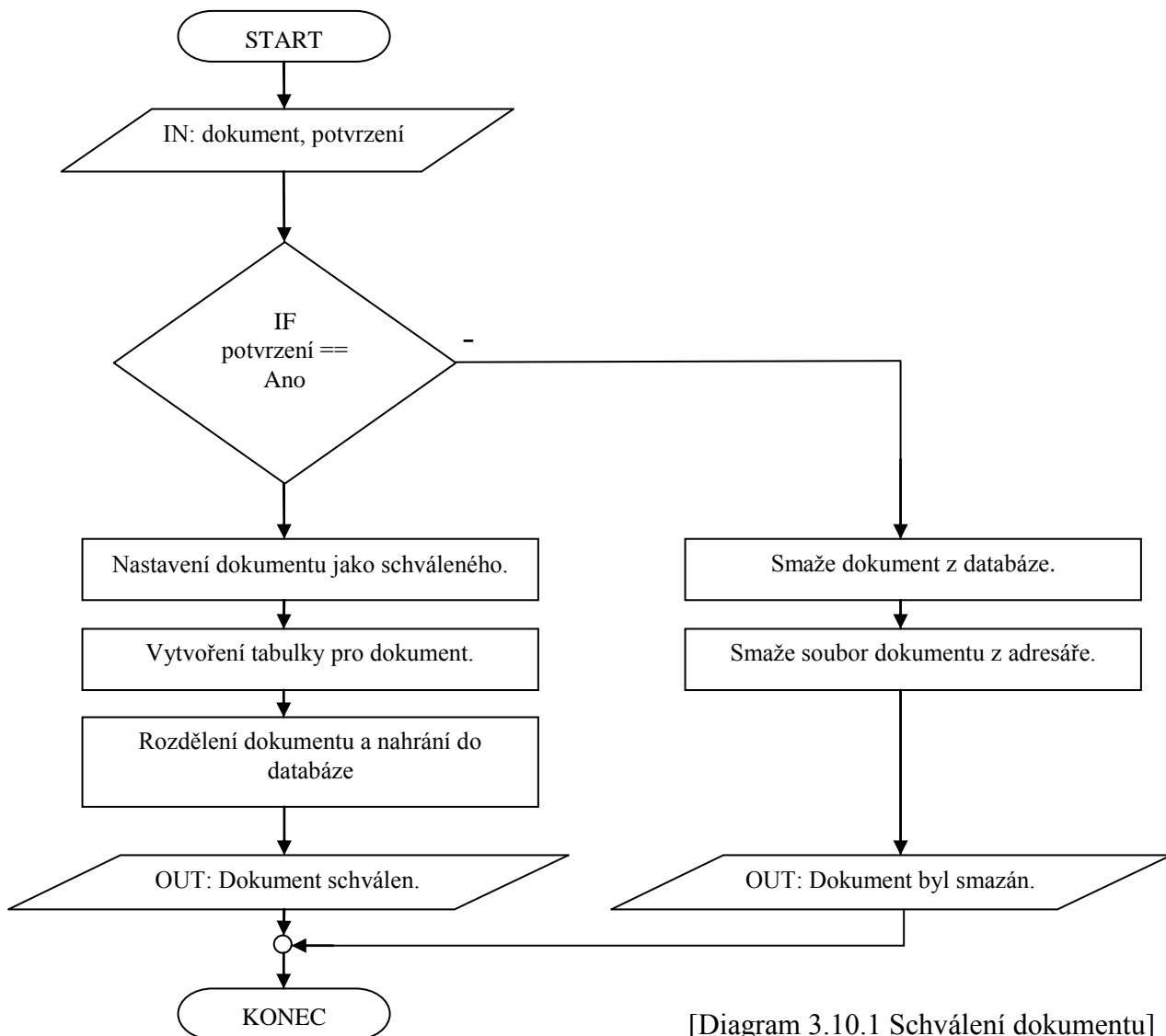
3.9. Uložení dokumentu



[Diagram 3.9.1 Překládání dokumentu]

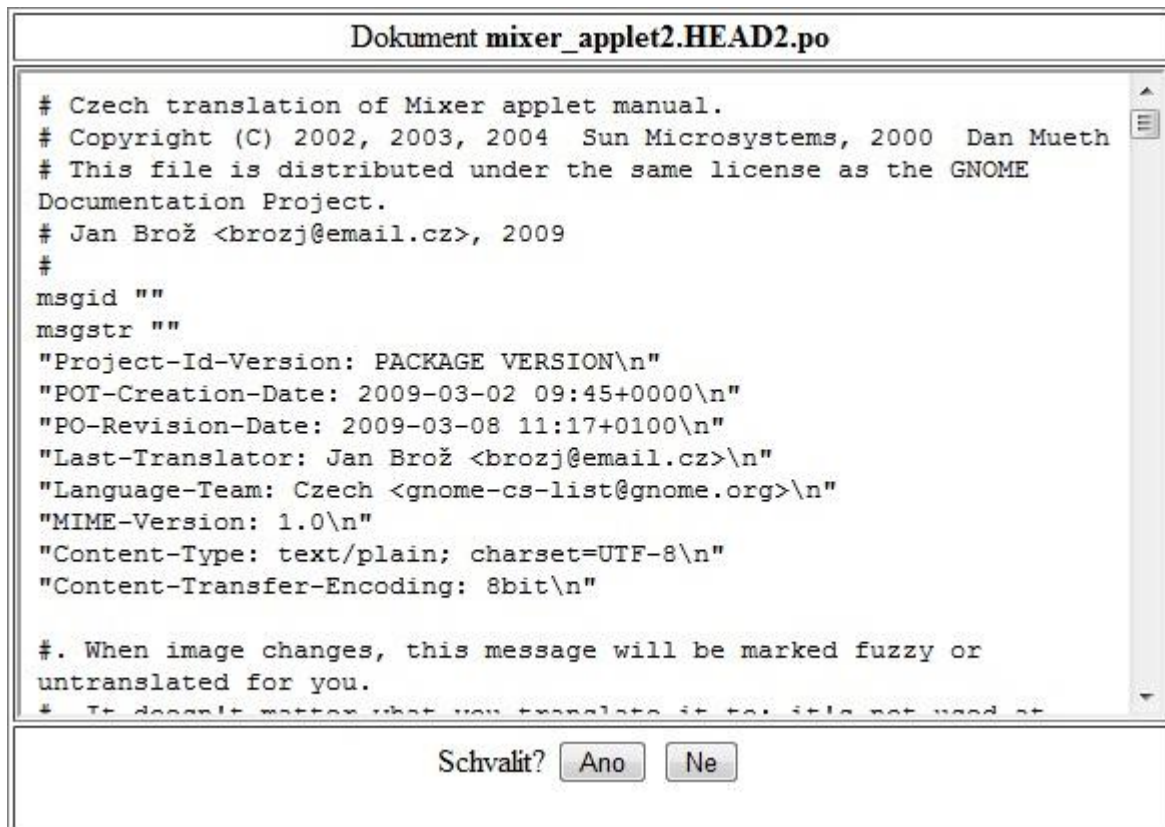
Při ukládání souboru testujeme, zda uživatel vůbec vyplnil adresu k souboru, jestli zde soubor již není a zda má soubor dovolenou velikost a podporovaný formát. Pokud se toto vše provede bez problému, soubor je uložen do databáze a následně do adresáře a čeká na schválení adminem.

3.10. Schválení dokumentu



[Diagram 3.10.1 Schválení dokumentu]

Admin si otevře soubor, který čeká na schválení, a zkontroluje ho. Pokud potvrdí schválení, soubor se nastaví jako schválený, takže ho překladač mohou překládat, vytvoří se tabulka v databázi, do které se soubor rozdělí podle řádků, a vypíše se zpráva informující o schválení souboru. V opačném případě se soubor smaže z databáze a adresáře, poté jsme informováni o smazání dokumentu.



[Obr. 3.10.1 Schválení dokumentu]

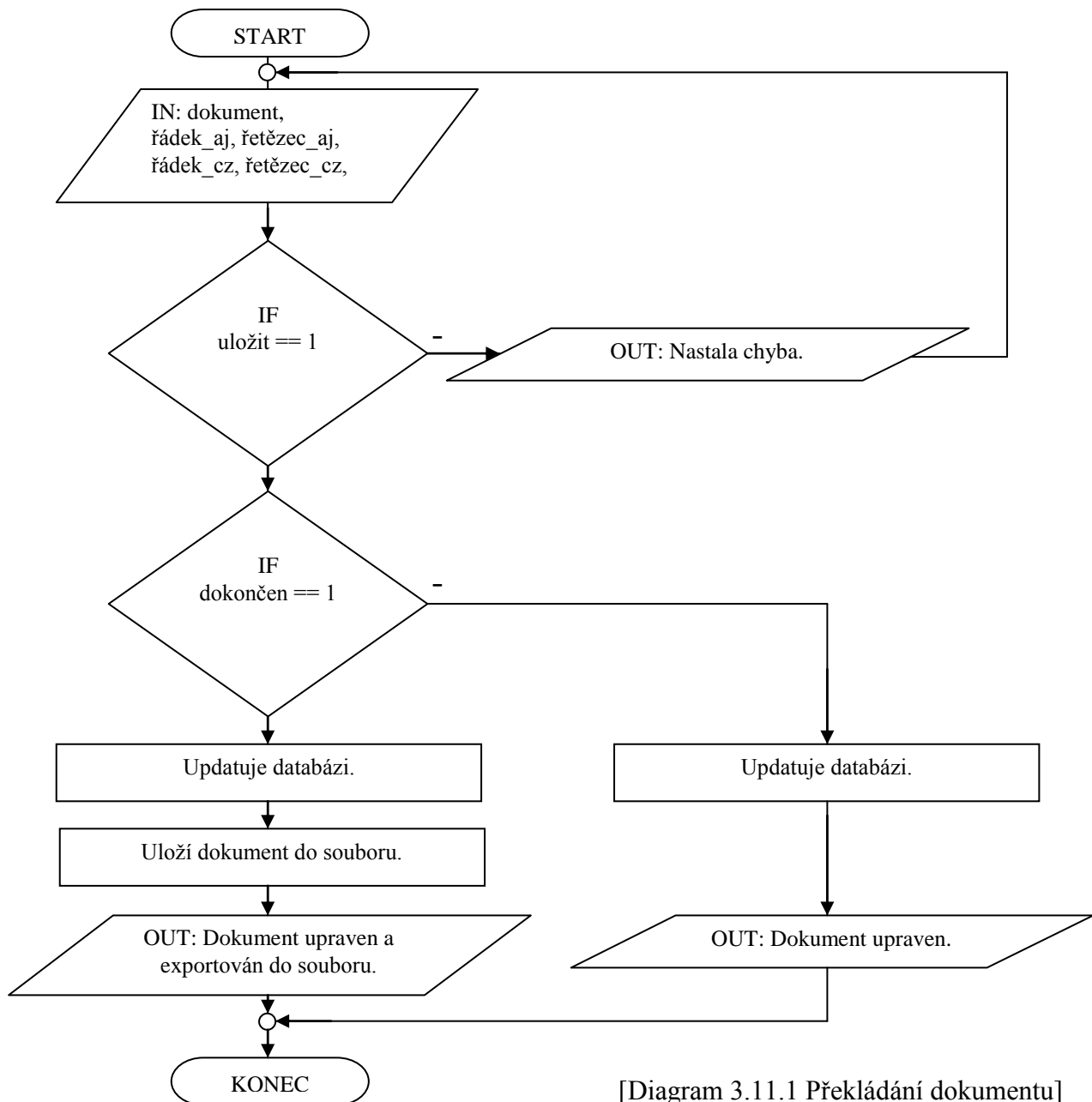
3.11. Překládání dokumentu

Soubor byl rozdělen na řádky, ale všechny řádky nechceme vypsat pro úpravu. Vypisujeme pouze řetězce začínající na *msgid* a *msgstr*. Tato část řetězce nám naznačuje, že se jedná o anglický či český text.

Celý dokument se načte z databáze a testuje se, zda řetězec začíná právě na *msgid*. Pokud tomu tak je, v řetězci se nachází anglická část licence a začne se provádět série příkazů, které nám vyberou jen text pro úpravu, otestují délku vybraného řetězce, příliš dlouhý řetězec je zkrácen funkcí *delka_textu()* a je zobrazen v tlačítku. Do tlačítka je uložen celý překládaný řetězec i s číslem řádku daného řetězce.

Po anglickém řetězci vždy následuje řetězec pro český překlad. Stačí nám tedy jen zvětšit číslo řádku o jedna a provést stejné příkazy pro tento řetězec jako pro anglický.

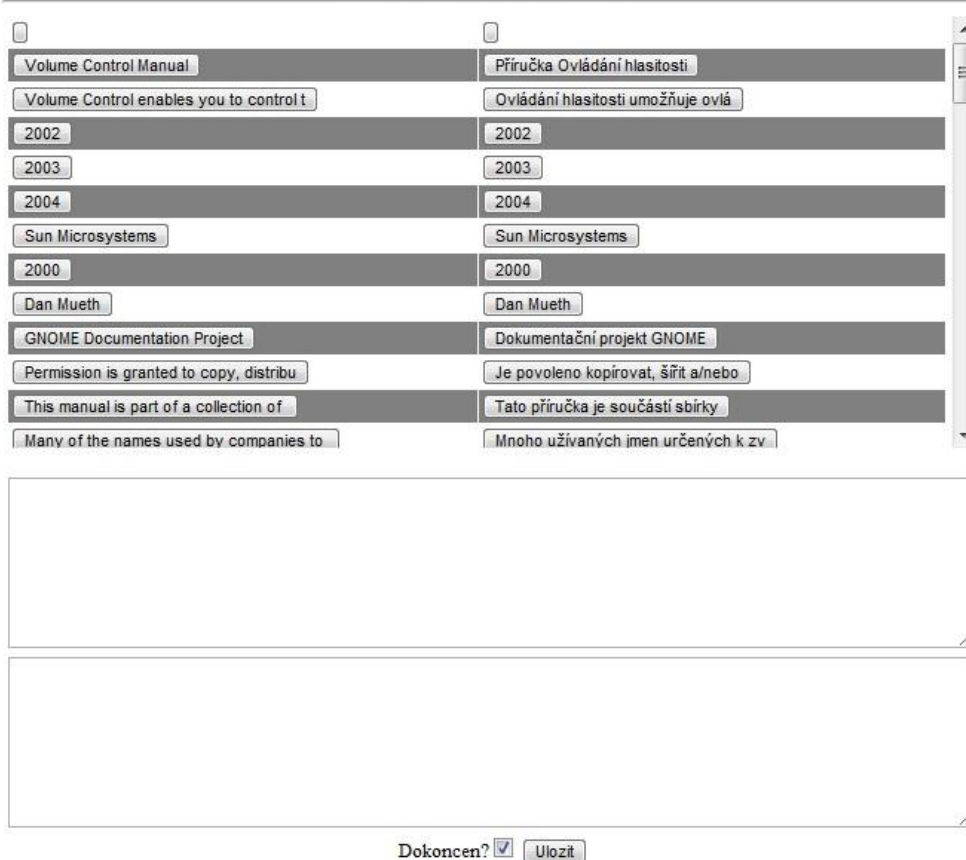
V řetězcích jsme museli změnit určité znaky na HTML entity pro správné vypisování v kódování UTF-8.



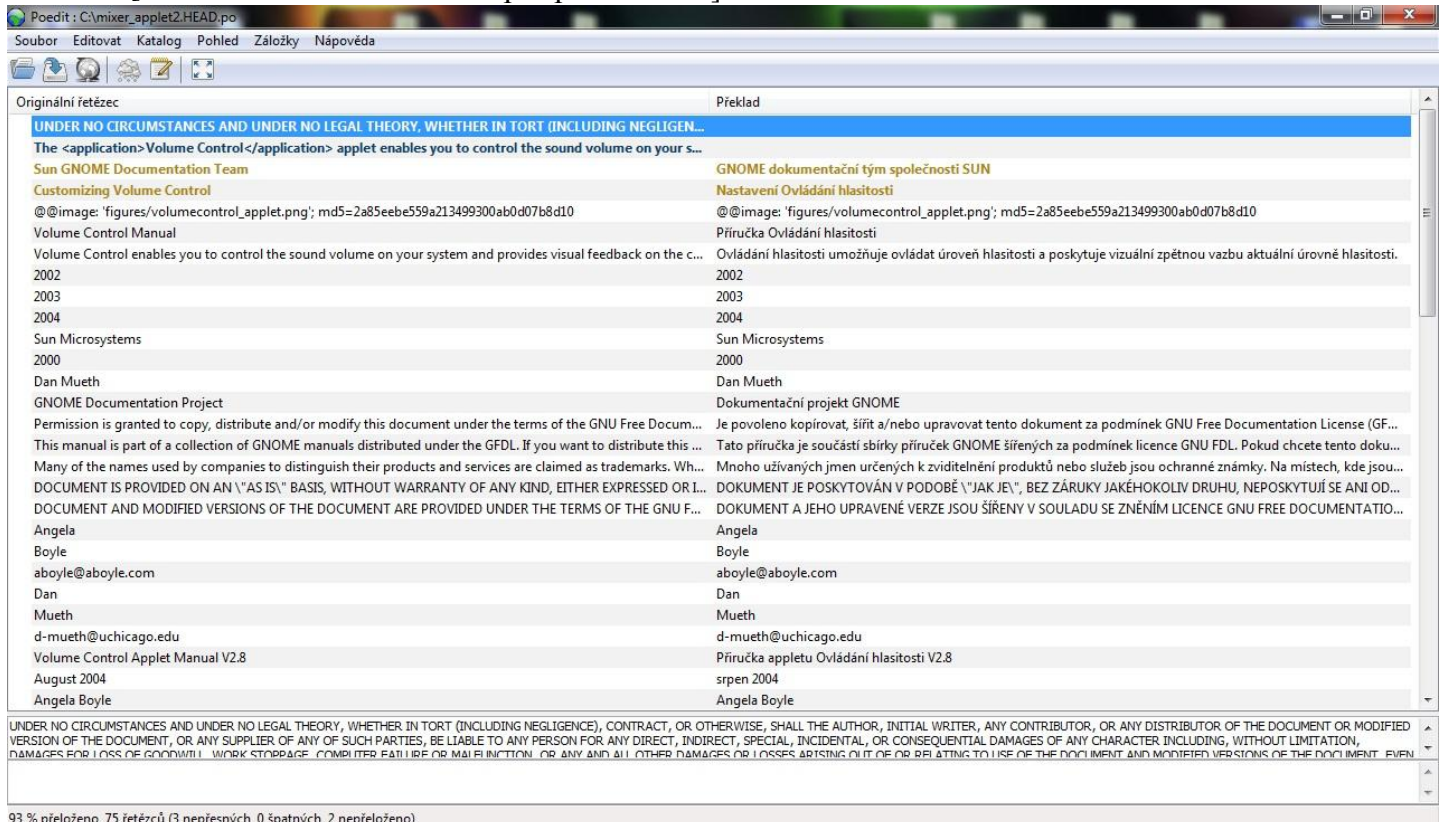
[Diagram 3.11.1 Překládání dokumentu]

Zvolený dokument pro překlad se zobrazí v prostředí (obr. 3.11.1), které je podobné PoEditoru (obr. 3.11.2). Překladač vždy upraví řetězec, uloží jej a může pokračovat v práci. Po dokončení překladu se dokument nastaví jako dokončený a automaticky se exportuje do souboru s koncovkou *.po.

Prekladani mixer_applet2_HEAD.po



[Obr. 3.11.1 Online rozhraní pro překladatele]



[Obr. 3.11.2 Rozhraní programu PoEdit pro překladatele]

3.12. Čtení dokumentu

Online rozhraní pro čtení dokumentu je totožné s rozhráním pro překlad. U čtení pouze chybí funkce pro uložení změn řetězců. Samozřejmě si přeložený soubor můžete stáhnout.

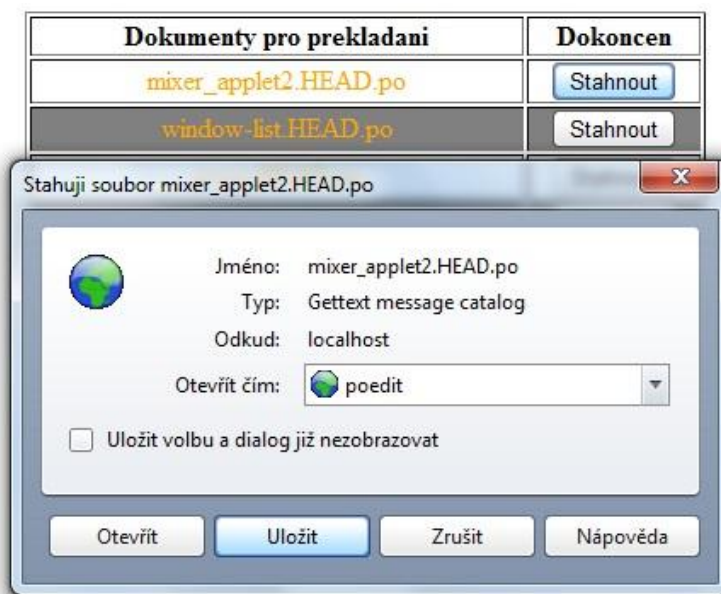
3.13. Ukládání dokumentu z databáze do souboru

Pokud překladatel nastaví dokument jako dokončený, automaticky se exportuje do souboru a lze si jej stáhnout.

Načte se celý soubor, který je rozložen na řádky. Řádky se spojí v jeden řetězec a na konec řádku se vždy přidá znak `\n`, který nahradí námi vložený znak `;\n`Line označující konec řádku. Dále se vytvoří nový soubor ve složce `dokumenty_hotove` a řetězec se do něj uloží.

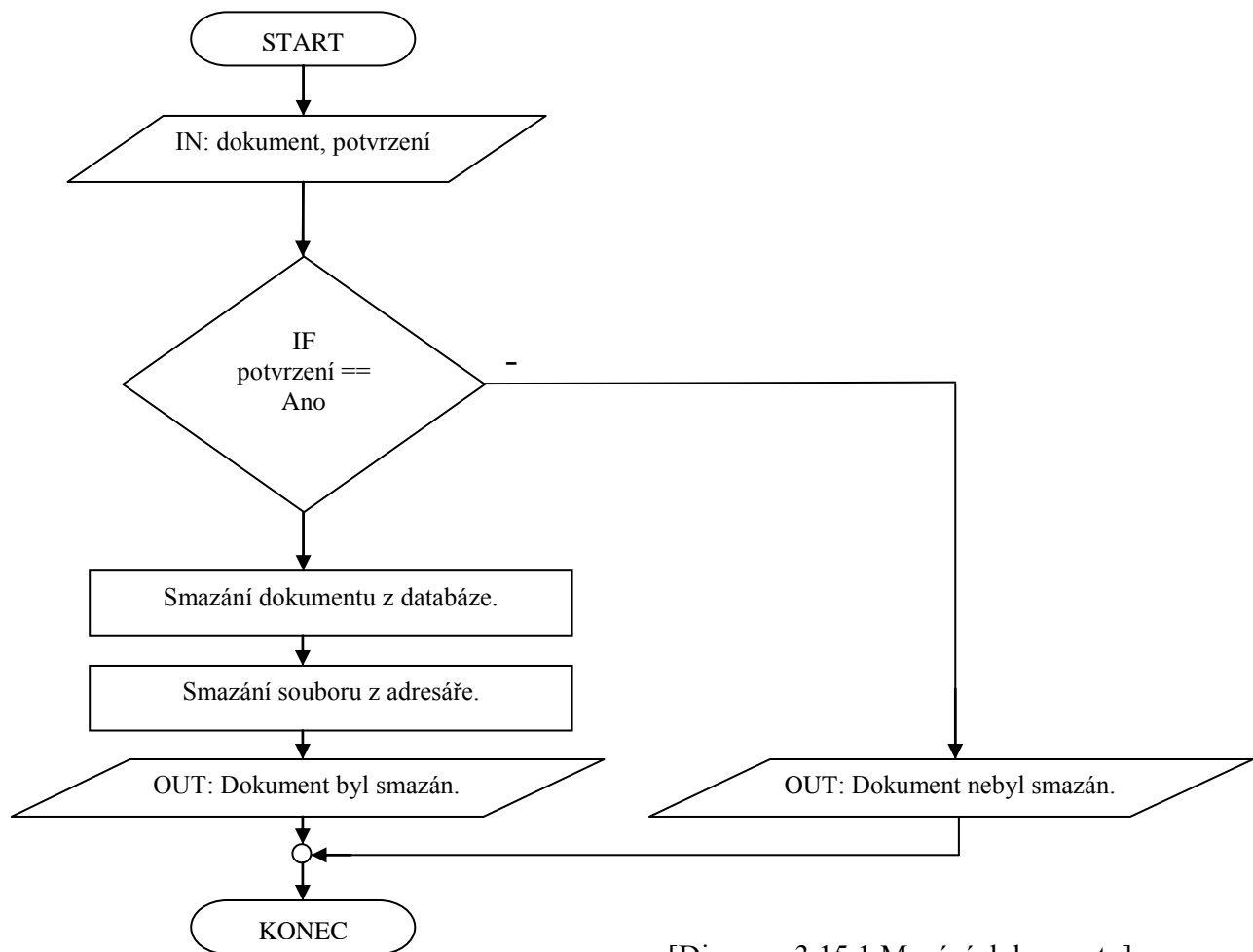
3.14. Stažení dokumentu

Pokud je soubor přeložen, každý si jej může stáhnout. Soubor dokumentu má opět koncovku `*.po`, jako při uložení do překladače a lze jej bez problému otevřít v programu PoEdit.



[Obr. 3.14.1 Stažení dokumentu]

3.15. Mazání dokumentu



[Diagram 3.15.1 Mazání dokumentu]

Mazání dokumentu je velice podobné schválení dokumentu (obr. 3.10.1). Admin si otevře dokument z původního souboru a po odpovědi, zda chce dokument smazat či nikoli, se provede následující větev podmínky. Při potvrzení je dokument smazán z databáze a soubor smazán z adresáře.

3.16. Slovník a seznam frází

Slovník a seznam frází může upravovat jen admin. Slouží pro překladatele jako výpomoc při překládání dokumentů. Více se bude používat seznam frází, protože všichni víme, jaký je problém s významy slov a spojení určitých slov v anglickém jazyce.

Ve slovníku i seznamu lze hledat určitá slova, a tak nemusíte projíždět celý seznam a hledat je.

3.17. Uživatelé

V této sekci admin může upravovat pouze pravomoci u uživ. účtů. Vidí ovšem i přihlašovací jméno, heslo a id účtu.

Hledání uživatele:

Uzivatele

ID	Jmeno	Heslo	Admin	Prekladatel	
1	striker	Ax007	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Upravit"/>
2	zkouska	zkouska	<input type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Upravit"/>
3	prekladatel	prekladatel	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Upravit"/>
4	zdenek	abc	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="button" value="Upravit"/>

[Obr. 3.17.1 Výpis uživatelů pro úpravu privilegií]

Upravit uzivatele

ID	Jmeno	Heslo	Admin	Prekladatel	
<input type="text" value="3"/>	<input type="text" value="prekladatel"/>	<input type="text" value="prekladatel"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="Ulozit"/>

Pokud chcete nastavit, pak zaskrtnete.

[Obr. 3.17.2 Úprava privilegií vybraného uživatele]

4. Závěr

Mnou vytvořený webový překladač obsahuje pouze základní možnosti úpravy účtů a správy dokumentů. Chybí zde podrobnější informace o dokumentech a uživateli a nějaké jednoduché fórum nebo chat. V této práci bych chtěl dále pokračovat a dodělat tento překladač na určitou úroveň. Se svojí prací jsem velice spokojen, vůbec jsem na začátku nedoufal, že to bude vypadat tak, jak to vypadá. Kdybych teď začínal s prací od začátku, moc věcí bych asi nezměnil, snažil jsem se programovat jednoduše a přehledně, snad se mi to povedlo. Určitě je zde hodně programátorských chyb, ale překladač funguje tak, jak by měl, a problémy jsem nepozoroval žádné nebo jsem je opravil.

Doufám, že má práce najde plné využití v překládání GNU gettext licencí, pro kterou byl můj překladač navržen. Díky mému překladači lze tyto licence překládat odkudkoliv s přístupem k internetu, třeba i z PDA, a stáhnout si hotové překlady později doma. Nepotřebujete k překládání žádné programy, jen se zaregistrovat.

5. Seznam obrázků a diagramů

Obr. 3.4.1	Menu nepřihlášeného uživatele
Obr. 3.4.2	Menu uživatele bez práv
Obr. 3.4.3	Menu překladatele
Obr. 3.4.4	Menu admina
Obr. 3.10.1	Schválení dokumentu
Obr. 3.11.1	Online rozhraní pro překladatele
Obr. 3.11.2	Rozhraní programu PoEdit pro překladatele
Obr. 3.14.1	Stahování dokumentu
Obr. 3.17.1	Výpis uživatelů pro úpravu privilegií
Obr. 3.17.2	Úprava privilegií vybraného uživatele

Diagram 3.5.1	Registrace účtu
Diagram 3.6.1	Přihlášení účtu
Diagram 3.7.1	Odhlášení účtu
Diagram 3.8.1	Smazání účtu
Diagram 3.9.1	Překládání dokumentu
Diagram 3.10.1	Schválení dokumentu
Diagram 3.11.1	Překládání dokumentu
Diagram 3.15.1	Mazání dokumentu

6. Použitá literatura

Úvod do CSS | CSS. (2004). [Použito 9. Březen 2010]:

Dostupné z URL: <http://www.webtvorba.cz/css/uvod-do-css.html>

DOSER.CZ - Dominik Fišer - webové technologie PHP, MySQL, JavaScript, AJAX, CSS, XML, XHTML, ... (2001-2010). [Použito 9. Březen 2010]:

Dostupné z URL: <http://www.doser.cz/html.php>

Ořez a skrollování - *Jak psát web píše Yuhù, Dušan Janovský* [Použito 4. Únor 2010]:

Dostupné z URL: <http://www.jakpsatweb.cz/css/css-orez.html>

MySQL (1) - pestrý svět databází -Linux Software - Petr Zajíc (1. 3. 2005) [Použito 9. Březen 2010]:

Dostupné z URL: http://www.linuxsoft.cz/article.php?id_article=731

PHP (1) - Historie a budoucnost -Linux Software - Petr Zajíc (27. 5. 2004) [Použito 9. Březen 2010]:

Dostupné z URL: http://www.linuxsoft.cz/article.php?id_article=171

PHP (69) - Stavíme portál 2. -Linux Software - Petr Zajíc (22. 11. 2004) [Použito 2. Prosinec 2010]:

Dostupné z URL: http://www.linuxsoft.cz/article.php?id_article=524

PHP (53) - Sessions -Linux Software – Petr Zajíc (8. 10. 2004) [Použito 4. Prosinec 2010]:

Dostupné z URL: http://www.linuxsoft.cz/article.php?id_article=440

Jiří Kosek, PHP-tvorba interaktivních internetových aplikací, Grada Publishing, spol. s.r.o. (1999)
ISBN 80-7169-373-1